

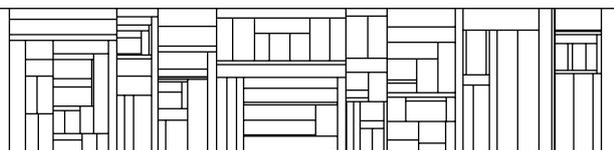
**Proceedings of the  
First International Workshop on  
Activity Theory Based  
Practical Methods for IT-Design  
Copenhagen, Denmark, September 2-3, 2004**

**Olav W. Bertelsen, Mikko Korpela, and Anja Mursu (eds.)**

DAIMI PB - 574

December 2004

**DEPARTMENT OF COMPUTER SCIENCE  
UNIVERSITY OF AARHUS**  
IT-Parken, Aabogade 34  
DK-8200 Aarhus N, Denmark





# ATTIT 2004

Proceedings of the First International Workshop on  
Activity Theory Based Practical Methods for IT-Design

2-3. September 2004,  
Copenhagen, Denmark

Organized and edited by

Olav W. Bertelsen, Mikko Korpela & Anja Mursu



# Preface

ATIT 2004 - the First International Workshop on Activity Theory Based Practical Methods for IT Design - took place in Copenhagen, Denmark, September 2-3, 2004, in conjunction with The Third Nordic Conference on Cultural and Activity Research.

For two decades, the activity theory framework has gained increasing popularity within IT-design - information systems, human-computer interaction, computer supported cooperative work, participatory design, software, etc. It has, however, mostly been applied as a conceptual framework for researchers. There have been a few attempts to fill the gap between academic theorizing and practical design, but most often they have not quite succeeded in being a genuine resource for practical design. Based on the long list of successful applications of AT in this domain it seemed to be time to extract a collection of practical methods.

The purpose of the workshop was to discuss and refine methods for IT design based on activity theory. Thereby, stimulate the evolution of design methods beyond pure analyses, and general perspectives. The call for papers required participants to submit a practical AT based method or technique for IT-design. Submissions should be composed of a description of the method or technique itself, in a format that would be suitable for a textbook for practitioners, and a short paper, reflecting on the method, its basis in activity theory, its history, its use in practice etc.

The present collection consists of revised version of papers presented at the workshop. All presenters have been asked to integrate the academic reflection into the practical description. In addition to these papers we have included a short overview paper by Mikko Korpela as an introduction to the four papers from the ActAD group in Koupio.

We wish to thank all the participants for their contribution

Olav W. Bertelsen, Mikko Korpela & Anja Mursu

# TABLE OF CONTENTS

Method and tools for analysis of collaborative problem-solving activities .....	5
<i>Nikolaos Avouris, Vassilis Komis, Georgios Fiotakis, Angeliqe Dimitracopoulou &amp; Meletis Margaritis</i>	
Activity Walkthrough - a quick user interface evaluation without users .....	17
<i>Olav W. Bertelsen</i>	
Activity Theory applied to Requirements Elicitation of Multi-Agent Systems.....	22
<i>Rubén Fuentes, Jorge Gómez, Juan Pavón, &amp; Lorna Uden</i>	
Appropriating Patterns for the Activity Theory Toolkit.....	32
<i>Elizabeth S. Guy</i>	
Systemic-Structural Activity Analysis of HCI Video Data.....	48
<i>Steven Robert Harris</i>	
Shared Object – Asymmetrical and Conflicting Objectives .....	64
<i>Stefan Holmlid &amp; Henrik Artman</i>	
RUBICON: Transforming work practice descriptions into IT requirement elicitation via AT modelling .....	71
<i>Kristina Lauche</i>	
The Activity Theoretical Iterative Evaluation Method.....	85
<i>Amanda Quek &amp; Hanifa Shah</i>	
Designing Weak and Strong Interactions: An Approach based on Cultural-Historical Psychology.....	96
<i>Antonio Rizzo, Simone Pozzi, Luca Save &amp; Mark Sujjan</i>	
IS Design as Domain Construction .....	110
<i>Lars Taxén</i>	
Narrative Transformation: Designing Work Means by Telling Stories .....	122
<i>Bettina Törpel</i>	
ActAD in information systems analysis and design: Academic reflection.....	134
<i>Mikko Korpela</i>	
Analyzing the present state of an activity .....	137
<i>Anja Mursu</i>	
IT needs exploration .....	147
<i>Marika Toivanen</i>	
Acquiring information about IS development needs from the viewpoint of an organisational unit .....	152
<i>Heidi Häkkinen</i>	
Deriving software architecture from activity analysis .....	157
<i>Annamari Riekkinen</i>	

# Method and tools for analysis of collaborative problem-solving activities

Nikolaos Avouris<sup>1</sup>, Vassilis Komis<sup>2</sup>, Georgios Fiotakis<sup>1</sup>, Angelique Dimitracopoulou<sup>3</sup>,  
Meletis Margaritis<sup>1</sup>,

<sup>1</sup>Electrical & Computer Eng. Dept, HCI Group, <sup>2</sup>Early Childhood Education Dept.  
University of Patras, 26500 Rio Patras, Greece

<sup>3</sup>LTEE Laboratory, Univ. of Aegean, 85100, Rhodes, Greece

{N.Avouris, Fiotakis, Margaritis }@ee.upatras.gr, Komis@upatras.gr, ADimitr@rhodes.aegean.gr

## ABSTRACT

This paper provides an overview of the “Object-oriented Collaboration Analysis Framework (OCAF)” a method proposed for analysis and evaluation of collaborative problem solving activities of groups of actors, mediated by collaboration-support technology. This framework puts emphasis on the abstract and tangible objects that appear during the development of a solution to a given problem. The notions of the “*objects’ histories*” and “*objects’ ownership*” are introduced by this analytical framework. In the paper tools that have been developed to support this framework are described, together with extracts of studies that have been undertaken, during which OCAF has been effectively used.

## Keywords

Collaborative Problem Solving, Analysis of collaborative activity, OCAF

## INTRODUCTION

Analysis of activities of groups of people engaged in problem solving- collocated or at a distance- is important for gaining an insight in the problem-solving process and understanding of collaboration. Socially inspired theories, supported by the growing development of network and CSCW technology, have increased research on technology-based collaborative problem solving environments. The outcomes usually influence our considerations on quality of the collaborative problem solving process, as well as the design of the *collaboration-support tools* involved. According to both these perspectives, the methodological issues of collaboration analysis are of prime importance,

given that they are directly related to the development of this research and technology area and the common understanding of the various disciplines involved.

In problem-solving collaborative learning activities in which the computer environment constitutes itself a mediational resource, it contributes to the creation of a shared referent between the social partners (Rochele et al, 1995). Typically these direct manipulation environments are characterised by actions on objects representing entities or on concepts meaningful to the users. Usually operations on these objects have a reversible incremental effect on the ‘environment’ represented on a shared computer screen. Often more than one actor interact directly or indirectly with the objects in this world modifying their state, communicating between them and through the objects, as they advance problem solution. Analysis of these problem-solving situations is usually done through discourse analysis (Baker et al., 1999), task analysis (Van Welie et al, 2003), communication and interaction analysis (Bodker 1996), or even a combination of methods, with the objective to evaluate the situation, the problem-solving process and often the tools used. However in these analysis techniques the actors and the dialogues are usually the centre of attention, while the developed objects enter the scene as items on which operations are effected and as subjects of discussion. Using Activity Theory as a conceptual framework for analysis of such activities shifts the emphasis to the activity, which is directed towards *objects* that can be hierarchically decomposed. Breaking down the activity to actions and primitive operations directed towards these objects, permits refinement of the process and analysis of the activity at various levels of abstraction.

In this paper we outline an innovative framework for analysis of collaborative problem solving activities, inspired by key aspects of Activity Theory. This framework has been used for conceptualization of the situation of groups of individuals engaged in exploratory and design problem-solving activities and for evaluation of the effectiveness of the design of IT tools supporting the process. This methodological framework is called “Object-

oriented Collaboration Analysis Framework (OCAF)” and was originally proposed by Avouris et al. (2002, 2003). Recently, analysis tools have been built to support this framework, while OCAF has been used in a number of field studies investigating various aspects of collaborative problem solving (e.g. Komis et al. 2002, Margaritis et al. 2003, Avouris et al. 2004).

In this paper we outline the main characteristics of the OCAF method, extending and refining the original proposal. The findings of the last two years of use of the method and the experience gained by the implication of the tools developed (*Synergo* and *ColAT*, described in Avouris et al 2004), have yield an improved process, discussed here.

OCAF studies the activity through the objects of the solution, that is the objects that exist in the problem-solving context, which become the center of attention and are studied as entities that carry their own history and are “acted upon” by their owners. This perspective produces a new view of the process, according to which the solution is made up of structural components that are “owned” by actors who have contributed in various degrees to their existence. This view of the world, can be useful, as it reveals the contribution of the various actors in parts of the solution, and the relevant focus shifts (Bodker 1996, Bertelsen and Bodker, 2003), identifies areas of intense collaboration in relation to the final solution and can relate easily to other analysis frameworks like interaction analysis.

In this paper, a notation of the OCAF model is proposed. Subsequently, an outline of the OCAF method is included together with presentation of the functionality of the tools that have been proposed to support the method. The first tool, called *Synergo*, is associated to a synchronous collaboration-support environment, which permits direct communication and problem solving activity of a group of distant users, manipulating a shared diagrammatic representation. Through the *Synergo* analysis tools, the researcher can playback the activity off-line and annotate the activity and the produced solution using an annotation scheme which can be defined and adapted according to the specific objectives of the study. The second tool, called *Collaboration Analysis Tool* (ColAT) is a tool to handle data of various forms, collected during field evaluation studies of collaborative activities and review the activity by building interpretative structures of operations and actions. Examples of use of the framework and the tools in collaborative problem-solving situations are also presented.

The importance of the proposed framework is also related to the fact that the existing Activity Theory based methodologies supporting information technology design, e.g. Activity Checklist (Kaptelinin et al. 1999), AODM (Mwanza 2002), ActAD (Korpela et al., 2000) do not include explicit models and tools for the analysis phase, so the proposed here framework can be seen as a proposal

with wider implications.

## MODELLING COLLABORATIVE PROBLEM SOLVING ACTIVITY

In this section we describe the key parameters through which we can model collaborative problem solving activity. The model of the activity is going subsequently to be used for analysis and evaluation of the process through the proposed method.

We suppose that the activity involves a group of subjects (actors) who are engaged in collaborative problem solving mediated by computing technology. The main motive of the activity is to produce collaboratively a solution to a given problem. This solution takes the form of a representation in symbolic form. The generation of this solution involves manipulation of intermediate objects (either tangible or abstract ones). Problem solving activity is usually considered as a process of refinement of abstract ideas (“abstract objects”) and externalization of these ideas in the form of parts of the solution to the given problem. Collaborative activity is based on communication, which takes the form of either direct communication acts or by observing operations in the shared activity space (feed-through communication, Dix et al. (1999)). Operations of the group members are defined as events that are either non-trivial changes of the state of the activity space or communication messages. An example of such event is shown in fig.1 from a collaborative activity in physical space.

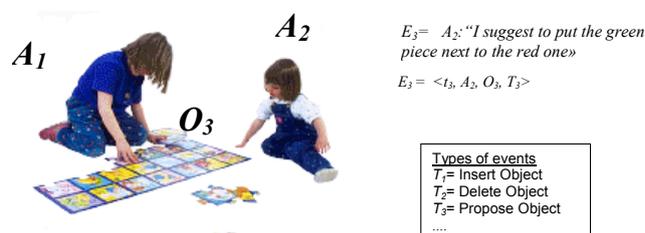


Fig. 1 Two partners  $A_1$  and  $A_2$  manipulate objects  $O_1, O_2, \dots$  and interact during problem solving.  $E_3$  is an example of an event.

The activity is defined according to the following four dimensions:

**The time dimension:** A universal activity time is assumed. This is assumed discrete in order to handle issues of concurrency of distributed activities, as discussed below.

**The actors dimension:** All actors, remote or collocated, who are involved in this activity are defined here. If we assume  $k$  actors, then this dimension is related to the set  $A = \{A_1, A_2, \dots, A_k\}$ .

**The objects dimension:** All objects that are involved in the problem solving activity define a set of  $l$  components:

$O = \{O_1, O_2, \dots, O_\ell\}$ . These objects can be either existing tangible objects (digital or physical), objects that can be built using available tools, or conceptual objects. In the example discussed in the following section these objects are components of a *concept map*. In the frame of the tools discussed later a solution is considered as made of:

- (i) concrete components (objects that compose the final solution),
- (ii) rejected components and
- (iii) abstract components.

**The typology of event dimension:** This is a dimension through which interpretation of the activity can take place. We assume that there is an existing analytical framework, which defines this typology. If  $r$  is the finite number of expected event types, then we define a set  $T = \{T_1, T_2, \dots, T_r\}$  as the analytical framework of the study. While in the original OCAF proposal we have included such a closed set  $T$ , see fig 2 from (Avouris et al. 2003), in this current version, we consider the method as independent of a specific analytical framework, so set  $T$  can be defined by the framework user.

ID	Functional Role	Derived from :	Example
I=	Insertion of the item in the shared space	action analysis	Action: 'Insertion' of Entity "Velo"
P=	Proposal of an item or proposal of a state of an item	dialogue analysis	Message: "I believe that one entity is the firm "ABC" or "let us put the value of entity flow to state locked"
C=	Contestation of the proposal	dialogue analysis	Message: "I think that this should be linked to the entity B by the "analogue to "relation"
R=	Rejection / refutation of the proposal	action and/or dialogue analysis	Message: "What their attributes will be ? I don't agree". Or Action: "Delete" Entity "Velo"
X=	Acknowledgement/ acceptance of the proposal	Action and / or dialogue analysis	Message: "That's right" or Action: Insertion of a proposed entity
M=	Modification of the initial proposal	action & dialogue analyses	Message: I suggest we put the state to "unlock" Action: "Modify"
A=	Argumentation on proposal	dialogue analysis	Message: "I believe that I am right because this is ..."
T=	Test/Verify using tools or other means of an object or a construct (model)	actions & dialogue analyses	Message: Let us run this model to observe this part of the model behavior Action: Activate 'Graph Tool' , or 'Barchart Tool'

Fig.2 The set of types of events according to OCAF

Using the above four dimensions we can describe any given activity as a set of discrete non-trivial events produced by the actors. These define an ordered set of  $m$  events  $E = \{E_1, E_2, \dots, E_m\}$  of the activity. Each one of these events is related to meaningful operations of the actors who interact with objects of the set  $O$ . Each event is defined as a tuple  $E_{i, AOT} = (t_i, A_i, [O_i], [T_i])_i$  where  $i \in [1, m]$ ,  $t$  the event timestamp,  $A$  the actor who performed the operation of the specific event,  $O$  an optional parameter referring to the object of the specific operation and  $T$  an optional parameter which interprets the event according to the analysis framework  $T$ .

This model of the activity is based on the assumption of events of zero duration, which is necessary in order to achieve serialization of the concurrent activities that may take place during collaborative problem solving. This is

somehow restricted; however, in the case of actions of longer duration then a starting and an ending event need to be defined in order to describe such an action, which is usually of duration  $t \neq 0$ .

This is a useful model for ethnographic studies. Every time an event is produced by the actors, this is recorded and a history of such events, i.e. an ordered list of  $E_s$  can be produced, as a result of such an activity. No mental or cognitive operators are defined, as these can be generated later as interpretations of the recorded activity. This model permits further analysis and interpretation of the activity, while quantitative indices of the activity can be easily produced or visualizations can be automatically generated (Margaritis et al. 2004).

Often the mediating computer tool adheres to a typology of generated events, thus automating the task of categorization of observed events, so for instance if a software tool is used that permits a number of operations, every time such an operation is recorded, this is automatically categorized according to a scheme of analysis.

A fundamental concept of the OCAF framework is the unified interpretation of action and communication acts. As also discussed in (Baker et al., 1999) mutual understanding among collaborating actors takes place via a combination of perception of action and communication. Furthermore, depending on the provided tools facilitating dialogue in a computer-mediated environment, the collaboration mode can vary from a more action-dominant mode to a more discussion-based mode. For these reasons, it is argued that there is a need to apply a unified analysis and interpretation of both dialogue and actions (Avouris et al. 2003). In this context, communicative acts are operations that also need to be interpreted in terms of the actors intentions in relation to the activity. In particular OCAF interprets exchanged messages (written dialogues during collaboration by distance), or oral utterances (during face to face collaboration), in relation to operations towards "objects" of the activity space, using a language for action approach (Winograd, 1987, Searl 1976), defining a unifying framework for analysis of the activity, as can also be seen in fig.2.

### Views of the OCAF Activity Model

Various analytical views of quantitative or qualitative nature can be generated using this model.

Some of them are related to quantitative measures of collaboration, like *density of activity*, if a time quantum is defined of  $t_q$  length.

An alternative index that often needs to be defined during collaborative problem solving activities is that of *balance of activity* between the partners. If an activity is monopolized by a certain partner, then this may be an indication that the activity is not truly collaborative. Definition of such index is not however easy, as all events

are not of the same importance. A specific example of definition of an index of balance of activity has been proposed by Margaritis et al. (2004), related to activity that produces diagrammatic representation of a solution made of a set of interrelated objects  $O$ .

In addition, a *spatial representation* of the activity can be generated by mapping the events to the produced final solution. This is a form of a spatial representation, as the components of the solution can include the sequences of the events that lead to their creation, i.e. for each object  $O$  of the solution, we can associate the sequence of events  $E_i$  for which  $O_i$  is of a specific object  $O$ . This is defined as the *object history*.

Additional secondary indices may be associated to these objects, like the *diversity of actors*  $A$  that appear in such set of events, the length of this history, i.e. the number of events associated to a specific object, etc. Also measures of *focus* of activity and *focus shifts* can be generated through this view.

The views created by the OCAF model are useful for the analysis and evaluation of problem solving, providing a perceptual view on these parameters. This view can directly be related to the produced solution, associating the history of interaction to the items involved. Also items discussed but not included in the solution appear in this view. One can consider the generated views as an attempt

analysis) to the space dimension (predominant in diagrammatic solution representation). Various transformations of this view can make it suitable for different users. For instance, by adequate color-coding of the participants and their roles, the association of ownership to solution items could become vivid, supporting reflection of problem solvers in a metacognitive level.

One of the prime advantages of the OCAF framework is that the OCAF activity model is generated and processed by adequate automatic tools, attached to a collaboration support environment. In particular, the action part analysis can be directly automated, while the dialogue part needs human interaction for dialogues analysis approaches. These OCAF-compatible analysis tools can be used by researchers studying collaborative problem solving. Also tools for collaboration visualization can be produced for various indices like the ones discussed in this section, that can be even used by the problem solving actors as metacognitive tools in order to help them self-regulate their collaborative or problem solving process. In the following section the functionality of such tools and an example of their use is discussed through a step-by-step presentation of a method of analysis using the OCAF model and views.

### OCAF METHOD AND TOOLS

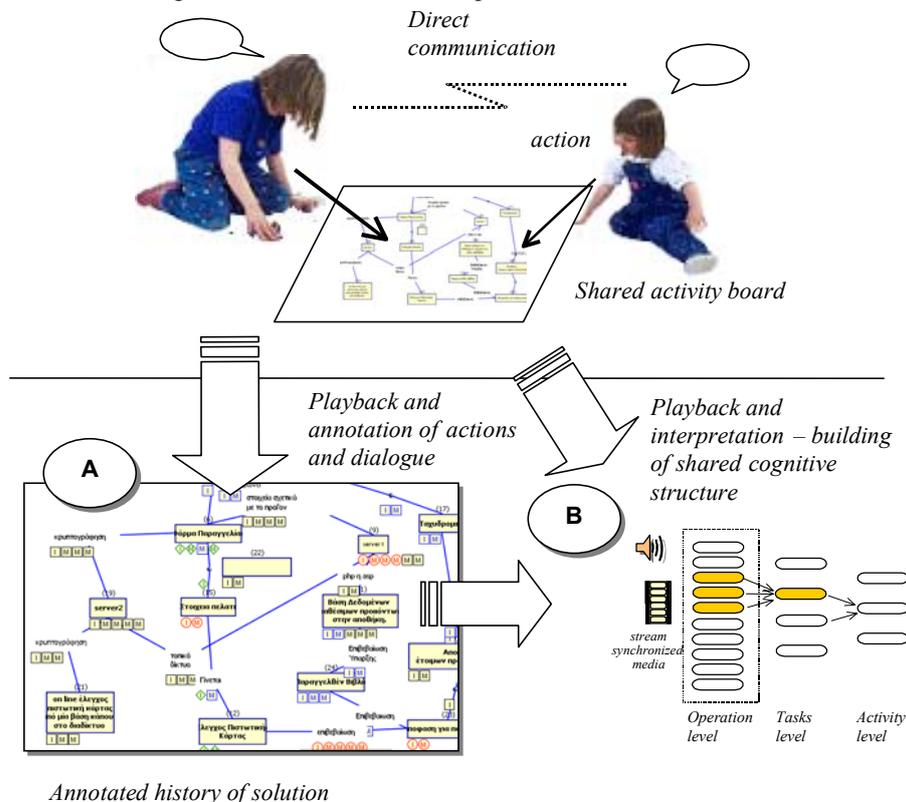


Figure 3 Overview of the OCAF method of analysis

to relate the time dimension (predominant in interaction

Collaboration is a phenomenon for which we lack adequate

analytic models. It is not claimed that the complex phenomena of social interaction and particularly of collaborative learning can be comprehensively reconstructed by analytic models. These models are bound to be partial, capturing only specific facets of actions or interactions in groups. The lack of such theoretical models is of prime concern for developers of CSCW technology. The value of an analytic model like OCAF, is related to its capacity to bring up interesting points of view and thus provide information to researchers aiming at answering relevant questions

Some of these points of view are related to quantitative aspects of interaction, and appear often in studies of collaborative environments, while others relate to a more cognitive and meta-cognitive view, as for instance is the case of solution validation strategies. These questions have been effectively tackled using OCAF, as demonstrated through various case studies.

In figure 3, an overview of the proposed method and related tools is outlined. An outline of a typical example of use of the proposed framework is discussed in this section, while in the next section specific examples of a case study of analysis are included.

In our example, a typical synchronous collaborative problem-solving situation, two or more actors, supported by networked equipment, collaborate at a distance by communicating directly and by acting in a shared activity board. A digital representation of a solution to a given problem may appear in this shared board. This activity is typically monitored through logging of the main events, recording the activity of the actors in the shared activity board and of the dialogue events, if they are in text form. In addition the dialogue can be captured, through video and audio recording, if videoconferencing technology has been implemented, while additional information of the activity and the context within which this has taken place, may be captured in other forms. The collaboration-support tool used in recent studies has been Synergo, a tool that permits collaborative building of diagrammatic solutions to problems in the form of flow charts, concept maps or other graphical representations (Voyiatzaki et al. 2004). This environment has been built using the *Abstract Collaborative Application Building Framework* developed in the frame of the ModellingSpace project (Avouris et al. 2004). Synergo includes tools for annotation of the solution according to the OCAF approach and visualization of a number of indices of the process. The analysis methodology involves three phases supported by associated tools, as discussed in the following.

#### Phase A: Definition of an event typology

During phase (A) we define an interpretative scheme of the expected operations during the problem solving activity. This scheme defines a closed set of event types *T*. In the provided analysis tool, the user can define such a set and

associate typical events included in the log file to them. In figure 4 the dialogue box through which we defined the event typology for our experiment is shown. *Proposal, Contestation, Rejection and Acknowledgement* were the events that were related to dialogue acts and *Insert, Modify, Connect* were related to events on the activity space.

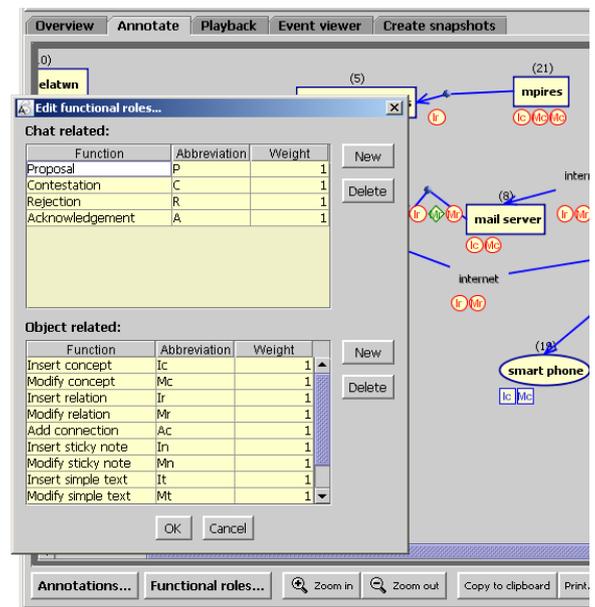


Figure 4. Definition of event typology

#### Phase B: Annotation of operations

During phase (B) the Synergo analysis tool is used for presentation and processing mainly of the logfiles, produced during collaborative problem solving activities. These logfiles contain time-stamped events, which concern actions and exchanged text messages of partners engaged in the activity, in sequential order. The logfile events are produced by exchanged control and textual dialogue messages and need to adhere to a defined XML syntax. These events can be viewed, commended and annotated by the tools discussed here. The activity can be reproduced using the Playback tool of Synergo that reconstructs the group problem solving activity on the actors' workstations desktop step by step, through a single view. Annotation of the events is done, according to the specific analysis typology defined in phase A, that permit building of an abstract view of the activity.

The activity playback and solution annotation tool is shown in fig.5. The result of this phase is an annotated history of the problem solving activity and of the produced diagrammatic solution, through this activity. In the example of fig.4 one can see the graphic representation of this history and annotation of the solution in the shared activity board. In a separate window, the history of textual dialogue events is presented. Each item of the diagrammatic solution of a problem (a concept map representing a web service in



Through these views, one may observe the level of activity during various phases of problem solving.

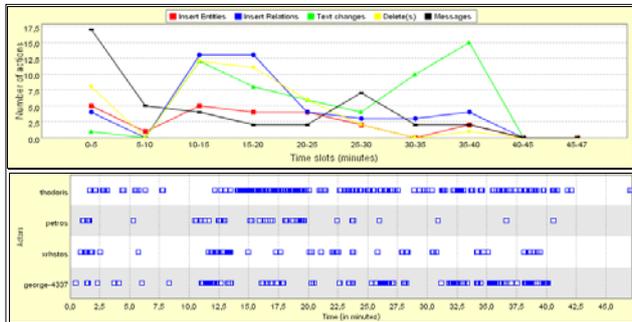


Figure 7. Views of the activity produced by Synergo: (a) density of activity per type of event, (b) activity per actor.

### Phase C: Analysis of activity

In phase (C), the *Collaboration Analysis Tool (ColAT)* environment is used for building an interpretative model of the joint activity in the form of a multilevel structure, following an Activity Theory approach. ColAT permits fusion of multiple data by interrelating them through the concept of universal activity time. The analysis process during this phase, involves interpretation and annotation of the collected data, which takes the form of a multilevel description of the collaborative activity.

The ColAT tool, discussed in more detail in (Avouris et al. 2003c), uses the form of a theater's scene, in which one can observe the action by following the plot from various standpoints. The *Operations-view* permits study of the details of action and interaction, as generated in phase B, the *Actions-view* permits study of purposeful chunks of action, while the *Activity-view* studies the activity at the strategic level, where most probably decisions on collaboration and interleaving of various activities are more clearly depicted.

This three-level model is built gradually: the first level, the *Operations level*, is directly associated to log files of the main events, produced and annotated in phase B, and is related through the time-stamps to the media like video. The second level describes *Actions* at the actor or group level, while the third level is concerned with *motives* of either individual actors or the group. In fig. 8 the typical environment of the ColAT tool for creation and navigation of a multi-level annotation and the associated media is shown. The three-level model is shown on the screen, while the video/audio window is shown on the right-hand side.

The original sequence of events contained in the logfile is shown as level 1 (operations level) of this multilevel model. The format of events of this level in XML, is that produced by Synergo, ModellingSpace and other tools that adhere to this data interchange format. Thus the output of the first phase can feed into ColAT, as first level structure.

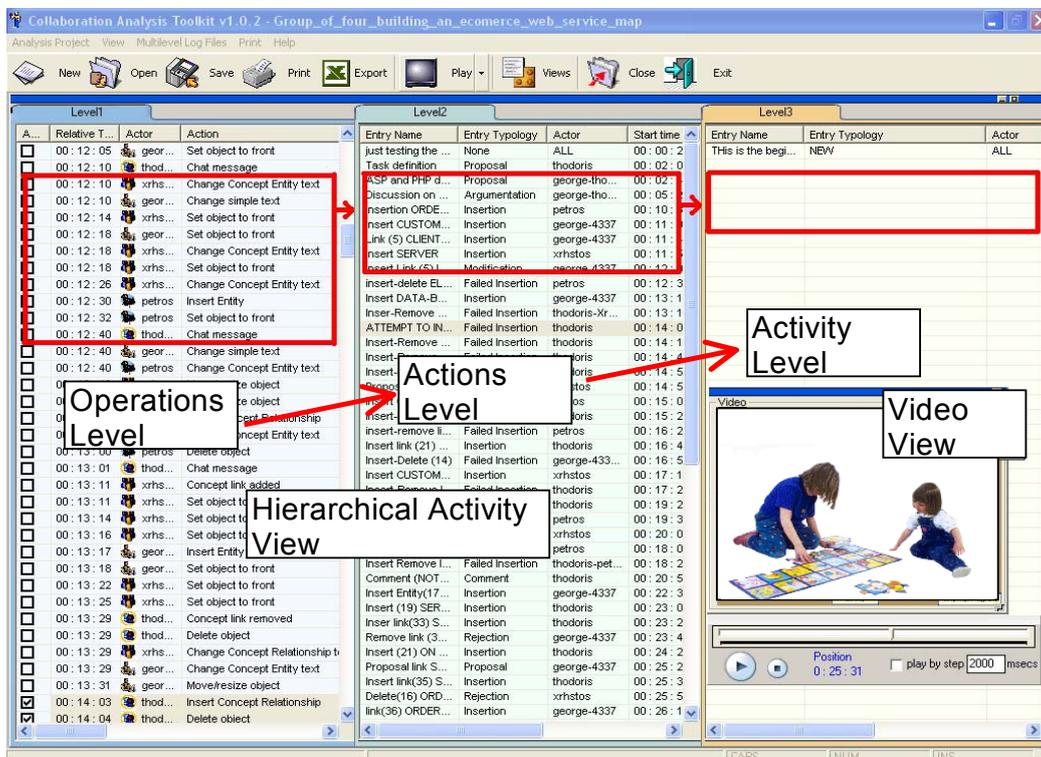


Figure 8. Navigation of multi-level interpretation of collaborative problem solving activity

A number of such events can be associated to an entry at the actions level 2. Such an entry can have the following structure: <ID, time-span, entry\_type, actor(s), comment > where ID is a unique identity of the entry, time-span is the period of time during which the action took place, type is a classification of the entry according to a typology, defined by the researcher, followed by the actor or actors that participated in the task execution, a textual comment or attributes that are relevant to this type of action entry. Examples of entries of this level are: "Actor X inserts a link ", or "Actor Y contests the statement of Actor Z".

In a similar manner, the entries of the third level (Activity level) are also created. These are associated to entries of the previous Actions level. The entries of this level describe the activity at the strategy level as a sequence of interrelated goals of the actors involved or jointly decided. This is an appropriate level for description of plans, from which coordinated and collaborative activity patterns may emerge. In each of these three levels, a different typology for annotation of the entries may be defined. This may relate to the domain of observed activity or the analysis framework used. For entries of level 1 the OCAF typology may be used, while for the action and activity level different annotations have been proposed.

ColAT permits an alternative way of representation of the action and activity level. A typical view of this representation is shown in figure 9.

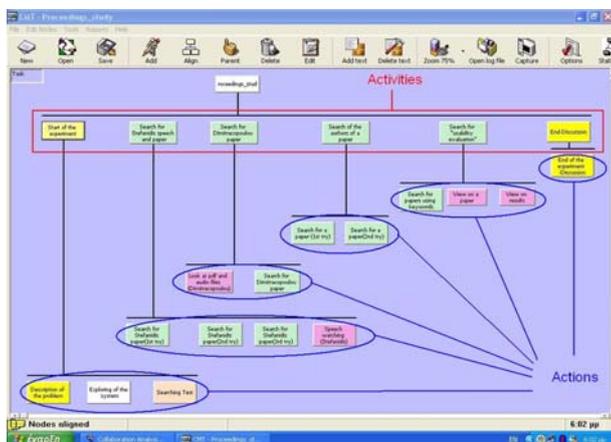


Figure 9. Graphical representation of actions and activities according to HTA

This view is one that describes the goals and tasks that an actor or a group of actors attempts to accomplish. So in figure 9 one can see the activities and actions in a graphical view similar to Hierarchical Task Analysis (HTA). Each activity and action is represented by a different colour that is established according to the OCAF scheme. This view is of high importance since it permits mapping annotated

group activity to top-down decomposition of the observed actors' activities.

#### Use of media sources in analysis

Various media, like video or audio can be viewed, using the ColAT tool, from any level of its multi-level model of the activity. As a result, the analyst can decide to view the activity from any level of abstraction he/she wishes, i.e. to play back the activity by driving a video stream from the operations, actions or the activity level. This way the developed model of the activity is directly related to the observed field events, or their interpretation.

Other media, like still images of the activity or of a solution built, may also be associated to this multilevel model. Any such snapshot may be associated through a timestamp to a point in time, or a time slot, for which this image is valid. Any time the analyst requests playback of relevant sequence of events, the still images appear in the relative window. This facility may be used to show the environment of various distributed users during collaboration, tools and other artefacts used, etc.

The possibility of viewing a process using various media (video, audio, text, logfiles, still images), from various levels of abstraction (operation, action, activity), is an innovative approach. It combines in a single environment the hierarchical analysis of a collaborative activity, as proposed by Activity Theory, to the sequential character of ethnographic data.

#### CASE STUDY OF ANALYSIS

A number of experimental studies have been performed using the outlined methodology and tools. These relate to various aspects of collaborative problem solving analysis. In Avouris et al. (2004) the group size is related to the group performance and patterns of collaborative activity. In Fidas et al. (2005) the effect of heterogeneity of the available resources has been studied for various collaborative-learning experiments. In Avouris et al. (2003) the effect of the floor control mechanism is studied, while in Komis et al. (2002) evaluation of the effectiveness of the environment in the educational process is discussed along various dimensions, like group synthesis, task control, content of communication, roles of the actors and the effect of the tools used. In these studies, various versions of the presented tools have been used.

First the Synergo tools have been used for playback and annotation of the activity, while visualizations of the collaboration factor have been produced. Subsequently the produced video and sequences of still images, along with the logfiles of the studies were fed in the ColAT environment through which the action structures of the activities were built. A specific extract of the analysis of one of these studies is described in this section.

**Context of the study**

The discussed study took place in the frame of the laboratory work of the undergraduate course “Software Internet Technology” of the Electrical & Computer Engineering Department of the University of Patras. Eighteen (18) students participated in the experiment in the frame of a scheduled laboratory class that took place in two lab sessions. A number of groups of students with similar characteristics were formed. Each group consisted of 3 or 4 members.

The members of the collaborating groups, were dispersed in the computer lab. They interacted for a certain period of time, using exclusively the Synergo environment (chat tool and a shared drawing board) in order to tackle a given data-modelling problem in a simulated distance-collaboration setting. Each collaborating group of students was asked to produce, by the end of the laboratory session, a conceptual map of a *web service*. The tutor intervened mainly at the beginning of the session to introduce the activity and the tools, and at the final stage for making comments on some of the produced solutions. Also activity logging was performed using the logging facility of the collaborative modeling tool itself.

**Building a multi-level view of the activity**

The objective of this study was to examine the effect of group size on problem solving and group coordination strategy. An additional objective was to test the usability of the collaboration-support tool in cases of groups of more than 2 members. At the end of the lab session, the observers collected field notes, the detailed logfile of events, a few snapshots (jpeg pictures) describing the main phases of the solution of the given problem. The analysis of activity was performed for all the groups that were formed. In the following we concentrate in the study of a specific four-member group.

We created a ColAT project, including all the data of the experiment, and synchronized them in the form of a master activity logfile using the appropriate tools of ColAT. The next step was to reproduce students activities, based on the analytical study of the logfile. This process requires adequate experience. Studying the logfile, we have built the Actions Level that is displayed in figure 10.

The original operations logfile contained 560 events, which were extracted from the automatically produced logfile of activity of duration of 47 min, after introducing annotations of phase B and clearing out trivial events. The purposeful *actions* built in level 2 were 69. These were related to identified goals of the actors. Certain actions of this level involved more than one actor. The actions of this level, contrary to the events of the first level have a certain duration which is defined according to the starting event and final event of the operations level. The ColAT tool through a drag and drop operation can define an action of level #2 as a set of operations of level #1. The actors

involved are identified from the actors of the primitive operations, while the tools engaged and the objects of the action are deduced from the attributes of the events of the lower level. In figure 9 an extract of the actions level for our example is shown. The operations that define an action are not necessarily consecutive in level #1. This is due to the concurrency of a collaborative activity. The typology of events defined in phase (A) may apply to this level as well. A mapping of patterns of types of operations to types of actions is in the current version of the tool performed by the analyst, while a machine learning approach that will automate this process in some degree is under investigation.

Entry Name	Entry Typology	Actor	Start time	End time	Comments
just testing the functionality of the tool	None	ALL	00:00:29	00:01:52	no task related
Task definition	Proposal	thodoris	00:02:05	00:02:32	Agreement of
ASP and PHP defined as entities	Proposal	george-thodoris	00:02:49	00:04:30	
Discussion on entity SERVER and CLIE...	Argumentation	george-thodoris	00:05:27	00:08:19	
Insertion ORDER-FORM(6)	Insertion	petros	00:10:31	00:11:05	
Insert CUSTOMER(7)	Insertion	george-4337	00:11:06	00:11:41	
Link (5) CLIENT to ORDER-FORM	Insertion	george-4337	00:11:47	00:11:48	
Insert SERVER	Insertion	xrystos	00:11:57	00:12:26	
Insert Link (5) Label	Modification	george-4337	00:12:04	00:12:40	
Insert-delete ELEG	Failed Insertion	petros	00:12:30	00:13:00	
Insert DATA-BASE (11)	Insertion	george-4337	00:13:17	00:13:31	
Inser-Remove Link(6)	Failed Insertion	thodoris-xrystos	00:13:11	00:13:29	
ATTEMPT TO INSERT RELATION	Failed Insertion	thodoris	00:14:03	00:14:06	
Inser-Remove Link(10)	Failed Insertion	thodoris	00:14:17	00:14:37	

Figure 10: Extract of the action level interpretation of the collaborative problem solving activity

As a result of this process visualization of activity at the action level may be produced. An example of this is shown in figure 11. This has been produced by representing an action as a bar in a Gantt chart fashion in the time dimension (vertical view). The view included in fig. 11 corresponds to the extract of the actions of fig. 10.

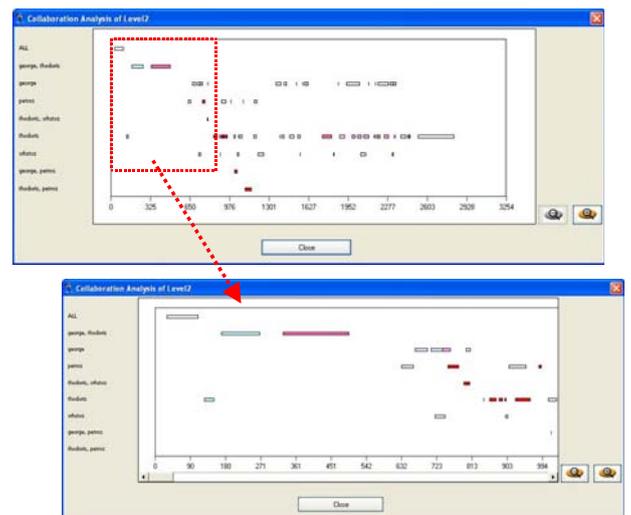


Figure 11. Diagrammatic representation of actions

The duration of the actions and the concurrency of activity of the group are shown through this view. Spells of inactivity appear, perhaps attributed to cognitive or preparatory mental activity. Also the actors that participate in actions are shown in this view. As an example of the activity of our study, during the initial phase (0-500 sec) there is a sequential collaborative action involving all actors. For a certain period, between 300 and 500 sec, an action takes place involving actors *George* and *Thodoris* without participation of the rest of the group, who presumably are observing the activity. Subsequently there is a period of autonomous actions of the actors with a degree of overlap (period 660-780 sec). This phase of activity was the result of the original negotiation, which resulted in a phase of individual experimentation with the concept mapping tool by the partners who attempted to introduce key concepts and relations before later on entering a new phase of negotiation of the externalized ideas. Some of the actions were clearly related to usability problems of the tool and misconceptions of its operation. For instance both users *Thodoris* and *Xrhstos* experienced difficulty with linking of concepts in the activity space, identified as actions of “*failed insertion of relation*” in fig. 10.

**Analysis of action level**

The action level of this multi-level view is particularly important, since through this view conscious goal-directed activity is described. In this view a sequence of individual or common goals are identified and tracking of their achievement through operations and mediating tools is identified. The analyst can move between level #1 and level #2 smoothly by identifying the means by which the action goals are achieved, as identified by the operations of level #1. The ColAT tool supports this by highlighting the operations corresponding to a selected action. Since goals can be hierarchically structured, we used the third level for representation of high level goals. However in a study of more complex activity, this third level is destined to be used for representation of webs of activities.

In our case the action view made evident and gave a quantitative representation of the collaboration strategy used in this group. The students first experimented with the tool and negotiated the specific problem to be solved (in this particular case they decided to build a concept map of an electronic bookshop). Building of the actual concept map involved consecutive phases of independent concurrent activity of group members and negotiation of the externalized ideas in the form of chunks of concept maps.

Generation of quantitative representations of the multilevel view of the activity is a straightforward process. For instance in fig. 12 the contribution of the partners according to the various levels of activity is included. From fig. 12, actors *Thodoris* and *George* seem to contribute

more significantly than *Xrhstos* and *Petros* in both the operations level and actions level. Actor *Thodoris* has even more prominent role in actions than in operations, since *Thodoris* participated in 50% of actions.

Actor	Operations Level	Actions Level
George	32%	27%
Xrhstos	12%	13%
Petros	11%	10%
Thodoris	44%	50%

Figure 12. Contribution of the group members in the collaborative activity

Observing fig. 12 we come to the conclusion that in this group of four non-coordinated distant partners (mediated by collaboration support tool involving textual and shared activity board interaction support), eventually a small kernel of actors plays the leading role, while the rest take secondary roles or just observe the activity. This conclusion is reached also by qualitative analysis of dialogue and interaction. This asymmetry may be attributed partly to the nature of the mediating tools, thus identifying a relation between the division of labor and tools for this particular case of activity.

**CONCLUSIONS**

The OCAF method presented in this paper constitutes an analytical framework that supports contextual studies of collaborative problem solving activities. OCAF implements many of the key concepts of Activity Theory: The unit of analysis is an activity, which is studied through an object-oriented view. The internalization-externalization process of objects is supported through views of spatial representation of the concepts that are subject of discussion and later take the form of tangible objects upon which operations are effected, through a refinement process. The activity is decomposed in a hierarchy of purposeful actions, which are effected through operations. The OCAF method supports analysis of data collected during ethnographic studies of various forms through which interpretation of the activity can take place. It has been used effectively for evaluation of IT design in the case of collaboration support groupware.

New innovative concepts of the OCAF method are the history and ownership of the objects, as well as the various views over the activity, supported by the tools that have been developed. A key concept is the unification of dialogue and action and the object oriented character of both, through the event analysis scheme. In the original OCAF method proposal, such a scheme was included, while since then other researchers have applied different analytical frameworks using the same method effectively,

(e.g. Voyiatzaki et al. 2004, Lavidas et al. 2004). A number of quantitative indices have been generated from the proposed OCAF model, like the collaboration factor, which produce a visual effect of the activity at run time, or can be used for analysis later on.

The contribution of the OCAF tools to interpretation of the activity using various views and levels of abstraction is substantial, since the tools are capable of reproducing the activity, either using the logfile in the case of Synergo or video and audio sources in ColAT.

OCAF is a simple method to apply, since it incorporates three steps supported by the provided tools:

- (a) Definition of the event analysis scheme, which can be based on a theoretical or empirical framework of the study.
- (b) Annotation of the observed events using this scheme, inspection and interpretation of the produced views of the activity in the time and space dimensions (density of activity, symmetry of interaction, annotated solution, etc.)
- (c) Finally building of a multilevel interpretation of the activity by assigning the recorded operations to purposeful actions and generation of quantitative views of them.

However, one current drawback of the OCAF approach is that it does not yet fully incorporate some of the more recently developed refinements of Activity Theory, relating to subject and object orientedness of collaborative activities, e.g. (Bedny & Karwowski, 2004). One specific concept that needs to be further developed is that of “object-orientedness” and “object-ownership”. Originally the term in OCAF was meant to be specific to the “world” of a shared distributed modelling software environment, while in this paper we have made an attempt to extend it to collaborative problem-solving in general. However in this wider context, the components of collaborative activity - subjects, motives, tools, objects, goals, results – need to be seen as functionally variant as their specific content may change with time. This implies that the term “object” needs to be considered as a functional label given to that which is being explored or manipulated through the actions of a subject, those actions being mediated by various mental or material tools, as proposed by the Systemic-Structural Theory of Activity, see (Harris 2004).

The proposed here method has been applied in various cases of analysis and evaluation of problem solving activity of collocated or distant groups in the frame of studies like usability evaluation of IT technology, understanding of collaborative learning process etc. It is the objective of future research to examine applicability of the framework in other cases, like asynchronous collaborative activities, larger groups like communities of practice, organizational structures etc.

## Acknowledgement

The development of the methods and tools described in this paper has been partly funded by the European Commission IST program, through the ModellingSpace project IST-2000-25385, by the Greek Ministry of Education Project ModelsCreator of Pinelopi program and by the European Social Fund (ESF) -Operational Program for Educational and Vocational Training II (EPEAEK II) Basic Research Program IRAKLEITOS. The tools Synergo and ColAT, presented here, are available for experimental use and can be downloaded from [www.ee.upatras.gr/hci/hcien](http://www.ee.upatras.gr/hci/hcien). Special thanks are also due to peer reviewers S. R. Harris and A. Rizzo for fruitful comments on earlier draft.

## REFERENCES

- Avouris N., Komis V., Margaritis M., Fidas C., (2004), ModellingSpace: A tool for synchronous collaborative problem solving , Proceedings Ed Media 2004, AACE Conf., Lugano, June 2004, pp. 381-386.
- Avouris N.M., Dimitracopoulou A., Komis V., Fidas C., (2002), OCAF: An object-oriented model of analysis of collaborative problem solving, G. Stahl (ed), Proceedings CSCS 2002, pp.92-101, Colorado, January 2002, Erlbaum Assoc. Hillsdale, NJ, 2002.
- Avouris N., M. Margaritis, V. Komis, (2003), Real-Time Collaborative Problem Solving: A Study on Alternative Coordination Mechanisms, Proc. of 3rd IEEE Intern. Conf. Advanced Learning Technology (ICALT), pp.86-90, Athens, July.
- Avouris N., M. Margaritis, V. Komis, (2004). The effect of group size in synchronous collaborative problem solving activities, Proc. ED Media 2004, AACE Conf., Lugano, June 2004, pp. 4303-4306.
- Avouris N., V. Komis, M. Margaritis, G. Fiotakis, (2004), An environment for studying collaborative learning activities, Journal of International Forum of Educational Technology & Society , Special Issue on Technology – Enhanced Learning, 7 (2), pp. 34-41, April 2004 .
- Avouris N.M., Dimitracopoulou A., Komis V., (2003), On analysis of collaborative problem solving: An object-oriented approach, Computers in Human Behavior, 19, (2), March 2003, pp. 147-167.
- Baker M., Hansen T., Joiner R., & Traum D. (1999). The role of grounding in collaborative problem solving tasks. In P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches. pp. 31-64, Advances in Learning and Instruction series, Pergamon, Elsevier .
- Bedny, G. Z., & Karwowski, W. (2004). A functional model of the human orienting activity. Theoretical Issues in Ergonomics Science, 5(4), 255-274.
- Bertelsen O.W., Bodker S., (2003), Activity Theory, in J. M Carroll (ed.), HCI Models, Theories and Frameworks, Morgan Kaufmann, 2003.

- Bodker S., (1996), Applying Activity Theory to Video Analysis: How to make sense of video data in Human-Computer Interaction, in Nardi B.A. (ed), Context and Consciousness, MIT Press 1996.
- Dillenbourg P. (ed.) (1999). Collaborative-learning: Cognitive and Computational Approaches. Advances in Learning and Instruction series, Pergamon, Elsevier.
- Dix A., Finlay J., Abowd G, Beale R., (1998), Human-Computer Interaction, Prentice Hall
- Fidas C., Komis V., Tzanavaris S., Avouris N., (2005), Heterogeneity of learning material in synchronous computer-supported collaborative modeling, *Computers & Education*, 44 (2), pp. 135-154, February 2005..
- Harris S.R., (2004), Systemic-structural activity analysis of video data: a practical guide, ATIT workshop proc, Copenhagen, September 2004.
- Kaptelinin V, Nardi S., Macauley C., (1999). The activity checklist: a tool for representing the 'space' of context, *ACM Interactions*, 6 (4), 27-39.
- Komis V., Avouris N., Fidas C., (2002), Computer-Supported Collaborative Concept Mapping: Study of Synchronous Peer Interaction, *Education and Information Technologies*, 7:2, 169–188.
- Komis V., Avouris N., Fidas C., (2003), A study on heterogeneity during real-time collaborative problem solving, In B. Wasson, S. Ludvigsen, U. Hoppe (eds.), *Designing for Change in Networked Learning Environments*, Proc. CSCL 2003, pp. 411-420, Kluwer Academic Publ., Dordrecht, 2003.
- Korpela, M., Soriyan, H. A. and Olufokunbi, K. C. (2000) Activity Analysis as a Method for Information Systems Development. *Scandinavian Journal of Information Systems*, 12, 191.
- Kuutti K. (1996). Activity Theory as a Potential Framework for Human-Computer Interaction Research. In Nardi B. (edited by) Context and Consciousness. Activity Theory and Human-Computer Interaction. Cambridge, Massachusetts: The MIT Press.
- Lavidas K., Komis V., Avouris N., (2004). Study of interaction in learning environment using modeling educational software, Proc. ETPE 2004, September 2004, Athens.
- Margaritis M., Avouris N., Komis V., (2004), Methods and Tools for representation of Collaborative Learning activities. Proc. ETPE 2004, September 2004, Athens.
- Mwanza, D., (2001) "Where Theory meets Practice: A Case for an Activity Theory based Methodology to guide Computer System Design." In Hirose, M. (Ed), *Proceedings of INTERACT'2001*, Tokyo, Japan, July, 2001. IOS Press Oxford, UK.
- Roschelle J. & Teasley S.D. (1995). The construction of shared knowledge in collaborative problem solving. In C.E. O'Malley (Ed), *Computer Supported Collaborative Learning*, (pp. 69-197) Berlin: Springer-Verlag.
- Searle, J. R. (1975). A taxonomy of illocutionary acts. In K. Gunderson (Ed.), *Language, Mind and Knowledge*, 344-369. Minneapolis: University of Minnesota Press.
- Tselios N. K. and Avouris N. M., (2003), Cognitive Task Modeling for system design and evaluation of nonroutine task domains, in E. Hollnagen (ed.) *Handbook of Cognitive Task Design*, Lawrence Erlbaum Associates, pp. 307-332.
- Van Welie M., G. Van der Veer (2003), Groupware Task Analysis, in E. Hollnagen, Hollnagen E. (ed.), *Handbook of Cognitive Task Design*, Lawrence Erlbaum.
- Voyiatzaki E., Christakoudis C., Margaritis M., Avouris N., (2004), Algorithms Teaching in Secondary Education: A collaborative Approach, Proc. ED- Media 2004, Lugano, June 2004, pp. 2781-2789.
- Winograd T., (1987). A Language/Action Perspective on the Design of Cooperative Work, *Human-Computer Interaction* 3:1 (1987-88), 3-30.

# Activity Walkthrough - a quick user interface evaluation without users

Olav W. Bertelsen

University of Aarhus, Department of Computer Science,  
IT-Parken, Åbogade 34, DK-8200 Århus N.  
olavb@daimi.au.dk

## ABSTRACT

Based on activity theory an expert review method, the activity walkthrough, is introduced. The method is a modified version of the cognitive walkthrough, addressing some of the practical issues arising when non-experts apply the cognitive walkthrough to non-trivial interfaces. The presented version of the activity walkthrough is work in progress. Initial results from student experiments are reported to show that the procedure needs to be explained better and made simpler.

## INTRODUCTION

In this paper I introduce a method for analytical evaluation of user interfaces, i.e. a method by which, one person (possibly a member of the design team) can assess the use qualities of a new interface, based on only a rough specification. The specification can be made in a suitable formalism or it can be a paper- or computer-based prototype. The method does not require real users to be involved in the evaluation. The person doing the evaluation can be one of the designers, a person from the developments organizations usability section, or a person from an independent sub-contractor.

Ideally, HCI provides methods that can be applied easily by engineers and systems designers, to ensure that measurement of and concern for the use situation is brought into the design process (Card et al. 1983). At the same time, conceptually rich approaches, like the human activity framework (Kuutti 1996; Bertelsen & Bødker 2003), have been proposed as alternatives providing a fuller background for designing systems for the real world. However, it appears to be difficult to commit these alternative approaches to the production of engineering methods because they are based on the tenet that the design problem is too complex to be solved at the back of an envelope, and that it is fundamentally impossible to formalize human behavior. Therefore this paper also aims to set an example for how human activity based IT research can formulate engineering methods.

The cognitive walkthrough (Lewis et al. 1997), hereafter CW, is a popular theory-based method that is readily applicable for practical assessment of a design specification without involving real users in the assessment. The CW is based on a theory of exploratory learning, but the use of the

method does not require that the evaluator is knowledgeable in that theoretical framework.

The CW is used for assessing a new system at a very early stage, possibly (and preferably) at a point in the design cycle when only a specification or a mock-up exists. The method is analytical in the sense that it is done by the evaluator(s) alone without involving users or other test persons. The evaluator starts by identifying a number of essential tasks to be supported by the new software. Each task is then broken down into a sequence of simple components, e.g. keystrokes. Then, each task sequence is "walked through" in order to assess if it is likely that the intended users will be able understand what to do to complete the task. For each simple component in the "walked through" sequence, the evaluator asks three questions. Is the correct action visible to the user? Is the user able to connect the correct action to the desired outcome? Will the user notice that progress has been made? The negative answers, to any of these questions indicate possible usability problems that can be dealt with based on the explanation given in the answer to the problematic step in the sequence. All answers, both positive and negative are recorded, though, because positive answers can be based on false assumptions and because they can serve as inspiration for solutions to identified problems. The CW does not help the evaluator assess the efficiency of a system, but primarily helps in assessing whether the system is usable for users using the system for the first time, or only casually.

However, sometimes following the steps in the CW is not simple. In such situations, the CW does not help the inspector get insights into what is understandable and how things make sense from the users' point of view. Thus, the simple question about visibility may be difficult to answer without detailed knowledge about how users interpret what they see. Understanding this interpretation is important and not well supported by the method. The CW is based on the assumption that perception and action are separated. In contrast, activity theory assumes the unity of consciousness and activity - what users are able to see depends on what they want to do. The basic problem with the CW is the absence of the real context of interaction. Based on activity theory (Bertelsen & Bødker op. cit.) it is possible to discuss these difficulties in further detail and to point to a possible

solution retaining some of the efficiency of the CW and at the same time provide more systematic help for the inspector. The primary problem is that the task analysis is "hypothetical" in the way that it is broken down based on the sequence of machine operations required to complete the task. According to activity theory the basic unit of analysis is activity, i.e. the level of human conduct that is motivated and directed to human needs. Activity is realized through conscious goal-directed actions that in turn are realized through automated operations that are not in the users conscious focus. Thus, the activity perspective takes motivated human activity as a meaningful unit of analysis rather than sequences of machine operations (for more details on activity theory in the context of IT use and design see e.g. Bertelsen & Bødker 2003, Bødker 1991, Kuutti 1996).

The important difference from the way a task is broken down into machine operations in the CW is that the division between actions and operations is not stable in activity theory. Actions become operations through learning and operations can become actions again if the conditions change. Thus, the way an action is realized through operations depends on the users' repertoire of operations, the conditions in the environment, the structure of the action and possibly the activity the action is realizing. This leads to two questions supplementing the CW procedure: Firstly, do the typical tasks correspond to purposeful actions realizing users activities? Secondly, do the machine operations trigger operations in the users repertoire?

Because these questions can only be fully answered through empirical investigations may lead to refusal of the CW. While this would be theoretically valid, it is not very practical. Practical situations may call for quick assessments without involving real users. Therefore, I will briefly outline an activity theory based walkthrough.

#### **ACTIVITY WALKTHROUGH**

Activity Walkthrough (AW) is a method, or procedure, for evaluation of an interactive system at a point in the design process when the system has not yet been implemented.

Like with the CW, the AW does not involve real users in the assessment. Instead the evaluator makes the assessment in an analytical manner "at the desk". Therefore, the quality of the assessment depends largely on the earlier phases of the design cycle, in particular user studies and requirements analysis.

One or two evaluators who can be members of the design team or specialists recruited outside can perform the AW. It is likely that the evaluator has to have a prior understanding of basic activity theory concepts.

The strength of the AW is that it is cheaper than assessment procedures involving real users and an implemented system. The weakness is, clearly, that it is then a bit more "hypothetical", however, not as much so as the CW (see procedural description below and concluding discussion).

The application area of the AW includes the kind of "walk up and use"-like systems that the CW is targeted at, but because it also addresses the learning aspects, and because it, it must be expected that it will be useful for assessing a much broader range of systems.

#### **The procedure**

In the following sections the activity walkthrough is outlined. Each step of the procedure is illustrated with a railway ticket vending machine, as the recurring example. This example is chosen for its simplicity but may suffer from being too simple in relation to some aspects of the walkthrough. The examples are given in *italics*.

#### **First phase: preparation**

In preparing the activity walkthrough, the inspector identifies the typical tasks to analyze, based on the requirements specification. This identification of typical tasks is similar to the preparation of a CW. In some cases this identification is problematic, in other cases it is carefully considered in the requirements specification and in early analysis.

*In the case of the ticket machine the typical (and only) tasks are variations of the purchase of tickets.*

#### **Second phase: contextualization**

Human users are oriented to activities and actions, not the tasks defined by the system (Bertelsen & Bødker 2003). In the case of the ticket machine users are not oriented to the operation of the machine but to the travel or to getting the necessary travel document. The activity walkthrough conceptually situates the application in the context of use by identifying users and activities in which the typical tasks are supposed to become embedded.

The procedure for this contextualization is outlined below. It should be emphasized that the economy of the activity walkthrough depends on that early analysis has gathered enough information to produce the contextualization. A sensible balance between what is available and what is needed should determine the degree of detail in the contextualization.

- Identify the activities in which the application mediates purposeful actions. Who use the application? What is the overall motivated (or need oriented) behavioral unit that the use of the application contributes to the realization of?

*The elderly couple using the ticket machine may be oriented to the visit they are going to pay their grand children; as opposed to merely buying a ticket or traveling to another town.*

For each activity

- Identify the actions through the application that contribute to realizing the activities, and the objects or outcomes that these actions are directed to.

*The elderly couple is oriented to getting the correct ticket when using the ticket machine (but the backdrop that this action should be analyzed at is the motive of the activity).*

- Consider other ways of realizing the activity without the application, e.g. earlier historical generations of the activity.

*Earlier, the elderly couple would have bought their travel document in the ticket office, or maybe they could have a special season ticket for retired people, or they could have been driving by their own motorcar.*

- Consider other artefacts, than the application in question, mediating the activity - focusing on the part of the activity where the application is going to be used.

*The elderly couple use various means for planning the travel, e.g. time table on the Internet, or telephone calls to the train station.*

- Consider the users horizon of expectation, i.e. their experience with using similar applications or tools.

*The elderly couple occasionally uses the ticket machines in the busses in the town where they live.*

For the application as such mediating in all the activities:

- Consider the application as being a mediator between various activities by situating it in a web of activities where it is used, and e.g. analyzing contradictions or tensions between these activities.

*The ticket vending machine mediates between the traveling couple and the public transportation system, thus the machine should not only be usable but also be convey information about, e.g. the price structure.*

- Consider the historical development of the web of activities and the historical predecessors of the application.

*Historically, price structure was concealed behind the ticket counter so the public did not have to deal with it.*

These two later points are not very relevant in the case of the ticket machine but could be of outmost importance for work-oriented systems where collaboration is an important aspect.

### **Third phase: verification of tasks**

Based on the contextualization of the application the inspector assesses to which extend each task corresponds to purposeful actions in the activities in which the application is going to be embedded. If the early design has been done in a proper way there will be a high degree of correspondence, and if it has not it is not very likely that formative inspection will solve the problem.

*In the case of the ticket machine, it may turn out that the task of buying a ticket between the actual station and*

*another town is not relevant because the elderly couple will arrive to the train station by bus and therefore need an extension to their bus ticket instead of regular ticket.*

### **Fourth phase: task analysis**

In this phase a number of tasks, that the coming computer-based system is going to support, are identified. This is done based on the requirements specification and the early user studies, but it does not at this point involve empirical data gathered from actual use of the new system. In this sense, the tasks analyzed are purely hypothetical.

The task analysis is carried out by breaking each task down into a sequence of atomic operations at the interface, just as it is done in the CW. This analysis should be done without taking the findings of the second and third phases into account. However, it should be noted if discrepancies between the task analysis and these insights are discovered already at this point.

In short, the task analysis is done from the point of view of the interface, and not from the point of view of the users activity. This is needed because it is the only practical way to generate a sequence without access to empirical data.

*In the case of the ticket machine the tasks will be broken down into sequences of key presses, money insertion and the collection of ticket(s) and change.*

### **Fifth phase: walkthrough**

The walkthrough is carried out for each task in turn for each of the activities unless it is not relevant.

*We do not have to make walkthroughs of maintenance tasks for the elderly couple.*

For each step in the task analysis ask the following questions. The insights generated in phases two and three are important resources.

**Q0:** what is the next step in the task analysis?

In CW the question is phrased "what does the user want to do?" but since the sequence of machine operations is not necessarily making sense as purposeful action for the user the more neutral formulation is used here.

*The next step in the purchase of a ticket could be that money is inserted into the vending machine.*

**Q1:** The first question is composed of three sub questions, which will be answered in parallel (i.e., iteratively). The sub questions are interdependent because it is not possible to separate perception from ongoing action. What is visible depends on what the user is doing. The three questions are partly redundant which helps the inspector making the analysis from the point of view of the user.

**Q1A:** Match at the level of purposeful action. Does the required machine operation make sense in the context of the users purposeful action as a step towards the goal?

*Inserting money into the ticket vending machine makes sense in the course of the elderly couples ticket purchase.*

**Q1B:** Is the correct machine operation perceivable in the users horizon of expectation in the purposeful action? Is it immediately visible or will it be obvious to the user what to do.

*The slot for insertion of a banknote is not perceivable as such for the elderly couple because vending machines in their experience always take coins.*

**Q1C:** Match at the level of operations. Will the appearance of the interface, in the structure of action, condition (or trigger) relevant established operations in the user, enabling the activation of the correct machine operation? This question may be answered in terms of (natural and canonical) affordances (Bærentsen & Trettvik 2002).

*Activation by a push button on the ticket vending machine panel is a canonical affordance, triggering the elderly couples operations. In contrast activation via a touch screen is to be learned explicitly.*

**Q2:** Will the system response match users horizon of expectation in a way so that makes it clear that progress has been made? Users may expect explicit confirmation of progress or they may expect only to get explicit response when something is wrong.

*If the elderly couple inserts the banknote in the slot they will see that the note is "eaten" by the machine, this may not convince the couple that they have made progress before they also see that the amount to insert decreases in the display.*

**Q3:** Consider the three levels of transparency and learnability.

- Will the machine operation trigger established operations in the user?
- Will the user be able to develop matching actions or operations in the situation? Does the interface support the development of new operations if appropriate operations are not established or sufficiently developed? (Bardram & Bertelsen 1995).
- Will the user need instruction to get to use the application?

This question may be addressed for each step in the task or it may be addressed for each task in its entity.

*Inserting bank notes into a slot is unlikely to be an established operation. Supported by the illustrations on the machine, this new operation can be developed through reconceptualization of the coin insertion operation. The instructions printed on the front of the machine may be sufficient, but for some getting to use the ticket vending machine requires more personal instruction.*

### **Sixth phase: Task analysis verification**

Finally, the task analysis is reviewed critically based on the walkthrough. Special attention is directed to how well the sequence of machine operations matches the users operations and actions, and the consistent flow of operations throughout the task is considered. This phase is needed to remedy the fragmentation introduced with the task analysis. It depends on the understanding of how the supported actions are embedded in activity as it is established in the contextualization, which is itself a top down view.

*In the case of the ticket vending machine there is a conflict between the elderly couples experience with inserting coins before specifying purchase and their experience of negotiating the type of ticket with the sales person before handing over the money.*

### **REPORTING AND FOLLOW-UP**

Upon the completion of the six phases a summary report, including design recommendations are produced. The Summary report should at least contain the following:

- A brief summary of the contextualization.
- A summary of the task verification, taking into account how each task matches each relevant activity.
- Specific problems identified in the walkthrough.
- A record of problems concerning the flow of machine operations in the structure of the tasks.

The design recommendations should detail solutions to the identified problems considering the importance of the problem from the point of view of the relevant activities and the const of the solution.

*The analysis of the ticket vending machine will detail a range of uses and their historical and actual contexts, including commuters and occasional users. For the elderly couple the lack of initial familiarity with advanced vending machines combined with the complexity of the ticket price structure in this specific county means that it is unlikely that interface tweaking alone can improve the conditions for successful use. This may, however be considered less important, because they may prefer to buy their tickets in the tickets office anyway, or via telephone. The flow of action is important in many respects. For the elderly couple, selecting before paying could be debatable, and more importantly ticket purchase at the platform could turn out to be the wrong time.*

### **FIRST EXPERIMENTS WITH APPLYING THE METHOD**

The Activity Walkthrough has been tested in an informal way with students participating in the authors' HCI course. The purpose of this preliminary experiment was to get information about the applicability and usefulness of the method and to get ideas for further development.

In an obligatory assignment the students, in groups of three, evaluated a user interface of their own choice. Empirical as well as analytical evaluation was required, and the empirical data were to be analyzed through focus shift analysis (Bertelsen & Bødker 2003). With respect to the analytical part the students could freely choose methods they knew. The students had been introduced to activity theory in general and were working practically with the focus shift analysis. The Activity walkthrough was introduced by an earlier version of this paper. Four out of ten groups chose to apply the activity walkthrough.

It was not easy for the students to get good results by using the activity walkthrough. The main problem, as noted by one of the groups, was that they did not spend enough effort on contextualisation. It seems likely from their reports that the main problem for them was to do the inspection outside the context of a development project. They did not have a requirements specification, therefore phase one was hard to complete. Similarly, the added realism and completeness introduced in the contextualization had to be built from scratch and therefore the students seemed to have reasoned that it was not worth the effort. One group, evaluating a digital camera, realized too late in the process that they needed a more complete contextualization.

On this background it is no surprise that the walkthrough to a large extent degenerated to become very similar to a traditional CW. Thus one group ended up mixing Q1A, Q1B and Q1C together thereby losing some of the intended analytical power.

In summary, the first experiment indicates that the activity walkthrough in the form presented here is too complicated to be used by students for a small assignment. On the other hand the experiment does not point to specific reservations to be taken with respect to the industrial application of the method, on the contrary several of the students concluded that the contextualization in phase is an important advancement over state of the art inspection methods.

## DISCUSSION

It has been demonstrated that it is indeed possible to modify the cognitive walkthrough to take advantage of the general insights yielded by the application of activity theory to HCI. The Activity Walkthrough provides a way of including more "context" into an expert review without having to include the whole wide world. It seems that this new method has potentials for in a future version striking a manageable balance between extreme simplicity, placing the complexity outside the review method, and excessive inclusion of all relevant aspects, making the method practically useless. It has not been shown empirically, however, that the method is applicable in practice. The

experiments with the student were mainly negative, whereas Steve Harris reported during the workshop that he together with a colleague had used the method with good results. Future work will include a further modification of the procedure as well as a more systematic testing in an industrial setting.

Theoretically, the AW challenges basic assumptions of most AT based IT research by loosening up the amount of specificity involved in analysis at the level of activity. As illustrated with the recurring example of the ticket-wending machine, it may be too complicated and possibly not useful to take actual activity into account. I will suggest that, even though it may be impossible to take the real activity into account, knowing that the purposeful actions are embedded in activity, and maybe outlining hypothetical activities will make the analysis at the level of action better.

## ACKNOWLEDGMENTS

Thanks to numerous colleagues and students, including Klaus Bærentsen, Susanne Bødker and Mikkel Godsk. Also great thanks to the participants in the workshop.

## REFERENCES

- Bardram, J. E., Bertelsen, O. W. (1995). Supporting the Development of Transparent Interaction. In Blumenthal, Gornostayev, & Unger (eds.). *EWHCI '95, Selected Papers*. Berlin: Springer Verlag, 79-90
- Bertelsen, O.W., Bødker, S. (2003). Activity Theory. In Carroll, J.M. (ed.). *HCI Models, Theories, and Frameworks*. Morgan Kaufman Publishers. 291-324.
- Bertelsen, O. W. (2004). The Activity Walkthrough: an expert review method based on activity theory. In *Proc NordiCHI 2004*. 251-254.
- Bærentsen, K. B. & J. Tretvik (2002): An Activity Theory Approach to Affordance. In *Proc NordiCHI 2002*. ACM Press. 51-60.
- Bødker, S. (1991). *Through the Interface – a Human Activity Approach to User Interface Design*. Hillsdale, NJ: Lawrence Erlbaum Assoc.
- Card, S. K., Moran, T. P., Newell, A (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Assoc.
- Kuutti, K. (1996). Activity Theory as a Potential Framework for Human-Computer Interaction Research. In Nardi, B. (eds.), *Context and Consciousness*, MIT Press, Cambridge, 17-44.
- Lewis C., Wharton, C. (1997). Cognitive Walkthroughs, In Helander, Landauer, Prabhu (eds.) *Handbook of Human-Computer Interaction, 2nd, compl. rev. ed.* Amsterdam: Elsevier/North Holland, 717-732.

# Activity Theory applied to Requirements Elicitation of Multi-Agent Systems

Rubén Fuentes, Jorge Gómez, Juan Pavón  
Dep. Sistemas Informáticos y Programación  
Universidad Complutense Madrid  
c/ Juan del Rosal 8, 28040 Madrid, Spain  
{ruben, jggomez, jpavon}@sip.ucm.es  
<http://grasia.fdi.ucm.es>

Lorna Uden  
Faculty of Computing, Engineering, and Technology  
Staffordshire University  
Beaconside, Stafford, ST18 OAD. UK  
l.uden@staffs.ac.uk

## ABSTRACT

Most of current Multi-Agent System (MAS) methodologies do not provide specific techniques for requirements elicitation. They usually rely on the same approach used in analysis and design. However, requirements elicitation demands a deep understanding of users' environment and their needs, which are not software artefacts. We believe that those techniques, oriented to software developers and their artefacts, have to be complemented with other techniques more oriented to common users and their own abstractions. In this paper, we propose using principles from Activity Theory (AT) to provide this additional support. AT provides us the theoretical framework for analyzing the social and cultural issues in the requirements of MAS. However, the use of AT is too abstract. To overcome this abstraction, we use a tool developed by Kaptelinin and others (1999) that makes concrete the conceptual system of the AT for the task of requirements elicitation: the Activity Checklist. The checklist allows us to consider several dimensions about the intended system and its surroundings: temporal, social, and spatial. It includes questions, which are based on the experience obtained in the study of human behaviours and activities, to elucidate the relevant information related to these dimensions. The modified Activity Checklist used here is embedded in a MAS software process supported by AT, as an assistant tool for making decisions about system requirements. This paper illustrates the approach through several examples of the development and use of the new Activity Checklist for MAS.

## Categories and Subject Descriptors

D.2.1 [Software Engineering]: Requirements/Specifications.

## General Terms

Design, Human Factors, Languages, Verification.

## Keywords

Multi-Agent Systems Development, Activity Theory, Activity Checklist, Requirements Elicitation.

## 1. INTRODUCTION

The aim of the software development process is to provide a path from customer requirements to the implementation of a system that satisfies them. However, the main problem of current software development methodologies, including agent-oriented methodologies, is that they are too system-centered (Lamsweerde 2000, Magne & Sindre 1993, Zave & Jackson 1997). The distinction between workflows in the process depends on the difference between “what” the system should do and “how” it should do it. The environment, defined as the real world outside the system, is not often considered with enough concerns. According to (Ross & Schoman 1977), requirements are a careful assessment of the needs that a system is to fulfil. Requirements must say why a system is needed, based on current or foreseen conditions, which may be internal operations or an external market. They must address the customers' goals why a software system is needed, the functionalities the system software has to accomplish to achieve those goals, and the constraints about its development. In order to avoid an implementation bias in system development, which should be delayed to later stages, all the statements about requirements should be done in terms of the environment (Zave & Jackson 1997).

The environment, as that external real world of a software system, is usually a human activity system (Nuseibeh & Easterbrook 2000). Consequently, Requirements Engineering needs to be sensitive to human behaviours, motivations, and interactions. Currently, most of the techniques and methods used for elicitation do not originate from computer science, but areas such as group-interaction research, organizational theory, the social sciences, ethno methodology and sociolinguistics (McGrath & Uden 2000). Therefore, to elicit requirements, it is often necessary to resort to Psychology and Sociology to provide both theoretical background and practical techniques. In this context, Activity Theory (AT) would appear to have much to offer. AT incorporates notions of intentionality, history, mediation, motivation, upstanding, culture, and community.

AT is well suited for these human environmental aspects for Multi-Agent Systems (MAS) analysis and design as shown in (Fuentes et al. 2003 a, Fuentes et al. 2003 b). In addition, AT management of contradictions is a powerful tool to guide and validate the development process. From the Requirements Engineering point of view, the use of the AT abstractions eases elicitation by bridging the communicative gap between customers and developers (Fjeld et al. 2002), allowing the tracing of the whole development process, and making possible an early work with contradictions (Fuentes et al. 2003 a).

To gain a full coverage of the software lifecycle with AT, we propose in this paper the use of the Activity Checklist (Kaptelinin et al. 1999) as a tool to elicit and model requirements. The benefit of the checklist lies in its relatedness to problems raised by designers in their uses. The checklist comprises meaningful questions that can be used by designers to help them in identifying their solutions.

In this paper we describe a modified version of the Activity Checklist to focus on requirements elicitation (Hilburn et al. 1999, Zave 1997) for MAS. The approach provides a sociologically based method to guide the acquisition of knowledge about the environment and requirements of the envisioned MAS. Our version of the Activity Checklist is formalized using an UML (OMG 2003) language to describe concepts from AT (Fuentes et al. 2003 a) with some additions. The representation of requirements as AT concepts allows us to work with the remaining stages of the software process (as described in (Fuentes et al. 2003 b)) based on common abstractions of AT.

The following sections are organized as follows. The next section discusses the reasons to use specific tools for requirements elicitation and the particular features of MAS. This is followed by an overview of the Activity Checklist in section three. Then, we describe the needed changes to make the Activity Checklist a tool for modelling requirements of MAS and how to carry out them. Some examples of the use of the modified checklist depicted with UML are also given. The paper concludes with recommendations about the applicability of the Activity Checklist for MAS requirements elicitation.

## 2. REQUIREMENTS ELICITATION WITH A SOCIAL THEORY

Requirements elicitation is a key to the success of the development of software systems such as MAS. Goguen and Linde (1993) have provided a comprehensive survey of techniques for requirements elicitation, focusing on how these techniques can deal with the social aspects of this activity. They raise the important concept of social order in requirements elicitation and conclude that the requirements elicitation problem is fundamentally social and, thus, unsolvable if we use methods that are based entirely around individual cognition and ignore organisational requirements. Current models could not provide a theoretical basis for understanding “regularly patterned” human activity (Probert 1999). In order to overcome the above mentioned problems, it is necessary to have a methodology and tools that can support the continuous evaluation of a statement of requirements as these evolve against a highly complex and dynamic problem situation. What is needed is to shift the focus from fixed and final requirements to those of a more dynamic nature. In particular, it is necessary to consider human information which, in social terms, does not have a physical reality and is not objective like physical information. Instead, it is based on individual, group, or organisational needs. Such information informs action in organisations and is thus closely related to organisational activity and organisational structure.

One reconceptualisation of human information that allows for social organisation processes is AT (McGrath & Uden 2000). AT has increasingly being suggested by researchers in recent years as an ideal candidate for system design including MAS. We think

that AT can be used as a framework for understanding the totality of human work and praxis and the deliberate processes changing this, i.e. a totality encompassing organisational development, design and use of computer artefacts (Bødker 1991). Also in AT, conflicts can be acknowledged and taken seriously in design.

For these reasons, we also believe that AT provides a conceptual framework to analyze human activities for MAS. The common features of human organizations and MAS, both they are social and intentional systems (Demazeau 1995, Maes 1994, Newell 1982, Sykara 1998), allow the application of AT to the development of MAS (Fuentes et al. 2003 a, Fuentes et al. 2003 b). So AT can be used as a framework to capture the requirements of MAS.

## 3. A BRIEF OVERVIEW OF THE ACTIVITY CHECKLIST

The Activity Checklist (Kaptelinin et al. 1999) is an analytical tool applied mainly in the field of Human-Computer Interaction (HCI). It is a guideline to model the impact and interaction of new technologies in human activities in terms of concepts from AT. The term technology is considered in a broad sense and it can be a software system, a technique, a process, or a tool. An example of its application can be found in (Gould et al. 2000), where the checklist is used to determine the needs of interaction in a web based information system.

The checklist can guide designers to specific areas of importance when trying to understand context. There are two versions of the checklist: one for design, which is intended for the early requirements, and other for the evaluation of existing systems.

The structure of the checklist reflects the five basic principles of the AT according to (Kaptelinin et al. 1999): object-orientedness; hierarchical structure of the activity; internalization and externalization; mediation of tools; development of the activity and its related elements. Based on these concepts of the AT and the experience of former studies, the checklist defines *areas*, *aspects*, and *questions*. The *areas* are connected with the focuses of interest in an activity according to the AT. Each *area* has a description of its intended meaning and includes different views, the *aspects*, of a social activity: past, current state, evolution, learning... Knowledge in these *aspects* describes the environment as seen by users. In order to collect information about the *aspects*, the Activity Checklist defines *questions* related to these features. The *questions* are written in natural language using concepts understandable for both customers and developers. They show users the contextual design space that represents the key *areas* of context specified by AT. That information collected with the *questions* enables the application of AT to a given case study. With the answers to the *questions* developers can elicit users' knowledge based on the theoretical principles of AT. This knowledge can be then translated to concrete properties of the system under study by researchers. Instances of *areas*, *aspects*, and *questions* of the Activity Checklist are shown in Table 1.

The four main *areas* covered by the checklist are:

- **Means/ends.** Relationships between users' goals and the system. It analyses the extent to which the technology facilitates and constrains the attainment of users' goals. It also considers the influence of the technology on the

contradictions of the organization, i.e. its impact on provoking or resolving conflicts between different artefacts (e.g. objectives, community and objectives, or tools).

- **Social and physical aspects of the environment.** Integration of the technology with requirements, *tools, resources, divisions of labour* and social rules of the environment.
- **Learning, cognition, and articulation.** This *area* considers how the target technology provides support to organize the *activity* and the acquisition of knowledge about it and the system. It also contrasts the internal components of the *activity* versus external ones and considers the support of their mutual transformations with target technology.
- **Development.** How the use of the technology has changed the previous form of the *activity* and the foreseen evolution of the current form.

ethnographic study about the related activities. That is why the checklist needs to be adapted for MAS.

#### 4. TAILORING THE ACTIVITY CHECKLIST FOR MAS

In order to apply the Activity Checklist to MAS development and integrate it in the software processes supported by AT, several goals have to be accomplished. The first one refers to changes in the focus of the original checklist, the second one is related with the way in which it is integrated with the remaining software stages, and the final one concerns the ease of use.

To meet the first goal, the Activity Checklist has to be transformed throughout two orthogonal axes: its focus and the level of its detail. The checklist therefore has to switch from HCI aspects of systems to those relevant in MAS. This shift does not involve a completely new checklist. The *areas* in the original version are still relevant for a MAS development. However, their *aspects* have to be reconsidered to adapt them to the specific characteristics of MAS design, such as workflows, coordination, or mental states of agents. The second axis of change is the level of detail. The Activity Checklist is more concrete than the general AT, but it is still too far from the level of detail about systems required to begin an analysis. To solve this problem, the *areas* of introspection and their related *questions* are grounded to more detailed forms that allow developers to gain a fine-grained vision of systems and their environments. The modified checklist would need to consider *questions* about elements such as tasks in workflows, their inputs and outputs, interactions, or roles of involved agents.

The second goal is to provide mechanisms to connect the Activity Checklist to a software process supported by AT. AT supports the MAS process by providing a language based in social abstractions to communicate agent concepts to users, and managing contradictions in development as a guide for system evolution. To use these techniques, developers are provided with mappings from MAS concepts to those of AT. As AT concepts are also present in the Activity Checklist, they can be regarded as the nexus throughout the whole process of MAS development. Then, to make the connection of requirements with other stages, developers would just require the initial correspondences from sample *areas* or *questions* of the checklist to concepts of AT. This step requires a good knowledge about AT and the checklist, but once done it can be used through different projects. With these correspondences, answers to *questions* in the checklist can be put in terms of AT concepts with a language close to UML (OMG 2003). AT concepts in this form can be then easily translated (Fuentes et al. 2003 a) to agent concepts in a given methodology. Table 2 shows an example of mappings between AT concepts and agent ones with the INGENIAS agent-oriented methodology (Gómez & Pavón 2003).

An additional advantage of mapping *questions* to specific patterns of AT concepts is that it allows cross-references between *questions* in the requirements elicitation. Patterns act as templates filled up with information coming from the answers to *questions*. Those templates allow that some or all of the information searched by a *question* to be obtained from other *question* templates. In this way, customers do not need to answer every

DESIGN VERSION			
USE	Areas	Aspects	Sample questions
	Means/ ends	Alternative ways to attain target goals through lower-level goals.	Is it necessary for the user to constantly switch between different actions or activities? If yes, are there "emergency exits" which support painless transition between actions and activities, and, if necessary, returning to previous states, actions, or activities?
	Environment	Role of existing technology in producing the outcomes of target actions.	Is target technology integrated with other tools and materials?
	Learning/ cognition/ articulation	Self-monitoring and reflection through externalization.	Is externally distributed knowledge easily accessible when necessary?
		Possibilities for simulating target actions before their actual implementation.	Does the system provide representations of user's activities that can help in goal setting and self-evaluation?
Development	Anticipated changes of target actions after new technology is implemented.	Are users' attitudes toward the system becoming more or less positive?	
		Are there negative or positive side-effects associated with the use of the system?	

**Table 1.** Examples of *areas, aspects,* and *questions* extracted from the design version of the Activity Checklist

Although the Activity Checklist is intended to make concrete the theoretical framework of AT, it is not directly applicable to MAS development because it is biased towards its applications in HCI. It focuses too much in topics that are not cardinal to general MAS, such as adaptation of users to the system or the

*question* or fill every concept, because some of this information can be obtained from other sources.

Activity Theory	INGENIAS
Activity	Task, Workflow, Interaction
Subject	Agent, Role
Object	Resource, Application, Fact, Believe
Outcome	Resource, Application, Fact, Believe
Objective	Goal, Mental state
Tool	Resource, Application
Community	Group, Organization
Rules	Mental states, Autonomous entities query
Division of Labour	Mental states, Autonomous entities query
activity → <i>accomplished</i> by → subject	– Agent Model agent(subject) → <i>GTPursues</i> → goal(objective)
activity → <i>satisfy</i> → objective	– Tasks & Goals Model agent(subject) → <i>WFResponsible</i> → task(activity)

**Table 2.** Some mappings between AT and INGENIAS. Note that correspondences between AT concepts and INGENIAS ones are not unique. Relationships appear in *italics*

The third goal is concerned with helping both customers and developers to understand *questions* and answer them. The modified checklist is not only intended for AT practitioners, it tries to be also a tool for people with a shallow knowledge of AT. To provide the required assistance for them, these *questions* have associated sets of previous responses in other domains. These examples of answers provide the team additional knowledge about the searched information or its relationships.

The usability of the method is increased by the use of a graphical notation for AT (Fuentes et al. 2003 a). The graphical notation also helps communication through the development process.

Our proposal shares many similarities with others in the field of requirements elicitation, such as *i\** (Yu 1997) and KAOS (Dardenne et al. 1993). These approaches also use a graphical representation for requirements. However, gathering requirements in *i\** and KAOS is merely based on the agent paradigm as a software conceptualization and its common abstractions. On the contrary, our approach focuses on the sociological aspect of the MAS from a human organizational point of view. Our approach also provides a set of *questions* based on AT to ease the introspection in the intended environment, with both a textual and

a graphical representation. This double representation addresses two aims of requirements, as account of users' knowledge and needs and as departure point for the software process. The following section describes how we modify the checklist to adapt it for use in requirements elicitation for MAS inside a software development process supported by AT (Fuentes et al. 2003 b).

These goals do not specify a unique formulation of the new Activity Checklist. Although we have developed a version for generic purposes, additional information about the envisioned system and its context could be obtained using domain specific versions. For example, this could be the case for retrieval information, telecommunications, manufacturing, or personal assistants. In order to give developers a guide to make these modifications, the next section specifies the method to build a new checklist.

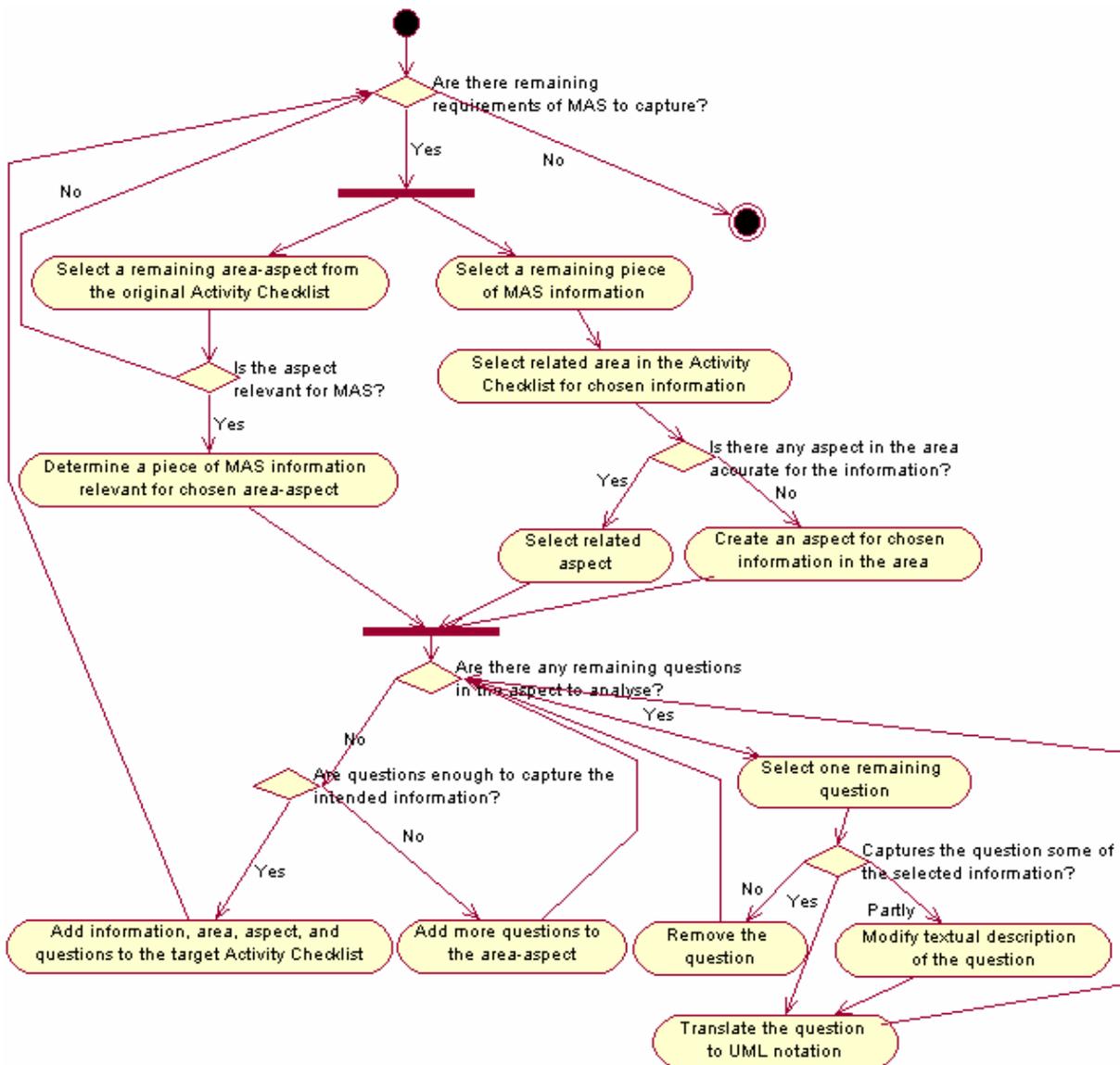
## 5. A CHECKLIST FOR MAS REQUIREMENTS ELICITATION

Neither the original Activity Checklist nor its evolution for MAS are conceived as static tools. Practitioners and developers should build or adapt them to their concrete concerns. This section presents a method to do these changes. A general overview of it can be seen in Figure 1. It has three main steps: choice of the information to obtain about a MAS; choice or creation of the related *area* and *aspect* in the checklist; adaptation or creation of the *questions* for the selected *aspect*. Now, we explain with further detail these steps.

The modification of the checklist for MAS requirements elicitation has two possible departure points. The first one is concerned with selecting a piece of knowledge required to model a MAS. This information has to be linked with the correspondent *area* of the Activity Checklist and associated with some of its *aspects*. If this *aspect* does not exist, a new one should be created. The other possibility is to select an *area* and one of its *aspects* of the checklist, and then determine if it captures some relevant feature for MAS design. Both approaches finish with a feature of MAS and an *area* and an *aspect* of the checklist relevant for that feature.

Following that selection, *questions* of the *aspect* are considered. These *questions* are analyzed with respect to their adequacy according to the following two objectives: obtaining the required knowledge to model a MAS and having a known translation into AT concepts. The first one is concerned with what the requirements are. It is based on a representation of the *questions* in natural language that does not require specific knowledge about MAS or AT. This representation is intended for customers and developers to discuss. The second one is concerned with the remaining development process. Here, our approach takes advantage of an UML-like language for AT concepts (Fuentes et al. 2003 a). This representation of *questions* acts as a template with slots that users fill in their answers. In this way, information is translated from natural language to the same vocabulary involved in the development.

Therefore, the aim of this analysis for a given *aspect* is getting, or building if needed, *questions* that satisfy the two already mentioned objectives using the proposed notations. The analysis of these *questions* can be split into two steps, one in textual form and other for using UML.



**Figure 1.** Process to adapt the Activity Checklist to MAS requirements elicitation

In the beginning, for every group of *questions*, developers analyze their textual form. All the textual forms of *questions* have to be understandable by non-experts and have to provide pieces of the required information related with their *aspect*. This stage needs a testing phase with people without knowledge of either AT or MAS. Those *questions* not satisfying these criteria have to be removed or modified. If there are not enough *questions* to capture all the intended knowledge, new *questions* have to be written. When this process finishes, the textual forms have to be translated to UML.

The second step in the analysis is the translation of *questions* from their textual form to their UML form. This last form of *questions* gives developers an idea of the kind of information expected in answers to those *questions*. In addition, it gives us templates that

put that information in a language that can be used in the remaining software process. This kind of translation is very similar to those made when conceptualising databases or in object-oriented analysis. The main sources of help here are the definitions of AT concepts and case studies already done in Sociology (Bednyi & Meister 1997, Engeström 1987, Gould et al. 2000, Kuutti 1996, Leontiev 1978, McGrath & Uden 2000, Uden et al. 2001, Vygotsky 1978). They do not provide mappings to analysis languages like those of software, but illustrate how to interpret the textual form of *questions* and answers under the light of AT. This transformation requires a deep knowledge about the Activity Checklist, AT, and MAS, but it only needs to be done once. *Question* mappings to UML can be used in other

developments without change, being part of the already available body of skills, for example embedded in automated tools.

In the next section we show some examples of how to modify the Activity Checklist and make its connection with AT concepts. They introduce the different situations that we found about possible MAS requirements to consider in the new checklist.

## 6. TYPES OF INFORMATION FOR THE CONSTRUCTION OF THE CHECKLIST

This section uses three examples to introduce different cases that appear when dealing with the different types of information about a MAS in the previous process. The first one corresponds to a situation in which the Activity Checklist is directly applicable to our problem with MAS. In the second one, the checklist requires some modifications in its use with MAS, not about the *aspect* but with its related *questions*. The third one is a case of information that may be useful when modelling MAS but that is not included as an *aspect* in the Activity Checklist.

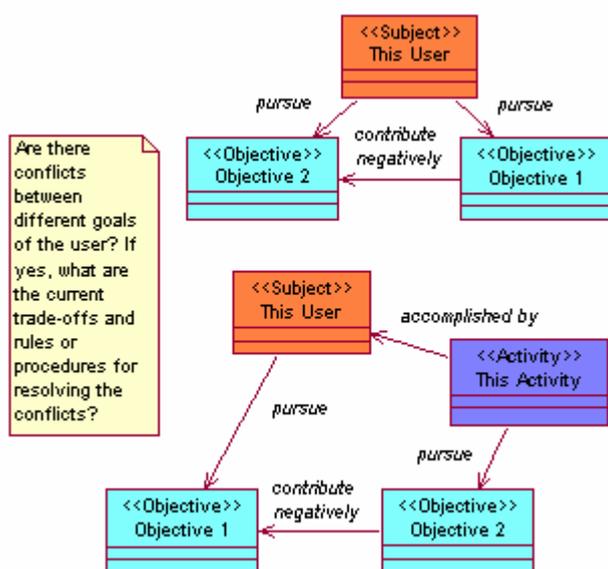


Figure 2. Sample graphical depiction of a question about conflicts between goals.

One of the aspects to analyze about the Means/ends area is “Potential conflicts between target goals.”. This aspect deals with negative contributions within the set of goals of the system or the environment to satisfy. The question “Are there conflicts between different goals of the user? If yes, what are the current trade-offs and rules or procedures for resolving the conflicts?” is related with this aspect. Both the aspect and the question are relevant for MAS. A MAS is composed by different agents and each one has its own agenda, which is not necessary working in the same path that others. Even when the formulation of that question is too evident to discover hidden problems, it represents information that has to be gathered. Possible graphical representations of this question can be seen in Figure 2. The upper diagram of Figure 2 shows that the user can have conflicts, or contradictions according to AT research, between his goals because he is pursuing objectives that are contradictory themselves. The lower diagram

illustrates that another source of conflict is that the subject accomplishes an activity that satisfies an objective contradictory with a user’s one.

Another different aspect of interest about a system is the “Role of existing technology in producing the outcomes of target actions.”, belonging to the Environment area. It refers to the importance that current technology (such as resources, applications, or agents) has to produce the desired result of actions. With this aspect, developers can evaluate the importance and integration of the system in a given activity. With that information, for instance, they can negotiate solutions to contradictory goals like those in the preceding question. One of the questions related with this aspect is: “Is target technology considered an important part of work activities?” As it is formulated, this question requires a subjective evaluation of the user answering the inquiry. In this case, perhaps the question can be replaced by this one: “What are the work activities involving the target system?” This new expression gives us an idea of how widespread the given system is and the importance of the processes involving it. Its representation can be as follows in Figure 3.

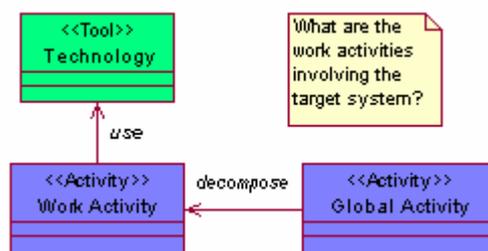
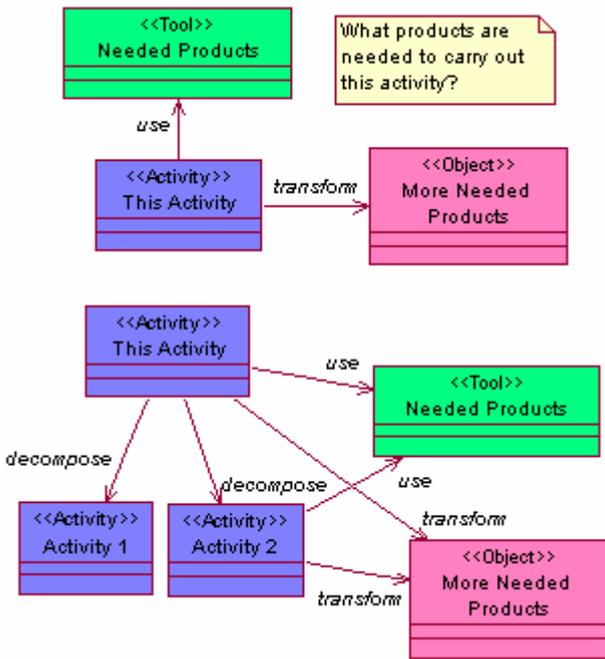


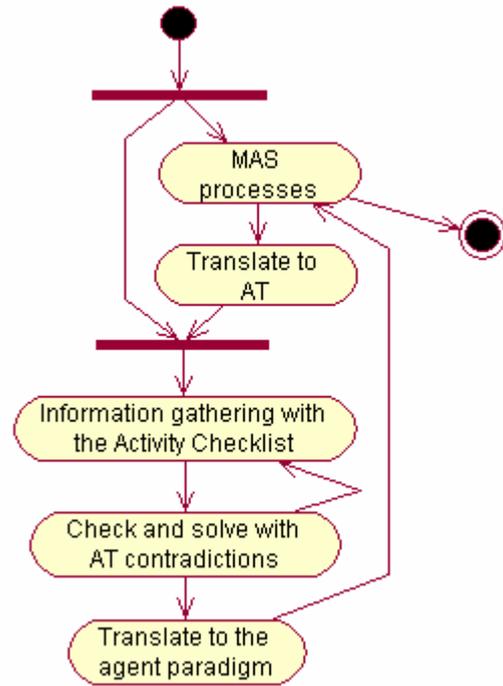
Figure 3. Sample graphical depiction of the role of technology in current activities.

The diagram in Figure 3 represents to developers the fact that the importance of a given technology for the organization is related to the activities in which it is involved. Studying these activities, users and developers can give a better evaluation of the role of the technology in the organization.

Finally, there are some important features for the requirements of MAS that do not appear at all in the original Activity Checklist. For example, functionality in MAS is quite often modelled as workflows, that is, as chains of tasks that use resources, consume items (which are related with their preconditions) and produce other items (which are related with their post conditions). This information is of a finer grain than that currently involved in the original checklist in (Kaptelinin et al. 1999) and thus, it does not appear in that with such details. Despite of this, it is very relevant to know how the involved subjects interact and what their dependencies are. This kind of information can be considered in the area of Environment with questions like “What products are needed to carry out this activity?” and “What are the expected products of this activity?”. The depiction of the first question, which is about the products, can be seen in Figure 4. The graphical representation reflects that the products needed for an activity are those directly used for that activity (see the upper diagram) or those used for activities that belong to its decomposition (see the lower diagram).



**Figure 4.** Sample graphical depiction of a question about items produced and consumed in workflows.



**Figure 5.** Requirements elicitation with the checklist in a software process supported by the AT.

The previous examples show the way to introduce modifications in the original checklist and new areas of introspection with their correspondent aspects and questions. As this section has illustrated, the most difficult part of the process is the translation to UML, because this is the point where AT concepts have to be identified over the textual form of the question, which is inherently ambiguous due to the use of the natural language

Of course, the new Activity Checklist does not intend to capture all the requirements of a MAS in every context. It is a tool that should be used combined with classic Software Engineering tools.

## 7. USING THE MODIFIED CHECKLIST

Once that the checklist is modified, developers have a set of questions with a known representation in UML. Using mappings like those in Table 2, agent abstractions representing requirements can be translated forward and backwards to AT concepts represented with UML. The translated specifications are the basis to elicit new knowledge. The AT representation for questions is also the departure point to use contradictions in models as a guide for the evolution of development according to AT (Fuentes et al. 2003 a). The method proposed for using the modified checklist appears in the activity diagrams of UML in Figure 5 and Figure 6. Figure 5 shows the general scheme of the requirements elicitation process while Figure 6 is focused on how questions are answered by customers with the help of developers.

As stated before, Figure 5 gives an overview of the requirements elicitation process. The first step consists in obtaining a translation of the current MAS specifications using mappings. After that, developers have to answer the questions of the checklist and put the information obtained in the UML form. The fulfilled templates are analysed according to the contradictions obtained from AT research. This kind of checking process is described in (Fuentes et al. 2003 b). Contradictions guide a refinement process about the information in requirements. When the statement of requirements is considered enough stable and the contradictions present in it as bearable, requirements can be translated to the abstractions of the given MAS methodology. These translated requirements are the basis for the remaining development process. Besides this, as mappings are bidirectional, MAS models can always be translated back to AT and checked against the original requirements or contradictions. Therefore, this process with mappings also allows the traceability of requirements in the products of the software cycle.

Figure 6 illustrates with more detail the way in which customers and developers answer questions together. The group for elicitation works selecting questions from the new Activity Checklist that are relevant for the information they are trying to gather. The adapted questions have a textual form and an UML form. These forms give to customers and developers the departure point for discussing, discover the information, and put it in the frame that is the UML form. A library of examples with questions and their answers in other projects can be consulted. This library allows people to get a better idea about what kind of answer questions request. Besides, information already included in the specifications is available to fulfil the slots.

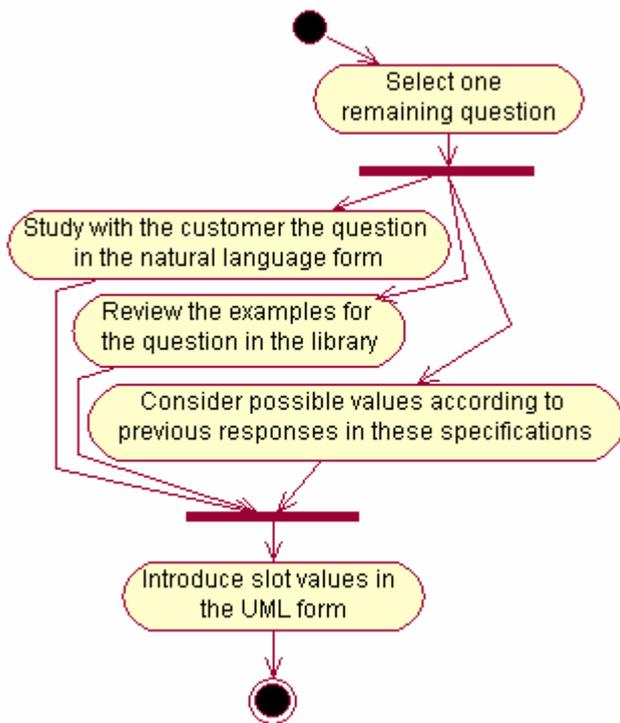


Figure 6. Information gathering with the Activity Checklist.

Of course, since the Activity Checklist itself cannot be complete for every possible requirement, this process cannot elicit the full set of features of a MAS: However, the process to modify the Activity Checklist previously introduced (see section 5) can be used to add new features to the checklist and, therefore, to customize it for a given domain.

## 8. THE ACTIVITY CHECKLIST IN INGENIAS

The present work with the Activity Checklist is embedded in an ongoing effort to make MAS a development paradigm that can be widely used in industrial settings. As part of this research, our group has developed the INGENIAS agent-oriented methodology (Gómez & Pavón 2003). In this broader framework, our aim is to provide the techniques to validate and verify the intentional and social aspects of MAS.

The use of software methodologies in real projects demands support tools that automate, at least partly, the work needed to develop a system with them. Attending these considerations, INGENIAS has a software environment for modelling and development called the INGENIAS Development Kit (IDK). This IDK allows users create their own models for MAS and process them with plug-ins called modules. Examples of these modules are those to generate documentation or simulate the behaviour of the system. To validate our approach, we have developed one of these modules (see the snapshot in Figure 7).

The module for the Activity Checklist in the IDK allows developers to define new questions for the checklist, with both its textual and UML form. Both questions and the mappings to

translate them and their answers are specified as XML files to improve the flexibility of the module.

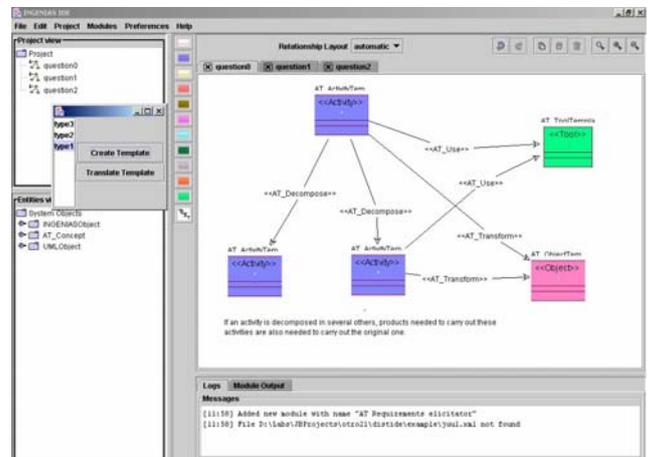


Figure 7. INGENIAS Development Kit with the Activity Checklist module.

The module automatically translates the answers to the questions to the language of the MAS methodology with the mappings. Beginning with the translated answers, developers and users can work with the information following the MAS process. When they try to check properties about requirements traceability or the social features, they can translate back the specifications to the UML language for AT and work with the abstractions and techniques of the AT.

## 9. CURRENT STATE

The method and its supporting tools have been checked in several case studies until now. The full specification of them with the INGENIAS methodology can be found in <http://grasia.fdi.ucm.es/ingenias>. Some of them are:

- **Juul Boklander.** It is written by Espen Andersen from <http://www.aspen.com>. The case study describes a bookseller company that sells books to university students. Our work consists in specifying the MAS for its website.
- **Robocode.** It is an environment to simulate tank battles that can be found in <http://robocode.alphaworks.ibm.com>. The aim was to model the collaboration in an army and the command of its captain.
- **PC Assistants.** A set of agents that help the user to customize the use of his computer and programs.

To study the benefits of using our tools for requirements elicitation, these case studies were proposed to two different groups of users. The first one just uses the core of the INGENIAS methodology. The other one was provided with the Activity Checklist for MAS. After the development, both sets were interviewed about topics like guidelines, usability, or quality of obtained requirements.

In general, users found the checklist a helpful guideline to make their elicitation go ahead. Users with the new Activity Checklist increase their performance in the task and usually obtain a greater number of requirements. As drawbacks, we have to consider mainly two. The first one is that users were many times reluctant to the proposed translation to the UML. They consider that the textual form of the question should have a different translation. This is not a real problem as users can model the requirement in different ways: the textual form allows quite different models in UML. The second one is that the checklist reduces the possible serendipity of users when discovering requirements. They are guided in a given way and do not try to make new paths.

## 10. CONCLUSIONS

Current approaches to discover and model the environment of MAS are for the most part “bottom up” ones. They start with an empirical analysis of contextual factors and gradually develop concepts that are finally put in an appropriate theoretical framework. This is the case for techniques such as “task decomposition”, “future workshops”, or “flow models”. Many researches point out that these approaches, “bottom up” or empirically-driven strategies can be complemented with a “top down” one. Between these last approaches is the Activity Checklist. The Activity Checklist is intended to make concrete the abstract concepts belonging to the AT. Despite this, it is still too general to be directly applicable to the design process of MAS. To solve this problem, our proposal works in three ways:

1. Study and modify *areas*, *aspects*, and *questions* of the Activity Checklist to make them more concrete and closer to the information required for MAS. This allows establishing a correspondence between the checklist and concepts of AT and MAS.
2. Give the correspondence of *questions* with AT concepts and provide a graphical depiction for them. Under this representation, *questions* act as templates where users fill slots with information. This is a way to ease the understanding of *questions* and their answering.
3. Provide a library with examples of these *questions* in order to allow that both customers and users can learn quickly about the required information.

This three-faced solution has several advantages. In first place, basing the elicitation process in a well founded sociological theory, the Activity Checklist and the AT, helps to understand and capture the complexity of the environment, especially the human one. As a second advantage, the requirements elicitation stage shares the same concepts of the AT with the rest of the development thanks to mappings. This connection allows a propagation of requirements through the whole development process easing the validation process. Finally, the techniques related with the management of contradictions from AT can be applied from the earlier stages of the software process.

Two are the main difficulties in applying the proposed method. The first one is that the field of application changes, from HCI to MAS, and it is very hard to state that the checklist is adequate to capture most of the relevant requirements for the MAS. Like the original checklist of Kaptelinin, the modified checklist can be considered as a supporting tool that developers should use with conventional engineering techniques. The second difficulty is to

translate questions from their textual language form to diagrams with AT concepts. This task demands a deep knowledge of involved concepts and tuning of mappings according to experience. Of course, provided mappings are not the only possible ones. Developers can modify them to get a better correspondence between provided information and its representation.

A final remark about this approach is that working with automated tools and people without a deep knowledge of AT, enforces to make an effort about the conceptualisation of AT and the execution of its techniques that is beyond the current ones. We consider that this effort could bring new insights to the field of AT research.

## 11. REFERENCES

- [1] [Bednyi & Meister 1997] Bednyi, G. Z., Meister, D., 1997. *The Russian Theory of Activity: Current Application to Design and Learning*. Lawrence Erlbaum Associates.
- [2] [Bødker 1991] Bødker, S., 1991. Activity Theory as a Challenge to System Design. In: H.E. Nissen, H.K. Klein & R. Hirschleim eds., *Information Systems Research: contemporary approaches an emergent traditions* IFIP, Elsevier Science Publishers. BV (North-Holland).
- [3] [Dardenne *et al.* 1993] Dardenne, A., van Lamsweerde, A., Fickas, S., 1993. Goal-directed Requirements Acquisition. *Science of Computer Programming*, 20, 3-50.
- [4] [Demazeau 1995] Demazeau, Y., 1995. From cognitive interactions to collective behaviour in agent-based systems. 1st European Conference on Cognitive Science, April 1995 Saint-Malo. 117-132.
- [5] [Engeström 1987] Engeström, Y., 1987. Learning by expanding. Helsinki : Orientakonsultit.
- [6] [Fjeld *et al.* 2002] Fjeld, M., Lauche, K., Bichsel, M., Voorhorst, F., Krueger, H., Rauterberg, M., 2002. Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware. In: B. A. Nardi & D. F. Redmiles eds. a Special Issue of *Computer Supported Cooperative Work (CSCW): Activity Theory and the Practice of Design*, 11 (1-2), 153-180.
- [7] [Fuentes *et al.* 2003 a] Fuentes, R., Gómez-Sanz, J. J., Pavón, J., 2004. Activity Theory for the Analysis and Design of Multi-Agent Systems. *4th International Workshop (AOSE 2003), July 2003, Melbourne, Australia. Lecture Notes in Computer Science*, 2935, 110–122, Springer Verlag.
- [8] [Fuentes *et al.* 2003 b] Fuentes, R., Gómez-Sanz, J. J., Pavón, J., 2004. Social Analysis of Multi-Agent Systems with Activity Theory. *10th International Conference CAEPIA 2003, November 2003, San Sebastián, Spain. Lecture Notes in Artificial Intelligence*, Springer Verlag. To appear.
- [9] [Goguen & Linde 1993] Goguen, J.A., Linde, C., 1993. Techniques for requirements elicitation. *1st IEEE International Symposium on Requirements Engineering (RE'93)*, 152-164.

- [10] [Gómez & Pavón 2003] Gómez-Sanz, J. Pavón J., 2003. Agent Oriented Software Engineering with INGENIAS. 3rd International/Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003), June 2003, Prague, Czech Republic.
- [11] [Gould *et al.* 2000] Gould, E., Verenikina, I., Hasan H., 2000. Activity Theory as a Basis for the Design of Web Based System of Inquiry for World War 1 Data. *23rd IRIS Conference, August, 2000, Lingatan, Sweden*, 761-770.
- [12] [Hilburn *et al.* 1999] Hilburn, T. B., Hirmanpour, I., Khajenoori, S., Turner, R., Qasem, A. (1999). *A Software Engineering Body of Knowledge Version 1.0*. Technical Report CMU/SEI-99-TR-004 [online]. CMU/SEI. Available from: [http://www.sei.cmu.edu/pub/documents/99\\_reports/pdf/99tr004.pdf](http://www.sei.cmu.edu/pub/documents/99_reports/pdf/99tr004.pdf) [Accessed 27 April 2004]
- [13] [Kaptelinin *et al.* 1999] Kaptelinin, V., Nardi, B. A., Macaulay, C., 1999. The Activity Checklist: A tool for representing the “space” of context. *Interactions*, 6 (4), 27-39.
- [14] [Kuutti 1996] Kuutti, K., 1996. Activity Theory as a potential framework for Human-computer interaction research. In: B.A. Nardi ed. *Context and Consciousness: Activity Theory and Human-Computer Interaction*, Cambridge, MA: MIT Press.
- [15] [Lamsweerde 2000] van Lamsweerde, A., 2000. Requirements Engineering in the Year 00: A Research Perspective. *22nd International Conference on Software Engineering (ICSE-2000), June 4-11, 2000, Limerick, Ireland*. ACM.
- [16] [Leontiev 1978] Leontiev, A. N., 1978. *Activity, Consciousness, and Personality*. Prentice-Hall.
- [17] [Maes 1994] Maes, P., 1994. Modeling Adaptative Autonomous Agents. In: C. Langton, ed. *Artificial Life Journal*, 1 (1-2), MIT Press.
- [18] [Magne & Sindre 1993] Magne Høydalsvik, G., Sindre, G., 1993. On the purpose of Object-Oriented Analysis. OOPSLA'93, September 26 – October 1 1993. Washington D.C., USA: ACM Press, 240-255.
- [19] [McGrath & Uden 2000] McGrath, M, Uden, L., 2000. Modelling Softer Aspects of the Software Development Process: An Activity Theory based approach. 33rd Hawaii International Conference on System Sciences (HICSS-33), Wailea, Maui, Hawaii, USA. Software Process Improvement. IEEE Computer Society Press. January 2000.
- [20] [Newell 1982] Newell, A., 1982. The knowledge level. *Artificial Intelligence*, 18., 87-127.
- [21] [Nuseibeh & Easterbrook 2000] Nuseibeh, B., Easterbrook, S., 2000. Requirements Engineering: A Roadmap. *22nd International Conference on Software Engineering (ICSE-2000), June 4-11 2000, Limerick, Ireland*. ACM Press.
- [22] [OMG 2003] OMG (2003). *Unified Modeling Language Specification. Version 1.5* [online]. Available from: <http://www.omg.org>
- [23] [Probert 1999] Probert, S. K., 1999. Requirements engineering, soft system methodology and workforce empowerment. *Requirements Engineering*, 4 (1999), 85-91, Springer-Verlag, London.
- [24] [Ross & Schoman 1977] Ross, D. T., Schoman, K. E., 1977. Structured Analysis for Requirements Definition. *IEEE Transactions on Software Engineering*, 3 (1), 6-15.
- [25] [Sykara 1998] Sykara, K. P., 1998. Multiagent systems. *AI Magazine*, 19 (2).
- [26] [Uden *et al.* 2001] Uden, L., Liu, K., Shank, G., 2001. Linking Radical Constructivism and Semiotics to Design a Constructivist Learning Environment. *Journal of Computing in Higher Education*, 12 (2) Spring 2001, 34-51. Amherst, MA: Norris Publishers.
- [27] [Vygotsky 1978] Vygotsky, L. S., 1978. *Mind and Society*. Cambridge MA, Harvard University.
- [28] [Yu 1997] Yu, E. S. K., 1997. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. *3rd IEEE International Symposium on Requirements Engineering (RE'97), Jan. 6-8, 1997, Washington D.C., USA*. 226-235.
- [29] [Zave 1997] Zave, P., 1997. Classification of Research Efforts in Requirements Engineering. *ACM Computing Surveys* XXIX(4), 315-321. December 1997.
- [30] [Zave & Jackson 1997] Zave, P., Jackson, M., 1997. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology*, 6(1), 1-30, January 1997.

# Appropriating Patterns for the Activity Theory Toolkit

Elizabeth S. Guy

University of Brighton, School of Computing, Mathematical and Information Sciences  
Lewes Road, Moulsecoomb, Brighton, BN2 4GJ, UK  
e.s.guy@bton.ac.uk

## ABSTRACT

This paper investigates a method for modelling computer-supported cooperative work, to provide a common language for users and developers collaborating in design. The research is grounded in an empirical study of the in-house development of groupware and the work practice of system developers. Through an appropriation of Christopher Alexander's architectural pattern language, it is proposed that patterns have the potential to be a practicable tool that both embodies the principles and methodology of activity theory, and fits the requirements of this design process.

## Keywords

Design patterns, pattern language, activity model, computer-supported cooperative work

## INTRODUCTION

Over the past fifteen years there has been a steady growth of interest in activity theory (AT) among a new audience in the fields of Human-Computer Interaction (HCI) and Computer-Supported Cooperative Work (CSCW). Despite a growing corpus of research using it as a theoretical approach, there has not been a corresponding initiative in developing AT as a *practicable method*. By 'practicable' we mean useful to practitioners in real-world projects for the design of information systems (IS), user interaction and the associated work practice, rather than academic research. One of the reasons for this is possibly that AT is perceived, by its new audience, as being "[...] hard to learn" (Nardi, 1996, p.9). Mastering AT can be demanding for someone with no background in its philosophical traditions. The question of whether it is realistic to expect developers and users to be receptive to methods based on such seemingly alien principles is a valid one.

Despite the difficulties, researchers who are convinced that AT is well worth the trouble might feel obliged to engage with this problem. This means developing tools that are not only grounded in AT, but suit the needs, culture and constraints of practitioners' work. In order to explore the problem we first look at a precedent - the way that equally "hard to learn" theory has been effectively embodied in traditional IS development methods. We then consider the

essence of AT and its implications for tools that support the application of AT in design. The practical needs of systems developers for lightweight tools that support collective reasoning about design are considered, based on an empirical study of the in-house development of groupware to support information sharing. Next, we present a partly principled, partly opportunistic appropriation of Christopher Alexander's architectural pattern language (Alexander *et al*, 1977; Alexander, 1979). This is justified as a design tool that both embodies the principles and methodology of AT, and has the potential to fit the practice of systems developers and interaction designers. In the final sections the method is described in the form of a practical guide, followed by a discussion of the further work that remains to be done to evaluate it. In the Appendix we have included a subset of four patterns as a demonstration of the technique.

## COMPARISON WITH THEORY IN SAD METHODS

The failure of researchers to develop AT as a tool for practitioners suggests that many of us, after the effort of applying it ourselves, may feel that AT is simply too difficult to be used in this way. However, there are several precedents in the way that other theoretical approaches have been explicitly and implicitly embodied in practitioner methods in the fields of IS and HCI. To take just one example, the way that traditional systems analysis and design (SAD) methods have incorporated the concepts of systems theory is a heartening example of how this can be achieved. Systems thinking, with its origins in many disciplines such as theoretical biology, cybernetics, and information and communications theory (Lilienfeld, 1978; Checkland, 1981), is a hodgepodge of abstractions about social organisation as 'systems'. Systemic concepts are embodied in SAD methods and techniques, such as Checkland's Soft Systems Methodology (SSM) (ibid 1981; Checkland and Scholes, 1996), perhaps the purest embodiment of systems theory as a practical method for modelling "human activity<sup>1</sup> systems". SSM has a set of procedures and a number of modelling techniques, including a Formal Systems Model that is a checklist of the characteristics that are said to describe a system.

---

<sup>1</sup> Activity here is used in a very different sense to the way that it used in AT. Checkland's concept of a human activity system is that of an ideal, notional construct rather than real, concrete activity (Checkland, 1981, p314).

In the past, SAD methods have been practiced in the professional development of large transaction processing systems and, more recently, corporate databases. Practitioners have apparently not been deterred from using them by their problematic theoretical basis. Indeed, many text books on systems analysis present the techniques without any exposition of their foundational theory at all. This is not to say that in order to be acceptable to practitioners AT-based methods and techniques should adopt this approach and, like Clark Kent, keep their real identity and super-powers a secret. It does suggest that it is possible to embody theoretical principles in practicable methods, without making the unrealistic demand on practitioners that they should first become experts in the theory, before being able to use them.

### **DESIGNING ACTIVITY THEORY TOOLS**

The SAD example may demonstrate that the project is achievable, but, unlike systems theory, AT is not a set of abstract propositions with which to describe the world. Rather it can be understood as a methodology that seeks to understand activity in the process of change and development, and as a tool for intervention in change. It is therefore vital, in seeking to present AT as an accessible and usable method, not to fall into the trap of using it as a set of categories for merely describing activity, similar to the approach of Checkland's Formal Systems Model. Some recent discussions of AT have tended to move in this direction and to emphasise its undoubted usefulness as a tool to describe the totality of the social organisation and context of work (Nardi, 1996, p.3).

AT uses the conceptual tool of contradictions (Engestrom, 1987; Bertelsen and Bodker, 2003) to reveal the underlying dialectical relations that drive development. Designing a method that embodies the principles of AT requires not just that it should be able to describe the categories that are the outward form of activity. It should also support the revelation and explanation of the dynamic inner processes within activity systems, so that developers and users may be able to use it as a tool for change-oriented design. In order to do this, it should support the historical analysis of these processes; reconstruction of the process of change; and identification and representation of innovative ways of working with tools as they emerge from contradictions, using them to inform the design of future systems.

Vygotsky, writing about the development of a method adequate to explain the nature and development of psychological processes, talks about method as something that is "[...] simultaneously prerequisite and product, the tool and result of the study" (Vygotsky, 1962, p.65). This suggests that a method founded in AT should not be a fixed or rigid thing, like a prescriptive SAD set of procedures and techniques, but rather an artifact that can be dynamically configured and adapted to the requirements of the design project. It should be able to evolve historically as the field of design changes and developers learn from experience.

### **SYSTEM DEVELOPMENT PRACTICE**

An AT-based method has not just to be consistent with the methodology of AT, but also to fit the practical needs of developers. The case study and 'facilitator' set of patterns presented in this paper (see the Appendix) come from the findings of a wider research project into systems development practice, carried out between 1997 and 2002 (Guy, in preparation). The field site for the research was the international centre of 'GreenFam'<sup>2</sup>, a global non-governmental organisation which campaigns for human rights, based in London, UK. The work of the centre involves centralised administrative support for the GreenFam movement: coordination of global campaigns; research carried out by teams organised around world regions and sub-regions; and the mainly in-house provision of information technology (IT) support for these activities. The objective of the extended study was to study the methods and techniques used by GreenFam's IT Program (ITP), during a major, long term project. This involved supporting teams with tools for information sharing and the coordination of campaign work, using the technology of LOTUS NOTES™ (Notes).

The GreenFam study was carried out in two phases: participant observation over eighteen months during the initial stages of the Notes project; returning three years later to evaluate several Notes databases as they had evolved through use, in order to produce guidelines to inform future design projects. One of the outcomes from the project, in particular from the evaluation phase, is a GreenFam pattern language (Alexander *et al*, 1977) which is intended to represent some of the findings in a form that can be used as a tool for design. The patterns in the Appendix are a subset of over twenty GreenFam patterns which cover the design of common information spaces (Bannon and Bodker, 1997) for the support of work with information artifacts.

#### **The Right Tools for the Job**

Throughout the Notes project it was evident that the developers at GreenFam lacked the right tools for the job. They were experienced in the process of "rolling out" standard software to users organisation-wide, and had envisaged a modified roll-out where a template Notes document and discussion database would be provided for each team or campaign, without much need for further customisation. This method was inadequate in a context where the requirements and culture within and between teams varied enormously. The ITP, it transpired, in fact knew very little about the complex ways in which users worked on and with information. An alternative approach adopted in one project of intensively "working with users", involving prototyping and group meetings over a period of

---

<sup>2</sup> Names have been changed in order to protect the identities of the research partner and the people working there.

several months, was a valuable learning experience, but had too high an overhead to be practicable in every case.

The GreenFam developers needed what they articulated as “a new kind of analysis” and tools that would help to capture configurable, reusable design solutions, when designing each application from scratch was not feasible. This resonates with John Carroll’s justification for the method of designing with scenarios - “Systems development is now in need of a guiding middle-level abstraction, a concept less formal and less grand than specification, but that can be broadly and effectively applied.” (Carroll, 2000, p.17.)

The vision for a new, more collaborative way of working at GreenFam had been set out in a document which evaluated the different software alternatives before the project began. Scenarios envisioned a future way of working, where Notes databases would be used to share information and to create a permanent archive that could be accessed as a repository of past experience. The vision was communicated top-down from GreenFam’s decision-making bodies and was embraced by the ITP. For them the scenario was “[...] a prototype of [...] future-directed action – in which the future is more than the blindly inevitable fact of succession in time and includes some envisioned goal as its content.” (Wartofsky, 1979, p.141.)

This was not a vision that was shared throughout the organisation and it represented a big departure from current practice for many teams. Many individuals took responsibility for managing their own cases and personal networks of information sources, and did not perceive a need to put information in common. Where information was shared, existing tools such as email were preferred to the overhead of adopting a new tool and way of working.

As the project progressed and the ITP experienced the difficulty of gaining acceptance for the new tools, doubts and dissent surfaced among the developers. The Notes project crossed the functional boundaries which separated development of databases, and support for document production and management, into two specialised teams. Each team interpreted Notes in the light of their previous experience and it became clear that there was no shared understanding of what Notes was or how it should be used among the members of the ITP either.

The problem for the work of the ITP was a lack of appropriate secondary artifacts (Wartofsky, 1979) – models for representing alternative design solutions that could mediate between different groups and viewpoints. Taxén (see these proceedings) describes the importance of a method or tool to establish a “working consensus” among actors in complex, distributed projects. In particular, participants need to establish shared meanings for abstract concepts – such as whether Notes is a database or a document management system or, indeed, an environment for collaborative work. Bertelsen and Bodker’s (2002) metaphor of the “parallel rooms” of design practice and use

practice (*ibid*, p.410) is apposite for describing this problem. They relate the discontinuity between these parallel rooms to the heterogeneous groups who participate in design. At GreenFam we found that these discontinuities extended beyond the dyad of design and use, to the different practical cultures within the ITP, that had become established over time.

It follows from the GreenFam study that artifacts to support collaborative design must satisfy several conditions:

- Models must be equally accessible to developers and users, and be a *lingua franca* for dialogue and discussion (Erickson, 2000b).
- They must mediate between a vision of a future way of working with new tools, and past, present and emerging practices, in order that developers and users can engage with this design space and co-construct new solutions.
- They must support representation and resolution of the dynamic and contradictory features of mediated work.

#### **IDENTIFYING AND APPROPRIATING PATTERNS**

One of the benefits of a lengthy program of research in one organization is that after a while strong, recurring patterns of what works and what does not can be identified as they emerge in new practices with evolving tools. For example, there were some instances where Notes databases were used by teams as a real shared information space and became integrated with their work. Such examples were always associated with an enthusiastic individual who acted as a facilitator and took on the responsibility of posting information to the database, directing other team members to it. Crucial to this work was the ability to make use of email, the historically embedded and preferred tool of the organization, to send hyperlinks in messages. These, when clicked, took the recipient directly to the document in the database. As we began to identify patterns of activity in our field study, we were inspired to represent them in the form of a pattern language, modifying a method that has a growing following in software engineering (SE) and HCI.

Christopher Alexander, the creator of the architectural design patterns that are currently in vogue in object-oriented SE (Gamma *et al*, 1995), use case specification (Adolph *et al*, 2003), web site design (Graham, 2003; van Duyne *et al*, 2003), interaction design (Borchers, 2001a, 2001b), CSCW (Erickson, 2000a; Martin *et al*, 2001) and participatory design (Dearden *et al*, 2002), is a rather strange bedfellow for AT. It is beyond the scope of this paper to give a detailed review or critique of his work (Alexander *et al*, 1975; Alexander *et al*, 1977; Alexander, 1979; Alexander *et al*, 1985), but also not necessary. Just as with these other approaches that have appropriated the patterns idea, our development of patterns as a tool for AT-based design departs opportunistically from Alexander wherever his philosophy contradicts that of AT. Alexander’s guiding principles of the timelessness and naturalness of archetypal patterns that have “the quality

without a name”, and the “invariance” of design solutions that have this quality (*ibid*, 1979), are antithetical to the method of dialectical materialism. However, other Alexandrian principles are more directly transferable:

- The empowerment of users to shape their own environment through the tool of an accessible, shared pattern language with the support of an expert facilitator.
- The definition of a pattern as a “three-part rule, which expresses a relation between a certain context, a problem and a solution” (Alexander, 1979, p.247). This contextualisation of problems and solutions, which may be historical or to do with the current conditions, is consistent with the method of AT.
- The definition of a problem in context as being caused by a system of forces which arises in that context (*ibid*, p.253). This lends itself to being translated into the language of AT’s contradictions.
- The scope, modularity and unity of the systemic pattern language, which is made up of a network of related patterns. Patterns can be written for different levels of activity, actions, and for the conditions for operations (Leont’ev, 1979, 1981) while retaining the integrity of AT’s unit of analysis.

#### THE ACTIVITY PATTERNS METHOD

Having outlined the origins and theoretical basis of activity patterns, and discussed the empirical study in which the method is grounded, this section will briefly describe a practitioner method for designing activity with patterns. The context for which this method has been developed is the in-house development and customisation of proprietary groupware in a non-profit organisation. This has coloured the approach, but we believe that using patterns in the analysis and design of collaborative systems has a wider relevance. They could, for example, be used for modelling activity in the application domains of collaborative learning environments, interactive web sites, or new media for domestic or leisure applications. The method is well-suited to in-house development as it supports user participation in the design process. Consultants and software producers – who work in contexts where software and interaction design patterns are already being generated – might also find that activity patterns are a useful addition to their repertoire of techniques, because of their reusability.

#### Activity Patterns in the Design Process

An open question for this project is whether practitioners will be prepared to make the effort to learn AT. In the method proposed here it is necessary to learn some basic principles of AT, but the trade-off is that activity patterns require no special skills to write and are very versatile. They have the potential to be used in any part of the design process where an understanding of tool mediated work in its context is required:

- generalising data from a requirements investigation – literally discovering patterns in users’ work;
- specifying and documenting the higher level conceptual design of systems;
- participatory design and prototyping;
- as criteria for the evaluation of prototypes or systems in use;
- representing the findings of evaluation in a design-oriented way;
- as a benchmark to evaluate the support delivered by software products before purchasing.

Activity patterns are modular and can be re-used in similar contexts, thereby cutting down on the workload of systems development. They interface well with lower level system specification techniques, such as UML use cases or interaction design pattern languages (for example Borchers, 2001a; van Duyne *et al*, 2003). The patterns presented in this paper are addressed to the problem of broadly representing activity and actions, but not to the detailed modelling of how actions are accomplished as sequences of tool mediated operations (see papers by Bertelsen, Harris in these proceedings, also Harris, 2004).

#### Embodying Activity Theory Principles in Patterns

The unit of analysis of AT (Engestrom, 1987) is not something that can be decomposed, but is a dynamically related system. A *pattern language* preserves this unity. Although discrete patterns can be written for the elements of an activity system – for example, the design of *tools*; the work of a *subject*; *rules* such as organisational policy and procedures; the roles within the *division of labour* or *community* of the workgroup – each pattern is related to the larger patterns it helps to complete, and the smaller patterns that complete it. This is done by stating the associations between patterns in the definition of each pattern (for how this is done see the template in the Appendix, *Table II*). In this way a systemic pattern language is constructed, consisting of a network of explicitly related patterns. It would be a nonsense to consider any pattern in isolation: for example the ONE CLICK HYPERLINK or EMAIL ALERTS patterns in the Appendix are only meaningful in the context of a specific social and technical context. The same goes for the higher level patterns. The policy of treating INFORMATION AS COMMON PROPERTY is the policy of a concrete activity system; it is realised through its related lower level patterns.

We have used another AT concept in the activity patterns – the idea of hierarchical levels of activity (Leont’ev, 1978; 1981). This is used in a similar way to Alexander’s use of ‘scale’ to structure the scope of his patterns for the built environment. An activity is motivated by a human need to transform some aspect of the material world. Activities are realised through concrete actions which are directed to achieving specific goals. Actions are accomplished through a series of operations which are performed under the specific conditions in which an action takes place. As they are learned, operations are internalised and become

automatic, providing that the tool supports them in intuitive ways. The levels of activity, actions and operations are summarised in *Table I*.

Activity	Motive	E.g. Sharing information; organising a campaign
↑ ↓	↑ ↓	↑ ↓
Action	Goal	E.g. Posting a news article in the information space; writing an email alert message
↑ ↓	↑ ↓	↑ ↓
Operation	Conditions	E.g. Clicking a hyperlink to open a document or an action button to send an email

*Table I. Levels of activity with examples from GreenFam.*

The levels of activity – although they are hierarchical – cannot be decomposed. Patterns can be written for different levels of activity and the relationships between patterns once again preserve this unity. We have included patterns in our examples to demonstrate how they can be used to model these different levels. INFORMATION AS COMMON PROPERTY relates to the *activity* level; EMAIL ALERTS represents a tool-mediated series of *actions*. ONE CLICK HYPERLINK is at the level where the action could then be modelled as a series of operations, using a unit of analysis focussed on the individual (see Harris, these proceedings).

The concept of contradictions is fundamental to AT, and central to the representation of activity in the patterns. Christopher Alexander’s definition of a pattern is, as we have seen, a rule about how to resolve a “system of forces” which always arise in a given context. Alexander’s approach is often profoundly idealistic and it is not always clear exactly what he means by a system of forces relevant to the configuration of space in the built environment (Alexander, 1979, p.248) there is no coherent theoretical foundation. His concept of a force is certainly very different from the understanding of a contradiction in AT. In a statement which is anathema to AT he defines his holy grail - the “quality without a name” - as the freedom from inner contradictions (*ibid* p. 26).

In contrast AT sees activity systems as being in essence dynamic, undergoing a process of change and development. The driving force behind that dynamic is contradiction – the continual breakdown and temporary resolution of the inner relations within the system. There is no such state as freedom from inner contradictions in activity systems: to hold this would be to deny that people are continually re-shaping their environment through their activity and, in so doing, changing themselves as they learn through their experiences. We have therefore reinterpreted Alexander’s definition of a pattern and the problem statement at its core:

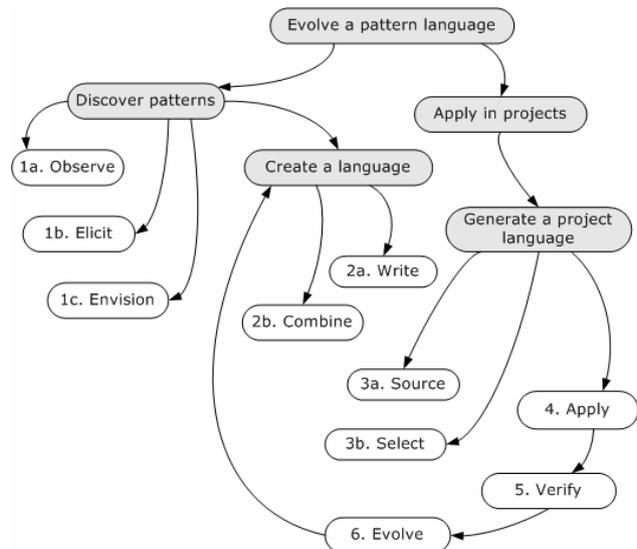
*An activity pattern is a three-part rule which establishes a relationship between a context, a contradiction that arises in that context, and its resolution, which takes it from its current state to a more developed one.*

We do this with the proviso that any pattern only resolves a contradiction temporarily and that a pattern language must evolve as activity and its context changes. Some patterns will have more longevity than others – depending on the rate of change in the activity they represent – but none will be timeless. It is the ability of a pattern language to change and develop that makes it such an apposite tool for AT.

In summary, activity patterns embody the concepts of AT – its unit of analysis; the hierarchical levels of activity, actions and operations; and the contradictions which are the driving force for change, which have been made a central part of the pattern. We will now go on to describe a framework for writing patterns and developing a pattern language through applying it in the design process.

### DESIGNING ACTIVITY WITH PATTERNS

It is a fundamental principle of the activity patterns method that the pattern language evolves continuously as it is put to use. It is both a *tool* that is applied in projects and an *outcome*, as what is learned from the project is fed back into further development of the language. This feature of the tool fits well with the rapid pace of change in technologies to support complex work. It also means that it can be used to capture and share the design knowledge and expertise of developers as they learn. We will now describe how to develop an evolutionary pattern language as a network of actions which are shown below in *Figure 1*.



*Figure 1. Evolution of a pattern language.*

#### 1 - Discover Patterns

Generally patterns are derived from experience and so may be discovered in two ways – either from direct observation of how people are using tools in context, or from the accumulated experience of developers and users. However, we also see the potential of patterns for visualising and designing future systems and ways of working. They can be used in this way to change people’s perceptions of what is possible, and bridge the space between the current activity and what is desired for the future. We therefore

describe three ways in which patterns may be discovered – by *observation*, *elicitation* or *envisioning* the future.

#### 1a - Observe

Empirical data about how users do their work is typically gathered in the preliminary requirements investigation phase of a project; or through evaluation of systems in use; or, if a participatory approach is adopted, through co-design and prototyping. Material for patterns could be discovered through any of these activities, using qualitative research techniques such as observation, workshops or interviews carried out with users as they work.

Requirements investigation has the potential to generate huge amounts of data about the work of individual users, and analysing this data is a process of discovering relevant themes. Patterns are an effective way of generalising some of the main findings in a form that is oriented to design. Once written up, the patterns can be used to check and validate the findings with users, and then be improved. Patterns are a technique that help to make sense of the rich and often inconsistent data from a workplace study, without moving prematurely into lower level modelling techniques that represent the operational level of work.

Being aware of the concept of contradictions can sensitise developers to look for the sources of a failure to progress and tensions in activities, and how these might be resolved. They should ask questions such as – ‘What do the groups and software solutions that ‘work’ and are developing in line with expectations have in common?’ ‘What is going wrong or is absent in the examples where change is not occurring or is proving difficult?’ Positive examples may often point the way to possible resolutions of contradictions – by redesigning the tool, or by taking steps to change working practices.

#### 1b - Elicit

“Your pattern language is the sum total of *your* knowledge of how to build.” (Alexander, 1979, p.203.) In a sense all designers use established patterns as a resource whenever they bring their design knowledge, the sum of their experience, to bear on a new problem. Users have domain knowledge, and proven solutions to problems that they use again and again. Eliciting the expert knowledge of developers and users in a pattern writing workshop is another way of getting started with the creation of a pattern language, in order that experiences can be shared and used by others.

#### 1c – Envision

Patterns of future activities – like future scenarios – can be written as a tool for investigating how activity is going to change in order to realise new outcomes, and how activity systems should be designed to support it. Patterns in this sense are experimental and should be written collaboratively with users in order to involve them in discussions of how their work could develop.

The pattern INFORMATION AS COMMON PROPERTY sets out a space for development between an envisioned future activity system and what is happening in the present. When GreenFam adopted the policy of working more collaboratively with information, patterns could have been written to represent how this change could be realised. In fact the developers learned through a process of trial and error, implementing collaborative information spaces, some of which were more successful than others. Future patterns have to be realistic and achievable, and so must be grounded in a thorough understanding of how work is done in the current situation.

## 2 - Create a Language

Writing patterns requires no special skills other than the ability to think and write clearly, although this does not mean that it is easy or straightforward to do. Unlike other system specification techniques, the fact that higher level patterns need no specialist technical knowledge means that users are likely to be just as skilled at writing them as developers (if not more so). For that reason they are a good technique for involving users in a design project.

Alexander and other writers (e.g. Dearden *et al*, 2002) see a pattern language as a tool to empower users to participate in designing, but we would recommend going one step further and involving users in actually writing patterns. The object-oriented software community has developed the practice of writers’ workshops, where writers present their patterns for discussion and critique, in order to refine them. Workshops are lead by facilitators who are experienced pattern writers. Users and developers could adopt this practice and co-write workplace patterns in a participatory workshop.

#### 2a - Write & 2b – Combine

These two actions are performed together. Patterns are written and then combined with the other patterns to form a unified *pattern language*. This is done by explicitly defining how they are related to higher and lower level patterns in the introductory paragraph and conclusion of the pattern (see *Table II*). Pattern writers will probably have to go through several iterations of this process in order to formulate a pattern that is sufficiently robust to be applied and tested in practice. Factors such as writing style, typography and layout are important for the clarity of the pattern: it is a visual representation as well as a text. A pattern template – adapted from Alexander – is shown in the Appendix, *Table II*; this specifies the content and form of a pattern. This template has been used to construct the four patterns that are also included in the Appendix, together with an example of how to represent a pattern language as a map or network in order to show the associations between its patterns (*Figure 2*).

#### Notes on the Pattern Template

- Naming is part of designing. The pattern name should express exactly what the pattern does in an emotive

way: this helps users to remember it and facilitates use of the pattern language.

- The ranking of a pattern – whether it has two, one or no asterisks – conveys the writers’ confidence in the validity of the pattern and the extent to which it has been verified through application.
- The illustration acts as a mnemonic and communicates how the solution can be implemented in a real example. Because of the intangible nature of systems design (unlike architecture and building) it might be necessary to use a metaphorical illustration for some patterns: in the pattern INFORMATION AS COMMON PROPERTY the picture of the library is a metaphor for a shared information space, with its dual character of information repository and space for collaborative work.
- Alexander says that “If you can’t draw a diagram of it, it isn’t a pattern” (1979, p.267). We have used two types of diagram: variations on the conventional activity triangle (Engestrom, 1987), and UML use cases to specify the functionality of lower level patterns. Other modelling techniques that could be used are: rich pictures, story boards, UML activity diagrams (for patterns of workflow procedures), screen shots, paper sketches of interface objects. The only restriction is that the diagrams should be fairly easy for anyone to understand and draw, and not limited to those with technical know how.

### Application in Design

The following actions cover the application of a pattern language in design projects and using this experience to evolve the language.

#### 3 - Generate a project language

The pattern language is used to generate a range of possible concrete solutions for specific design problems. The whole pattern language will not be required in any project, but a subset of patterns from it to fit the project’s scope. Generating a sub-language for a project is the key design activity – this amounts to specifying part of the conceptual design of the activity system by selecting the patterns that are applicable. There are two possible actions for configuring a pattern language for a project.

##### 3a - Source

A growing number of pattern languages for interaction design and software design are currently being published. Developers can source patterns from any of these collections in order to augment their own languages if they need additional patterns for a project.

##### 3b - Select

The main action is to select patterns from designers’ own pattern language. Selecting patterns is designing and this is therefore the most critical part of a project. Selection should be done in a workshop involving members of the development team or, if appropriate, developers and users.

The workshop should be facilitated by someone who is experienced in using patterns; has both domain and technical knowledge; and understands the underlying concepts of AT embodied in the patterns. The job of the designer-facilitator will include interpreting the patterns and giving examples of how they have been used.

- First look at the pattern language map and tick off all the ones that are relevant to the project. Think about the tensions that are likely to occur and the changes that will be needed as a result of introducing new software: find patterns that have helped to resolve these contradictions in the same context.
- Then select a top level pattern that describes the scope and motive of the activity in the way that our pattern INFORMATION AS COMMON PROPERTY defined the activity and motivation of the GreenFam project. This pattern, when it becomes part of the project language, will not be related to any higher level patterns as it sets the overall context.
- Finally select from the available lower level patterns top-down, until the lowest level activity patterns that are required have been identified.

System design workshop techniques such as affinity diagrams or card-sorting can be used to physically interact with the patterns. Following Dearden *et al* (2002) we recommend printing patterns out on cards to facilitate handling them. The cards should have a summary of the pattern on the front (identifier, name, visual representation, context, short contradiction and resolution statements, references to lower patterns); the full pattern on the back.

Patterns are not intended to be a ‘one-size fits all’, off the peg solution to a problem, but a starting point to stimulate design discussions. In this sense they fill the role of what Bodker and Christiansen call “springboards” for design (1997). It is more than likely that new patterns will be needed in any project - the pattern design workshop might have to adapt patterns to the specific context of the project, write new ones and prune patterns that have become obsolete. As well as the pattern map and the patterns themselves, a template for editing and writing patterns is an essential workshop tool.

#### 4 - Apply

The last three actions – Apply, Verify and Evolve - form part of an iterative cycle of testing patterns through use, evaluating the outcomes and using this experience to evolve the pattern language.

Alexander has a number of principles for building construction that are relevant to software design and even mirror methods that are used by practitioners. One of these is the principle of “piecemeal growth” (Alexander *et al*, 1975): developing a large complex of buildings in usable increments as opposed to “large lump development”. This principle resonates with the approach of rapid application development and agile methods in SE. He also has a

pattern for a process of GRADUAL STIFFENING (1977, p.962) for the construction of buildings. This pattern sets out a rule for “[...] weaving a structure which starts out globally complete, but flimsy; then gradually making it stiffer but still rather flimsy; and only finally making it completely stiff and strong.” The rationale for this pattern is that not all aspects of a design can be pre-specified: many changes may have to be made as a design is built and ‘flimsiness’ in the earlier stages facilitates this. Iterative development, starting off with flimsy and easily changed design models such as paper prototypes, followed by software prototyping, is a commonly practiced way of achieving ‘gradual stiffening’ in the design of systems.

### 5 - Verify

As patterns are implemented in concrete prototypes and new systems they can be verified. The ongoing evaluation of patterns is essential if the pattern language is to develop in line with changes in the field of design. Patterns can be verified not just by observing situations where implementation of a pattern has been a success and helped to support change. It can also be done through observation of counter-examples where a pattern is lacking and contradictions are apparent.

The facilitator patterns have been verified using both approaches. At GreenFam our evaluation of Notes databases in use found several instances of facilitators using the tools defined in the EMAIL ALERTS and ONE CLICK HYPERLINK patterns. In these cases participation in the shared information spaces was much better than the examples where there was no facilitator, or where there was no encouragement of users in this way.

We also had the opportunity of testing our findings in another project, to set up a shared information space for a research group at our University. We used an information space on the University’s student collaborative learning environment as a low cost solution, but found that it did not integrate the researchers’ normal email tool, not did it enable hyperlinks to be made to specific documents. Without these technical facilities people who wanted to involve their colleagues in using the shared space did not have the tools that they needed. This example with its contradictions and breakdowns is depicted in *Figure 2*, which should be contrasted with the resolved activity shown in the diagram for the pattern FACILITATORS ARE THE KEY.

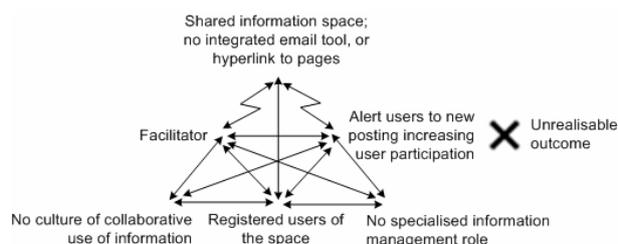


Figure 2. When the facilitator does not have the right tools.

The ranking of patterns by two, one or no asterisks is a way of indicating when a pattern has been validated and the degree of confidence that can be placed in it. Because we have verified the three facilitator patterns in several instances and two different workplaces we rate them fairly highly and feel confident that they should be considered whenever this pattern context is found.

### 6 - Evolve

Finally, developing a pattern language is evolutionary. In order to ensure that the language keeps pace with change its patterns must be refined, revised, and rejected whenever they are applied as design solutions and verified. New patterns will continually be discovered as activity systems develop and the context of design is changed. Even the application of a pattern will have the effect of changing its context somewhat, and this means that the pattern language will be in a continual state of flux. To cope with this dynamism it is important to adopt tools that support management of changes in the language and even automate the production of a pattern map.

### Summary of the Method

This section has described guidelines for the practical application of patterns to the design of activity systems; and for the evolution of a pattern language. Patterns embody the expert knowledge of designers and users, and domain knowledge about how work is carried out in its context. The starting point for those new to patterns is to write patterns based on their own knowledge and experience, and to combine these into an embryo pattern language. Application of patterns involves using the language to generate a subset language specific to the scope of the project; then applying it in design. The techniques of participatory workshops and iterative development, starting with low-cost prototyping techniques that facilitate changes to the specification are particularly appropriate. Finally patterns should be evaluated in use, verified and this knowledge used to develop and update the pattern language.

### CONCLUSIONS: VALIDATION OF THE APPROACH

It is proposed that the method of ‘activity patterns’ is a versatile tool that could be used throughout the cycles of use, evaluation and design (Bannon, 1996); as a participatory technique bridging the conceptual and semantic gap between developers and users; as a tool for establishing shared meanings in a multi-functional or distributed project team; to represent work and its organisational and technical interfaces; and to design future mediated, collective work. This approach was the outcome of a program of fieldwork, with the patterns being identified through an evaluation of information sharing tools, as use evolved over a period of time and new practices emerged. It can therefore be stated with some confidence that capturing the lessons of evaluation as patterns, in order to use these to inform future design, is a useful and valid approach. It has not been possible to test

the method at GreenFam, as the research project has now ended.

The appeal of appropriating a tool, rather than designing something completely novel, is that it is possible to benefit from existing resources and a community of users. Developers are not being asked to adopt a completely new method that is not yet part of the practice of software development. With other pattern user-developers in the domains of SE, HCI and participatory design, it is possible to refer to their experiences to validate the approach. However, it must be acknowledged that little evidence exists about how – or indeed whether – patterns are being used in real-world system development practice. Many of the books which are currently being published describe patterns as ‘good design’ heuristics, distilled from professional experience, without reflecting on the processes of developing and applying patterns (for example Adolph *et al*, 2003; Graham, 2003).

The patterns approach outlined in this paper proposes a novel application for patterns - to model computer-supported cooperative work at the level of activity and actions. Much of the recent interest in patterns has been at the level of interaction and software design, although a wider role in socio-technical design has been proposed by some authors (Erickson, 2000a; Martin *et al*, 2001; Herrmann *et al*, 2003). The approach is therefore still largely untested and needs to be trialed and refined in an industrial setting in order to evaluate its potential.

#### ACKNOWLEDGMENTS

The research was funded by the Engineering and Physical Sciences Research Council, EPSRC Award Reference 95306394 (1995-1998) and the Interactive Technologies Research Group at the University of Brighton (2001-2002). Thanks to all at GreenFam who contributed to the study, particularly Robin van Koert, my collaborator on the evaluation research. Bob Seago, the University of Brighton photographer, took the photos that illustrate the patterns. Thanks also to the workshop participants for their feedback, and especially to Lars Taxén for his helpful reviews.

#### REFERENCES

- Adolph, S., P. Bramble, A. Cockburn and A. Pols, (2003). *Patterns for Effective Use Cases*. Boston, MA: Addison-Wesley.
- Alexander, C., M. Silverstein, S. Angel, S. Ishikawa and D. Abrams, (1975). *The Oregon Experiment*. New York: Oxford University Press.
- Alexander, C., S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King and S. Angel, (1977). *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press.
- Alexander, C., (1979). *The Timeless Way of Building*. New York: Oxford University Press.
- Alexander, C., H. Davis, J. Martinez and D. Corner, (1985). *The Production of Houses*. New York: Oxford University Press.
- Bannon, L., (1996). Use, Design and Evaluation: Steps Towards an Integration. In: Shapiro, D., M. Tauber and R. Traummuller, Eds., *The Design of Computer Supported Cooperative Work and Groupware Systems*. Amsterdam: Elsevier. 423-443.
- Bannon, L. and S. Bodker, (1997). Constructing Common Information Spaces. In: Hughes, J., W. Prinz, T. Rodden and K. Schmidt, Eds., *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work*. ECSCW '97, Lancaster, England. Dordrecht: Kluwer Academic Publishers. 81-96.
- Bertelsen, O. and S. Bodker, (2002). Discontinuities. In: Floyd, C., Y. Dittrich and R. Klischewski, Eds., *Social Thinking, Software practice*. Cambridge, MA: MIT Press. 409-424.
- Bertelsen, O. and S. Bodker, (2003). Activity Theory. In: Carroll, J. M., Ed. *HCI Models, Theories and Frameworks: Towards an Interdisciplinary Science*. San Francisco, CA: Morgan Kaufmann Publishers. 291-314.
- Bodker, S. and E. Christiansen, (1997). Scenarios as Springboards in CSCW Design. In: Bowker, G., S. L. Star, W. Turner and L. Gasser, Eds., *Social Science, Technical Systems and Cooperative Work: Beyond the Great Divide*. Mahwah, NJ: Lawrence Erlbaum Associates. 217-233.
- Borchers, J., (2001a). *A Pattern Approach to Interaction Design*. Chichester: John Wiley and Sons.
- Borchers, J., (2001b). A Pattern Approach to Interaction Design. *AI & Society* 15 (4): 359-376.
- Carroll, J. M., (2000). *Making Use: Scenario-Based Design of Human-Computer Interaction*. Cambridge, MA: The MIT Press.
- Checkland, P., (1981). *Systems Thinking, Systems Practice*. Chichester: John Wiley and Sons.
- Checkland, Peter and Jim Scholes, (1996). *Soft Systems Methodology in Action*. Chichester: John Wiley and Sons.
- Dearden, A., J. Finlay, E. Allgar and B. McManus, (2002). Using Pattern Languages in Participatory Design. In: Binder, T., J. Gregory and I. Wagner, Eds., *Proceedings of the Participatory Design Conference*. PDC 2002, Malmo, Sweden. Palo Alto, CA: CPSR. 104-113.
- Engestrom, Y., (1987). *Learning by Expanding: An Activity-Theoretical Approach to Developmental Work Research*. Helsinki: Orienta-Konsultit Oy.
- Erickson, T., (2000a). Supporting Interdisciplinary Design: Towards Pattern Languages for Workplaces. In: Luff, P., J. Hindmarsh and C. Heath, Eds., *Workplace Studies: Recovering Work Practice and Informing System Design*. Cambridge: Cambridge University Press. 252-261.

- Erickson, T., (2000b). *Lingua Francas for Design: Sacred Places and Pattern Languages*. In: Boyarski, D. and W. Kellogg, Eds., *Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. DIS 2000, Brooklyn, NY. New York: ACM Press. 357-368.
- Gamma, E., R. Helm, R. Johnson and J. Vilissides, (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.
- Graham, I., (2003). *A Pattern Language for Web Usability*. London: Addison-Wesley.
- Grudin, J., (1988). Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. In: *Proceedings of the Conference on Computer-Supported Cooperative Work*. CSCW '88, Portland, Oregon. New York: ACM Press. 85-93.
- Guy, E. S., (in preparation). *Patterns of Activity: Modelling Computer Supported Cooperative Work*. Unpublished PhD thesis. School of Computing, Mathematical and Information Sciences, University of Brighton, UK.
- Harris, S. R., (2004). Morphological Analysis of HCI Video data Using Activity Theory. In: *HCI 2004: Design for Life*. HCI 2004, Leeds, UK. British HCI Group. 41-44.
- Herrmann, T., M. Hoffmann, I. Jahnke, A. Kienle, G. Kunau, K-U. Loser and N. Menold, (2003). Concepts for Usable Patterns of Groupware Applications. In: Pendergast, M., K. Schmidt, C. Simone and M. Tremaine, Eds., *Proceedings of the 2003 International SIGGROUP Conference on Supporting Cooperative Work*. Florida, USA. New York: ACM Press. 349-358.
- Leont'ev, A. N., (1979). The Problem of Activity in Psychology. In: Wertsch, J. V., Ed. *The Concept of Activity in Soviet Psychology*. New York: M. E. Sharp Inc. 37-71.
- Leont'ev, A. N., (1981). *Problems of the Development of the Mind*. Moscow: Progress Publishers.
- Leont'ev, A.N., (1978). *Activity, Consciousness and Personality*. New Jersey: Prentice-Hall Inc.
- Lilienfeld, R., (1978). *The Rise of Systems Theory: An Ideological Analysis*. Chichester: John Wiley and Sons.
- Martin, D., T. Rodden, M. Rouncefield, I. Sommerville and S. Viller, (2001). Finding Patterns in the Fieldwork. In: Prinz, W., M. Jarke, Y. Rogers, K. Schmidt and V. Wulf, Eds., *Proceedings of the Seventh European Conference on Computer-Supported Cooperative Work*. ECSCW 2001, Bonn, Germany. Dordrecht: Kluwer Academic Publishers. 39-57.
- Nardi, B., Ed. (1996). *Context and Consciousness: Activity Theory and Human-Computer Interaction*. Cambridge, MA: MIT Press.
- van Duyne, D., J. Landay and J. Hong, (2003). *The Design of Sites: Principles, Processes and Patterns for Crafting a Customer-centered Web Experience*. Boston, MA: Addison-Wesley.
- Vygotsky, L. S., (1962). *Thought and Language*. Cambridge, MA: MIT Press.
- Wartofsky, M. W., (1979). *Models: Representation and the Scientific Understanding*. Boston: D. Reidel Publishing Company.

## Appendix: Four Patterns For Information Sharing

In order to illustrate how to write patterns and integrate them into a pattern language, we have chosen a subset of four patterns from the GreenFam pattern language. These are represented in the pattern map shown in Figure 3, along with some of the other patterns to which they are related: the illustrated patterns are highlighted in grey. The arrows show the associations between patterns, with the arrow pointing from the higher level to the lower level pattern. The top level pattern is INFORMATION AS COMMON PROPERTY. The template for writing patterns is shown in Table II. This is based on Alexander's pattern form, with the changes that have been discussed in the paper. This is followed by the four patterns –

- INFORMATION AS COMMON PROPERTY
- FACILITATORS ARE THE KEY
- EMAIL ALERTS
- ONE CLICK HYPERLINK

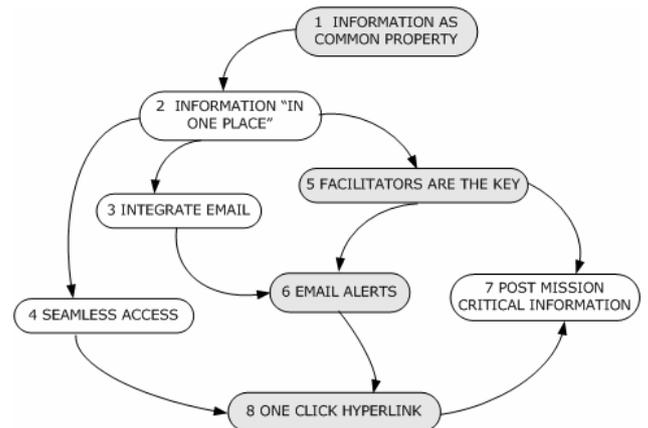


Figure 3. Subset of the pattern language highlighting the illustrated patterns.

Description of content	Formatting instructions
<Identifier> <PATTERN NAME> <Validity ranking>	Numeric identifier; pattern name in small capitals; if ranked use 1 or 2 asterisks * / **. Centre text.
<Illustration of a concrete instantiation of the pattern>	
...<introduction: sets the <i>context</i> of higher level patterns which this pattern helps to complete>	The introduction begins with ellipsis marks ... followed by the first word in lowercase. Higher level pattern names are in small capitals followed by the pattern identifier in brackets e.g. FACILITATORS ARE THE KEY (4). Justify margins.
❖ ❖ ❖	Symbols divide the introduction from the contradiction summary: centre.
<Concise summary of the contradiction>	<b>Bold</b> ; justify; indent first line.
<Detailed description of the contradiction that the pattern resolves, with empirical examples of how it can be manifested. May include illustrations of examples and counter-examples.>	Justify; indent first line of the new paragraph.
Therefore:	Indent.
<Resolution: expressed in the form of an instruction.>	<b>Bold</b> ; justify; indent first line.
<Model of the resolution in an appropriate diagrammatic form>	Centre.
❖ ❖ ❖	Symbols indicate the end of the body of the pattern: centre.
<End: <i>references</i> to the lower level patterns which help to complete this pattern> ...	Lower level pattern names are in small capitals followed by the pattern identifier in brackets e.g. EMAIL ALERTS (7). Ends with ellipsis marks ... Justify; indent first line.

Table II. Pattern template based on Alexander's model.

I INFORMATION AS COMMON PROPERTY



<sup>3</sup>Many organisations that work intensively with information have the goal of developing a culture and practice of treating information as common property, rather than as the private resource of departments and individuals. To achieve this goal they have to change attitudes and established ways of working, and introduce new collaborative tools to support information sharing.

<sup>3</sup> As a top level pattern INFORMATION AS COMMON PROPERTY starts with the contradiction summary rather than a list of related higher level patterns.

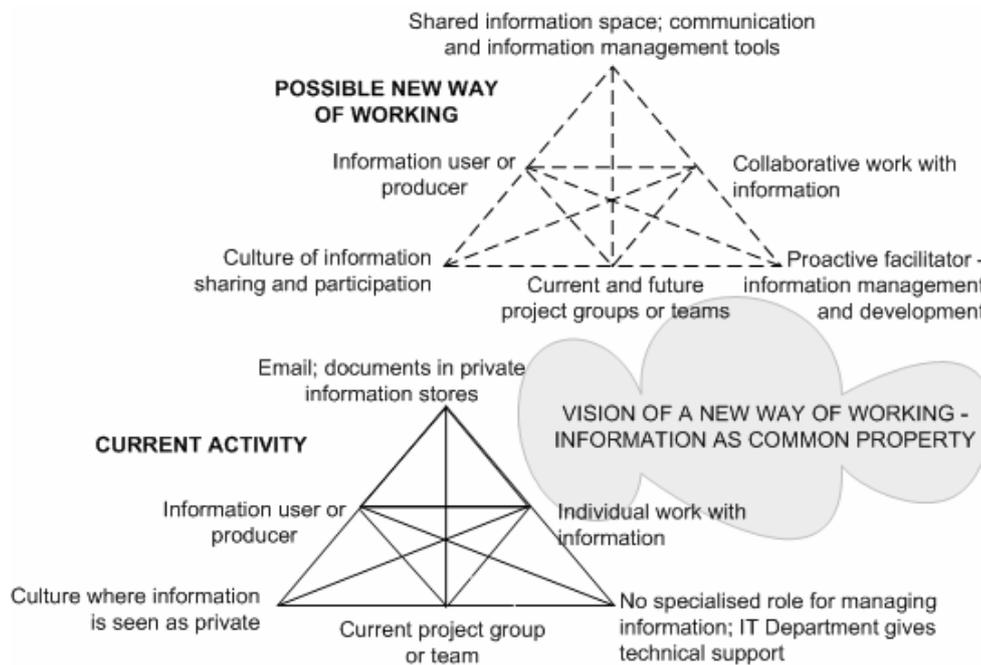
In many organisations a practice of treating information as the private resource of the organisational unit or individual who ‘owns’ it prevails. This prevents the sharing of information freely among everybody who needs access to it in order to carry out their work. The practice might have been established over a number of years and will be supported by a culture which rationalises it in terms of the need for control to prevent unauthorised access, and to maintain the accuracy and consistency of information.

In information intensive organisations the huge amount of information that has to be dealt with – articulated as ‘information overload’ – may make the people who work there reluctant to share information, or to use shared information spaces. Their immediate reaction to a proposal that information is treated as a shared resource might be that this will overload them with even more information and work from which they will not benefit.

Computer tools currently in use for communicating and managing information may not facilitate information sharing, and may fragment it into many private spaces. One example is email where messages are kept in private mailboxes, rather than being stored in a shared archive. Another is personal computers, where information resources are created, stored and managed on the computer’s hard drive: there is an overhead of additional work required to place these resources in a shared space. Another example is the group practice of putting document files in shared folders without proper tools for information management, where it may be difficult to search and retrieve information.

Therefore:

**Put information in shared spaces with integrated communication and information management tools, where people can work collaboratively as well as carrying out their own work. Take steps to change organisational culture by providing incentives for information sharing.**



*The change to an information sharing culture and practice*



To facilitate collaborative work and encourage an information sharing culture put INFORMATION “IN ONE PLACE” (2) ...



... cooperative work involving many people working in different locations and times is often coordinated through the use of shared information spaces – a practice that treats INFORMATION AS COMMON PROPERTY (1). By putting INFORMATION “IN ONE PLACE” (2) people have access to all the information resources they need to do a job.



**Users often do not readily adopt new tools for information sharing when placing information in common involves additional work and means they must change the way they work.**

When new collaborative information sharing tools are introduced to workplaces they will require changes in existing ways of working. For example –

- A new organisational policy may mandate that information is treated as a shared resource, while departments, work groups or individuals have been used to keeping their information resources to themselves.
- The person doing the additional work of posting information in the space is often not the person who directly benefits from using it (Grudin, 1988), so that busy users have no incentive to participate.
- When new tools for collaborative information sharing are implemented they will have to compete with tried and tested ways of working with information such as email, and other communicative practices which are embedded in the culture of the workplace.

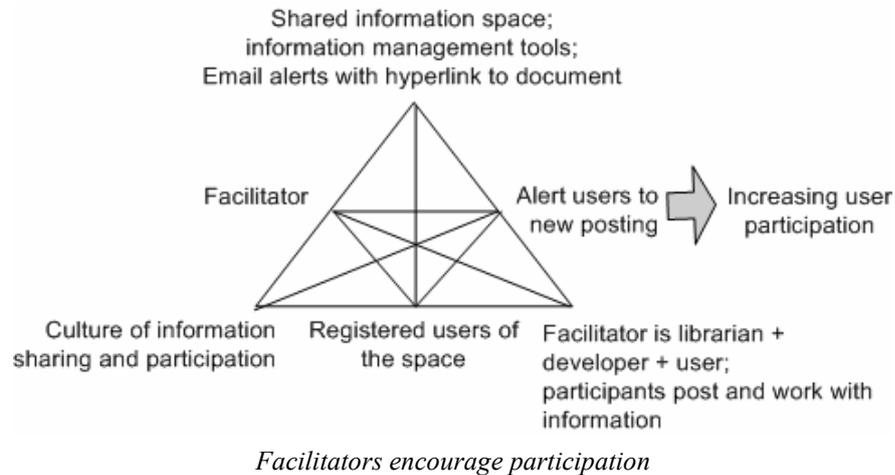
Where information spaces are adopted and used as intended it is often because an enthusiastic and motivated individual volunteers to take on the role of facilitation, encouraging the adoption of the new tool in their workgroup. They may do this by posting important working documents in the new information space, whether instead of, or in addition to, communicating them through established modes; by sending an email to the group alerting them to the new document on the database and directing them to it by means of a hyperlink; by organising the space – archiving out of date documents, evolving use-centred classification schemas, and repairing mis-classified information; by encouraging and exhorting the group to use the information space and thereby nurturing the emergence of a new culture of information sharing.

Where a facilitator does not emerge the space may never be used effectively, as the workgroup continues to work in the old ways. This will also happen if the facilitator does not have the tools to carry out their role, or the right skills. Individuals who take on this role voluntarily may quickly become demoralised if the workgroup does not begin to use the space.

The new role of facilitator requires additional work, for example communicating the same information in different ways and sending email alerts to users. In order to do this effectively the facilitator needs the right tools for the job.

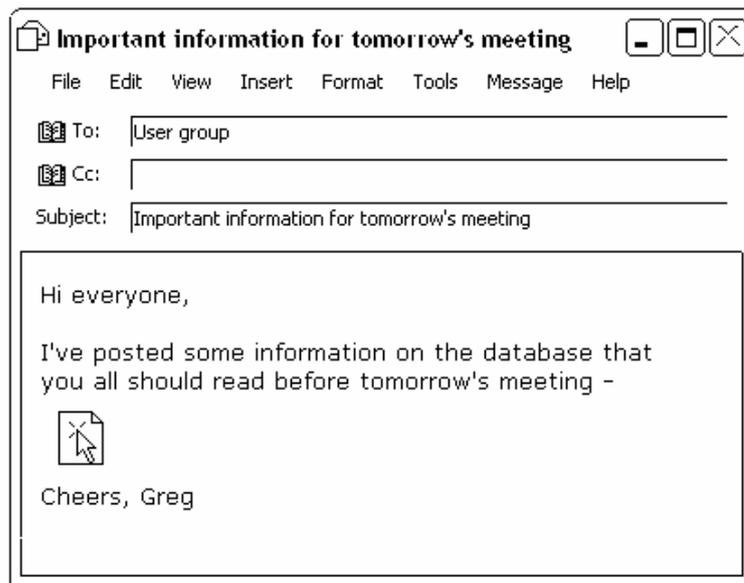
Therefore:

**Appoint a facilitator for the database and ensure that this role is recognised as a part of their job and valued. The facilitator must be allocated the necessary time to do the job properly and given training in the technical and information management skills required. They will be responsible for seeing that the objectives for the space are achieved; encouraging participation by registered users; and for information management and development of the space. The information space should have integrated tools that facilitators need to do the job.**



Facilitators alert users to new material they have posted by means of EMAIL ALERTS (6), which draw them into the space and open the required document by means of a ONE CLICK HYPERLINK (8). Facilitators POST MISSION CRITICAL INFORMATION (7) that everybody needs to read to carry out their work – such as agendas for meetings – so that users have to respond to the email alert and visit the space ...

#### 6 EMAIL ALERTS \*\*



... Where organisations want to change from an email paradigm to a shared information paradigm and put INFORMATION “IN ONE PLACE” (2) with tools which INTEGRATE EMAIL (3), FACILITATORS ARE THE KEY (5) to encouraging users to adopt the new way of working.



**It can be difficult to get users to change from using email to communicate information, to placing it in a shared information space where everybody can access it. Facilitators need the right tools to involve users and encourage participation.**

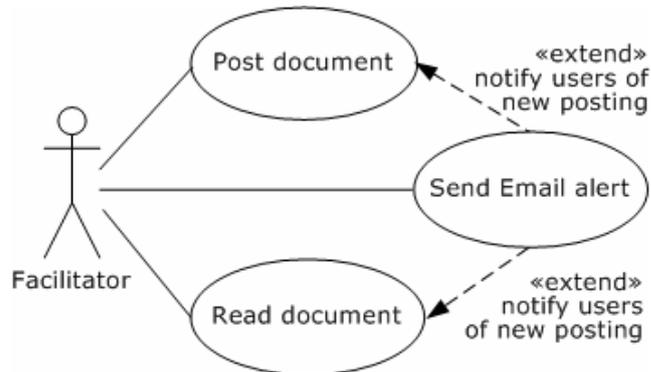
Email is a tool used to coordinate work and communicate information that is generally popular with users. It has the drawback that email messages are stored in personal mailboxes where the information cannot be shared. When new tools for collaborative information sharing are introduced to organizations they will have to compete with tried and tested ways of communication such as email, which are embedded in the culture and practice of the workplace and which are easy and quick to use.

If email is integrated in the information space the facilitator can easily email users after she has posted a new document in the space, especially if it contains mission critical information that they need to read. Or maybe she has just read an interesting document that someone else has posted and wants to draw it to the attention of one or more users who may not yet have got into the habit of checking the space regularly for new information.

Users who are accustomed to using email in their day to day work will read the message and, by means of the hyperlink to the document, are transported directly to the shared information space. By the means of email alerts users are accustomed to the space, which gradually becomes more widely used.

Therefore:

**Make a virtue of the popularity of email by using it to alert users about new information on the database and directing them to it by means of a hyperlink.**



*Use cases for EMAIL ALERTS*



The email message contains a ONE CLICK HYPERLINK (8): a mouse click on the hyperlink icon takes the user directly to the relevant document in the database ...

## 8 ONE CLICK HYPERLINK\*\*



... the facilitator of the shared information space sends EMAIL ALERTS (6) to users, which contain a hyperlink to a document posted in the space. Users must automatically be logged on to the space when they log onto their computer, so that there is SEAMLESS ACCESS (4) to the space when they click on the link.

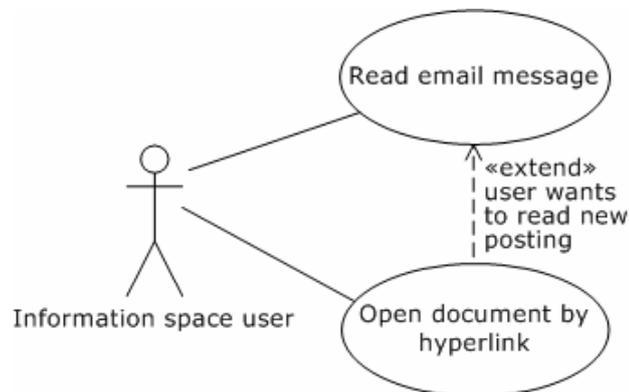


**It can be difficult to get people to adopt a new tool such as a shared information space in their daily work, if it requires them changing established ways of working.**

It can take along time before users get into the habit of using a new tool, particularly when they already have tools such as email that are established in their daily routine and do not see immediate benefits of changing the way that they work. If the space is perceived as being “yet another software tool” that they are supposed to learn how to use they may not log onto the space every day to check for new information. One of the jobs of the facilitator is to get users into the habit of visiting the space regularly and to make them familiar with it: this is a first step towards encouraging active participation. A hyperlink in an email alert message can be used to direct users to a document in the shared space. That way they do not have to remember to log into the space and check it each day – by one click use of the space becomes seamless with reading an email.

Therefore:

**Put a hyperlink to a document in the shared information space into an email alert, so that when the user reads the message they can click on the link and be transported directly to the space.**



*Use cases for ONE CLICK HYPERLINK*



As a further incentive to use the space the facilitator should POST MISSION CRITICAL INFORMATION (10) and email users a hyperlink to the document ...

# Systemic-Structural Activity Analysis of HCI Video Data

Steven Robert Harris  
Hypermedia Research Unit  
University of Glamorgan  
Wales, UK. CF37  
+44 1443 483276  
srharris@glam.ac.uk

## ABSTRACT

This paper describes methods for the structural (morphological) and functional analysis of HCI video data through the isolation and classification of tasks; the isolation and classification of actions; and the use of self-regulation models to study goal-directed activity during task performance. These methods were developed and applied within the context of a long-term, design-oriented research project involving field studies of collaborative computer-mediated activity by non-professional users in an educational setting. The conceptual basis for the methods is provided by the systemic-structural theory of activity, a modern synthesis of recent research in applied activity theory specifically tailored to the study and design of human work activity.

## Keywords

Video analysis, IT-design, activity theory, morphological & functional analysis, field data, systemic-structural theory of activity.

## 1. INTRODUCTION

### 1.1 General Introduction and Initial Academic Reflection

Approaches to human-computer interaction (HCI) and information technology (IT) design based on activity theory understand computer artifacts as mediators of human work activity, and attempt to take account of the reality that the (actual or proposed) use of interactive systems always takes place in specific, complex, and historically developing sociocultural and technical contexts.

Inasmuch as video recording is capable of capturing both the fine detail of individual sequences of IT use and the broader physical and sociocultural context in which they take place, it can provide an invaluable resource for activity-theoretical IT-design. Video allows the repeated review of complex and fleeting events (in both real-world and laboratory settings) at varying levels of granularity, bringing the opportunity to check and amend interpretations of observed user actions in the light of data from other sources (log files, interviews, verbal protocols, etc.) or new analytic insights (Jordan & Henderson, 1995). The possibility of making unobtrusive (Blomberg, 1993) or unattended (Bauersfeld & Halgren, 1996) video recordings also helps to minimize the impact of observation on the situations under study, an important

consideration when studying collaborative work activity in the field.

Bødker's seminal work on focus-shift analysis (Bødker, 1996) was the first attempt to develop video analysis methods using activity theory. Her focus-shift approach formulates interaction analyses (Suchman & Trigg, 1991; Trigg *et al.*, 1991) based on interpretations of general activity theory developed within the "Scandinavian School" of IT-design (Bertelsen & Bødker, 2003). This article aims to complement and extend that work by introducing methods based on the systemic-structural theory of activity (SSTA), a distinctive, modern, activity-theoretical approach specifically oriented toward the solution of practical problems in ergonomics, engineering psychology, and education (Bedny & Meister, 1997; Bedny, 1998; Bedny *et al.*, 2000; Bedny, 2004). Methods based on SSTA employ a carefully defined, standardized and unified terminology; clearly distinguish between various stages, levels, objects, and units of activity analysis; and, in addition to the qualitative descriptive analyses currently associated with activity-theoretical IT-design, allow the symbolic and quantitative modeling of computer-mediated work activity.

### 1.2 Scope, Applications, Outcomes and Analytical Perspective of the Methods Described

The focus of the methods outlined in this paper is on those stages of video analysis following the initial capture, logging and archiving of data, comprising:

- initial selection of data for analysis;
- the transcription of video sequences;
- the development of general time structures of use activity;
- the isolation & classification of tasks;
- using video data as a basis for the morphological (structural) analysis of use activity;
- using video data as a basis for the functional analysis of use activity.

The methods described can be used to support those aspects of the IT-design process where video data analysis is appropriate: e.g. requirements gathering, cooperative prototyping, developing support materials, usability testing, and evaluation. The various techniques are closely interrelated and will often be used recursively during the design process, i.e. the development of a detailed structural description of activity may require revisiting and refining

the general time structure or task classification stage of analysis.

The methods are suitable for use by researchers, analysts, and designers with some background in cognitive task analysis; their application also requires basic familiarity with the terms and principles of systemic-structural activity theory. To this end, concise definitions of relevant SSTA terms and principles are given in the text where appropriate.

#### *Outcomes*

The immediate outcomes of the methods are a set of integrated descriptions - in textual, symbolic and diagrammatic forms, and at varying levels of detail and abstraction - of the structure and function of activity during computer-mediated task performance, based on the source data. The purpose of these descriptions is to support analysts in developing a detailed understanding of the relationship between the computer artifacts in use; task goals and conditions; sociocultural, physical, and technical context; and users' actual and/or possible strategies of activity in the work process. Depending on the stage and nature of the IT-design lifecycle to which they are applied, these descriptions may prove sufficient for the purposes of the project in hand. However, they may also be used as a basis for further techniques of systemic-structural activity analysis, such as the development of detailed models of the temporal relationships between motor and cognitive actions during task performance, and the quantitative analysis of task complexity. Although these latter methods can only be briefly mentioned here, more detailed information (procedures, syntax, formulae, etc.) can be found in Bedny & Meister (1997).

#### *Analytical Perspective*

Activity theory offers a holistic conceptual framework within which the multidimensional data produced by video-based observation studies may be integrated, interpreted, and, to some extent, generalized (see Sections 2.1 & 2.2 below). Analyses within this framework can be undertaken from the point of view of three principal, and complementary, perspectives (Bedny & Karwowski, 2004b):

- the sociocultural, which studies culture, community, and historicity, and is concerned with situating and characterizing computer-mediated activity, both historically and in relation to other activities;
- the objectively-logical, which studies tasks, tools, work processes, and results, and is concerned with inputs, products, and the transformation process;
- the individual-psychological, which is concerned with actions, operations, and self-regulation, and studies subject-object relationships.

The methods described in this paper mainly make use of the latter perspective, i.e. the individual-psychological, which is currently under-represented in the English-language activity-theoretical IT-design literature.

The individual-psychological perspective on activity analysis encompasses the following methods:

- informational or cognitive analysis, which studies the various cognitive *processes* (perceptual, mnemonic, etc.) involved in task performance;
- morphological analysis, which studies activity during task performance as a *structure* made up of goal-directed (conscious) actions and their (unconscious) operational aspects;
- functional analysis, which studies activity during task performance a goal-directed, self-regulating *functional system* comprising function blocks (motive, goal, etc.) and their interrelationships through feed-forward and feedback mechanisms;
- and parametrical analysis, which studies various *parameters* of activity such as time on task, number of clicks, keystrokes, errors, etc.

In this paper, the main focus is on introducing methods for the structural and functional analyses of video data. Several methods for the cognitive process analysis and parametrical (e.g. error) analysis of HCI tasks are already well-known to IT-designers (see e.g. Card *et al.*, 1983; Norman & Draper, 1986; Nielsen, 1994; Shneiderman, 1998; Preece *et al.*, 1994, pp.417- 429; Diaper & Stanton, 2003).

It is important to note that the three perspectives of activity theory outlined above simply emphasize different ways of approaching the same complex object of study, i.e. human work activity. They are tightly interrelated and can never be considered entirely in isolation from each other. While the outputs from individual-psychological analyses can be especially useful for addressing issues such as hardware and software design, support, and training, they become more effective when coordinated with understandings of the communities, objects, tools and symbols involved in the work process as revealed by sociocultural and objectively-logical analyses.

## **2. THEORETICAL & EMPIRICAL BACKGROUND OF THE METHODS**

### **2.1 Theoretical Background**

Activity theory is a psychological approach with a long history of development in the former Soviet Union and other Eastern Bloc countries. Its philosophical and general psychological origins are associated with the work of S. L. Rubinshtein (1935, 1973) and A. N. Leont'ev (1977, 1978, 1981). One of the principal foundations of AT is the cultural-historical theory of the development of higher mental functions proposed by L.S. Vygotsky (1978, 1986). There has been a tendency for some English-language interpreters of activity theory (e. g. Cole 1996, Nardi 1996a, Engeström, 2000) to largely limit AT to the cultural-historical paradigm; in activity-theoretical IT-design this has contributed to a current situation where there is an under-utilization of those traditions and viewpoints - both within AT and in the "activity approach"<sup>1</sup> generally - whose roots lie outside the Vygotskian tradition. It is only now becoming recognized that, there have always been other important directions in AT - for example, in physiology and psychophysiology - which have played a role in shaping it into its current form

as a fundamental scientific approach which encompasses a variety of different schools and theoretical concepts.

The methods outlined in this paper are derived mainly from one distinctive school in AT, which is concerned with developing and applying the systemic-structural theory of activity (SSTA). This represents a modern systems approach which has arisen in response to the practical demands of applying AT to design problems in ergonomics, engineering psychology, training, and education (Bedny & Meister, 1997; Bedny, 1998; Bedny et al., 2000; Bedny, 2004). Drawing on earlier contributions by researchers such as Bernshtein (1967, 1969), Anokhin (1962, 1969), and others whose work has been of major importance for the development of an *applied* AT, SSTA has developed a unified and standardized set of concepts and terminology from the perspective of the systemic-structural analysis and design of activity (Bedny, 2004).

### 2.1.1 Definition of Activity

In SSTA, collective, culturally and historically situated, artifact-mediated work *activity* - specifically human behavior which always involves some conscious elements - is approached as a logically ordered system of mental (cognitive) and behavioural (motor) actions. Activity is energized by *motives* which arise from needs, and is directed toward the exploration and transformation of external and internal objects (object-oriented activity) and communication and coordination in social interaction (subject-oriented activity). During activity performance, cognition, behavior, and motivation are integrated and organized by mechanisms of self-regulation toward achieving conscious *goals* - more or less precise cognitive representations of the desired future result of activity. Activity has a multi-level organizational structure, and a multi-level self-regulation mechanism involving both conscious and unconscious levels (Bedny & Karwowski, 2004b).

Any human work activity involves four general stages: goal formation, orientation, execution, and evaluation (Bedny & Karwowski, 2003a). As activity unfolds, acting subjects continually reconsider and adjust their goals and behavior strategies in response to changing conditions. An important aspect of this self-regulatory process is the ongoing comparison of the goal with the *result*, i.e. the actual outcome, of activity. If the result of an activity does not coincide with the subject's goal, then she or he must reformulate their strategy for goal achievement, or reformulate the goal itself. In this dynamic process, evaluation of the result is affected by factors such as subjective assessments of task difficulty and significance, subjective standards of successful results, etc. (for a general model of the functional structure of self-regulation see Bedny & Meister, 1997 p. 77).

### 2.1.2 Definition of Action

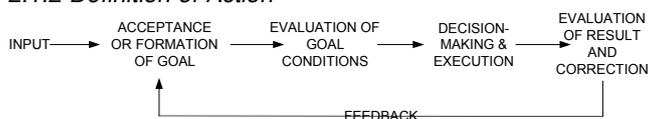


Figure 1. Action as a one-loop system.

An *action* is defined as a relatively complete element of activity that fulfills an intermediate, conscious sub-goal of activity. Individual actions are formulated in terms of their object, tools, goals and subject. Actions are temporal: goal acceptance or goal formulation constitutes their starting point; they conclude when the actual result of action is evaluated in relation to the goal. Actions can be described in terms of a non-linear, recursive loop structure, with multiple feed-forward and feedback interconnections. Figure 1 illustrates this idea with a highly simplified model of goal-oriented action as a one-loop system.

Cognitive and motor actions are in turn composed from smaller units (movements, psychological acts) which may be largely unconscious, and which are often referred to as *operations*.

This understanding of action as the basic unit of activity allows the systematic description of the continual flow of activity during task performance. It should be noted that this definition is formulated at a finer level of granularity and specificity than has been usual in activity-theoretical IT-design (also see discussion in Section 7 below).

## 2.2 Systemic-Structural Activity Analysis and Design

As has been noted, SSTA approaches activity as a complex object of study embodying multiple, distinct aspects, and asserts that multiple methods must be employed to describe any single episode of activity. Thus, the systemic-structural analysis and design of activity makes use of a variety of analytical methods and units of analysis, integrated within a recursive, multi-stage, multi-level framework (Bedny & Karwowski, 2003b).

There are four general stages of systemic-structural activity analysis: (1) qualitative description, (2) algorithmic analysis, (3) time structure analysis, and (4) quantitative (complexity) analysis. These four stages are recursively related; later stages of analysis usually require revisiting earlier descriptions in order to refine or refocus them. Each stage can be carried out from the point of view of some or all of the three perspectives described in Section 1.2 and at different levels of detail or decomposition; depending on the requirements of the research, not all stages may be appropriate or required.

The analytical process begins by focusing on qualitative and parametrical analyses of the general characteristics of the activity of interest: identifying the available means of work, tools and objects; their relationship with possible strategies of work activity; existing constraints on activity performance; social norms and rules; possible stages of object transformation; and changes in the structure of activity during skills acquisition. As activity analysis proceeds, the analytical focus shifts to activity during the solution of some specific problem. That is to say, it is the *task*, understood as some fragment of activity organized around a supervening goal, which becomes the central object of study.

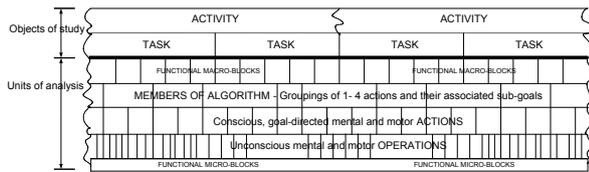


Figure 2. Objects of study and units of analysis in systemic-structural activity analysis.

Figure 2 presents a general scheme of the main structural components of activity as defined by SSTA. It is these structural elements - *activity*, *task*, *action*, *operation*, *function block*, along with a composite unit called *member of algorithm* – that provide the basic analytical objects and units of systemic-structural analysis. When undertaking activity analyses from the individual-psychological perspective (see Section 1.2 above), the first two components of this schema (activity, task) are considered primarily as the *objects* to be studied in activity analyses, the remaining components provide the *units of analysis* employed for the study of those objects.

### 2.3 Empirical Background

The methods described in this paper have been developed and applied in the context of a long-term participatory action research project involving the creative and collaborative use of information technologies by non-professional people from low-income, low-education backgrounds (Harris & Shelswell, 2001; Harris, 2002, 2004; Shelswell, 2004). This research has been carried out in the setting of an Adult Basic Education (ABE) center delivering training in literacy, numeracy, communication, and IT skills to people in the post-industrial valleys region of South Wales, UK. The principal concern of the research is to investigate how various aspects of IT-design either inhibit or support the development of fluent interaction in computer-mediated work and learning activity, and to make (re)design recommendations on the basis of the findings. To date, the project has included four phases of fieldwork. The illustrative examples in this paper are drawn from the first and second phases of fieldwork - a longitudinal study in 2000-2001 followed by a shorter study in 2002 - where adult literacy and numeracy learners were observed and video-recorded as they used IT in collaborative media projects.

In the longitudinal field study, 27 subjects (14 male, 13 female, ages 15-73, median 38), took part in 103 sessions of collaborative computer-mediated activity in which they conceived, planned, and carried out individual and group projects in Web and multimedia authoring, digital video, computer graphics and animation, virtual reality (VR), and computer programming. During this phase of study the primary focus was on developing general, qualitative descriptions of collaborative activity using macro-analytical techniques. Data capture was mainly through participant observation and ethnographic interview. Single-camera video recording was used to opportunistically record instances of interaction breakdown and fluency.

The second phase of field research focused on the detailed analysis of activity during the performance of a single task,

and used video recording as the primary means of data capture. One tutor and 8 learners (2 male, 6 female, ages 37-76, median 63) were observed as they worked over 3 hours to produce paper documents using desktop publishing software. Learners worked collaboratively in three subgroups, each subgroup making use of 1 personal computer (PC). Task requirements were set by the tutor, with learners able to choose from a range of topics and formats for their documents within the stated constraints. A total of 6 digital video data sources were used: 3 tripod-mounted cameras, in conjunction with screen recording software running on each of the 3 PCs in use.

### 3. INITIAL SELECTION OF DATA FOR ANALYSIS

When using time-based audiovisual data, analysts need to strike an appropriate balance between in-depth analysis and the needs and constraints of a particular project. Analysis time to sequence time ratios (AT:ST) typically range from 5:1 to 100:1, while detailed micro-analyses may approach ratios as large as 1000:1 (Fisher & Sanderson, 1996). In selecting data for analysis, the method described here follows the general outlines given in Trigg, Bødker, & Grønbaek, (1991), Suchman & Trigg (1991), and Bødker (1996). In accord with Engeström's approach to work development and IT-design research (Engeström, 1991; Engeström & Escalante, 1996; Engeström, 2000), these sources recommend the use of interaction breakdowns and involuntary focus shifts as indicators of potentially design-relevant episodes in the data.

In systemic-structural terms, a *breakdown* can be defined as forced changes of subjects' strategies of action caused by their evaluation of an unacceptable divergence between the actual results of actions and the conscious goals of those actions. Typically, breakdowns are characterized by subjects' (temporary or permanent) abandonment of the task in hand, which, depending on their motivation and the importance of the task, may lead them to reformulate the task-goal. In relation to IT-design, breakdowns suggest that a subject's approach to problem solution with the specific computer artifacts in use, in a specific use setting, and with regard to some specific task, is somehow proving inadequate; and that the opportunities for learning-in-use afforded by the artifacts or use setting are currently insufficient to the requirements of the situation.

Breakdowns are normally easily identified from video data, simply by noting subjects' visible behavior and verbal utterances. The term *involuntary focus shift* describes a somewhat less critical situation. This is where some aspect of the computer artifacts in use disrupts the flow of task performance, requiring the user to focus on understanding or manipulating the tool itself rather than the transformation of the work object. Although they can be highly indicative of e.g. interface design problems, such focus shifts are less easily identified than complete breakdowns, requiring closer attention to on-screen actions. Here, coordination of camera and screen-recording data is useful.

ONE  
SessNo: 46.      Coding: ONE

Description	Date:	Time:	Applications?	Discussion
User ID Name(s) (S: DATA MAKING) (E: ENERGY SCHEDULE)	S/2/01	4-6:15 PM	ULEADS VIDEO STUDIO	User's perception of cause NOTED DURING THIS SESSION BUT HAD BEEN ONGOING. LEADS ME TO THINK THAT USERS SEE CLIPS/SOUNDS AS 'THINGS' BUILDING BLOCKS AND EXPECT THEM TO HAVE AN IDENTITY THAT IS NOT AFFECTED BY
What activity? VIC TREE? VOICE TAB? MORE TIMELINE VIEW?	What interface features? USERS OPERATING ON 2 MACHINES, BOTH RUNNING UVS. ATTEMPTING TO SYNCHRONISE 'BY EYE' AND IN CONSULTATION	What problem?	What action? SYNCHRONISING FADING SOUNDTRACK	Researcher's perception of cause: MACHINE-SPECIFIC CONSTRAINTS, THE FEELING IS THAT THEY SHOULD HAVE THE SAME EXISTENCE IN BOTH AUTHORIZING + PLAYBACK AND ON DIFFERENT MACHINES, OUR EVOLUTIONARILY DERIVED SCHEMAS ARE VIOLATED?
What operation? DROPPING SOUND FILES ONTO TIMELINE	User input/action(s): PUTTING VOICE FILES OF DIALOGUE ON TIMELINE OF CLIPS	Outcome: BUT FILES RUNNING AT DIFFERENT RATES ON DIFFERENT MACHINES - ACTIVITY AS VISUALISED NOT SUCCESSFUL - SEEMINGLY SYNCHRONISING FILES BEING AUTHORIZING NOT SO ON PLAYBACK	Researcher comments:	

Breakdowns (profforma)

Audio  Photo  Video  Log  Screen Capture

TRE 2.2  
SEE LOG

Figure 3. Completed Breakdowns Pro-Forma.

In the longitudinal study (described in Section 2.3 above), which took an opportunistic approach to video capture, interaction breakdown was selected as a primary trigger both for initial recording and subsequent analysis. A paper-based instrument, the Breakdowns Pro-Forma (BDPF), was designed to support the identification and preliminary qualitative analysis of interaction breakdown and recovery across all longitudinal study data, including video. The BDPF could be completed concurrently with, or immediately following, observation or recording; during compilation of the field note record; retrospectively during field note review; and during initial review and logging of the video data. Figure 3 shows an example of a BDPF completed during initial video data review. Pro-forma categories included a unique identifier, date, time, task details, the application(s) in use, nature of the problem(s), interface elements involved, whether and how recovery was achieved, user and researcher comments, and codes for the relevant video segment and related data sources. BDPF data was subsequently compiled into a spreadsheet and linked to the database of video and interview transcripts.

In contrast to the longitudinal study, the short field study was specifically designed to capture all interaction in as much detail as possible. In this case the resulting body of video data was logged and transcribed in its entirety prior to further analysis. Following the identification of task and sub-task sequences within the recorded activity, functional analyses were conducted on the video data and sets of transcripts arranged into sequences representing activity by each participant sub-group (see Section 5 below). Incidents of interaction breakdown, involuntary focus shifts, and notably fluent interaction were then used as a basis for selecting data for detailed morphological analysis. Functional analysis was applied to the whole dataset.

## 4. TRANSCRIPTION & DEVELOPMENT OF GENERAL TIME STRUCTURES

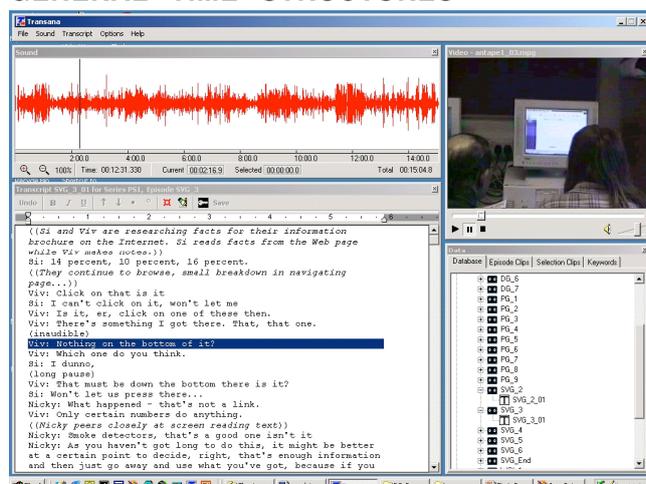


Figure 4. Transana interface, showing Waveform, Transcript, Video and Database windows.

The production of accurate transcriptions of subjects' speech actions during task performance is central to the approach described here. Time-coded transcripts provide a basis for other analytical techniques, in addition to being an important resource in their own right<sup>2</sup>. Transcription is not, however, simply about producing transcripts; engaging in transcription engenders the close familiarity with observed events that is one of the principal advantages of working with video data. It is recommended that, wherever practicable, persons performing activity analyses should also be responsible for transcribing the materials. The alternating focus between audio and video sources, scanning both verbal and motor behavior, encourages awareness of their complex interconnectedness, and of the interlocking individual-psychological and sociocultural aspects of use activity. In the studies discussed here, transcription was carried out using Transana<sup>3</sup>, an open-source video transcription and analysis tool that supports frame-by-frame viewing and the accurate linking and playback of transcripts and video segments (Thorn, 2002, see Figure 4.).

### Compiling Transcript Sequences

For practical reasons such as capacity of storage media, multiple camera sources, etc. video data is usually captured, stored and transcribed in discrete segments, often of varying size. For systemic-structural video analyses such as in the short field study described above, it is useful to compile separate segment transcripts into sequences representing the whole period of activity of interest for the study. Where multiple cameras and other video sources are used, multiple transcript sequences can be used to represent a whole observation session from the varying points of view of any of the sub-groups or individuals in the study. Similarly, in some cases it can be appropriate to use an editing application to compile video segments into longer sequences showing the whole arc of development of a particular task activity.

## 5. ISOLATION & CLASSIFICATION OF TASKS

The notion of activity during task performance as a fundamental object of study underpins systemic-structural video analysis. Human-computer interaction can be considered as a specific type of work process involving sequences of goal-directed tasks (Bedny & Harris, In Press). In SSTA, a task is understood a sequence of goal-directed actions involving an initial situation (the problem presented before task performance begins), a transformational situation (actions taken to solve the problem), and a final situation (initial situation changed). Video analysis can be used to study how the structure of a task changes during different stages of performance, and to identify how many basic transformational stages are required.

Tasks are understood as being organized around a supervening goal, with the vector motive-goal determining the directedness of activity during task performance (Leont'ev, 1978, and see Section 2.1 above). Tasks are always carried out under specific circumstances - task conditions - which determine the constraints on performance. These include interacting situational elements, rules, and alternatives for situation transformation. Tasks are structured by requirements that help to specify the goal, such as instructions or commands, and are also affected by the means of work in the given conditions, and the raw materials or input information being explored or transformed. In some circumstances task requirements and conditions may contradict each other, causing breakdowns in activity. Task attributes include complexity, subjective assessments of difficulty, and significance. These attributes involve elements (such as subjects' past experience) which cannot be directly evinced from video data but must be established through other techniques such as interviews or historical analysis. This emphasizes the need to integrate video analysis with other methods of research. Bødker (1996) and Kaptelinin *et al* (1999) have proposed the use of checklists in support of this integration process, an approach adopted by the field studies reported here.

### 5.1 Isolation of Tasks

Transcript sequences provide a basis for identifying the logical structure and temporal sequence of task and sub-task solution from the ongoing flow of activity. These general mappings underpin task classification and are a precursor to the techniques of morphological analysis described in the following section. In functional analyses, they allow comparison of the structure of actual task performance with objectively set task conditions, e. g. they enable the comparison of observed task-solving strategies with those given in verbal instructions, manuals, etc., and as presented in subjects' verbal protocols. Using line numbers as reference points, sequential transcripts are sub-divided into segments, based on identification of the task or subtask in which participants are engaged.

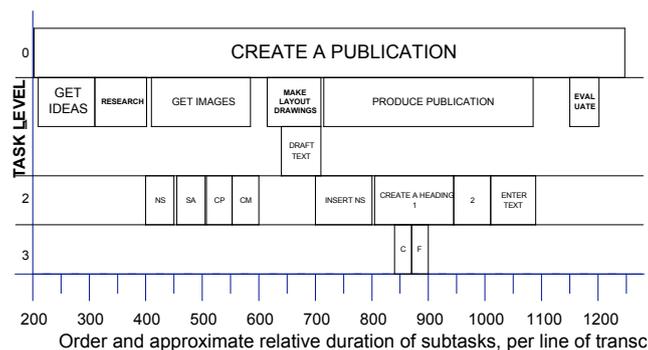


Figure 5. Structure & sequence of tasks in short field study.

The upper limit at which the isolation of tasks begins, and which sequences of activity are considered as primary or sub-tasks, is study-dependent, and will also be practically determined by the nature of the video data. For example, in analyzing video from the short field study, a broad differentiation was made between the initial portion of the 3-hour session, where the activity of the whole group was guided by the goal setting task conditions based on instructions by the tutor; the majority of the session, when sub-groups were involved in task solution; and a final part, when the whole group was involved in giving feedback on their outcomes. Each of these sections was then further sub-divided into sequences of activity organized around distinctive task-goals. This decomposition continued until the level of discrete actions was reached, producing a 4-level hierarchical representation of sequentially ordered tasks and sub-tasks. Figure 5 shows a schematic representation of this task structure arranged against an approximate "timeline" formed by taking counts of line numbers in the transcript sequence. When required, more accurate time-structure diagrams of activity at the task level can be constructed using video time codes or other methods of time measurement as markers.

As noted above, tasks and sub-tasks are primarily differentiated on the basis of their organizing goals. The formation or acceptance of a distinctive task-goal is taken as marking the inception of a task or sub-task. Achievement or abandonment of a goal is taken to mark task completion. Identification of these junctures is achieved through examination of the video and transcript. Goal formation or acceptance is often associated with some orienting activity; in collaborative work this may be observed as discussion, sketching, note-taking and so on, as in the transcript extract below:

- 227 S: So what we gonna do then?  
 228 S: Health and safety sounds alright doesn't it?  
 229 T: Health & safety in the home? Ok, so now you want to jot down all the (inaudible)  
 230 S: (to V) Why has (inaudible) or something like that?  
 231 (T talks to other students while S and V discuss the topic. Mostly inaudible.)  
 232 (V is drawing on a pad on the table)

The next phase of a task is marked by sequences of executive actions organized around the task goal, e.g.:

- 887 ((S scrolls through fonts menu in Publisher))  
 888 T: If you want something a bit fancy there's Victorian, I dunno what that

889 S: ((turns to V)) That's quite nice innit?  
 890 V: Yes  
 891 S: Want that?  
 892 V: Alright  
 893 T: Or you can, you can bold it if you want... and then, if you want a shape... ((points to menu))  
 894 S clicks to pull down shapes menu))  
 895 T: then you can choose whether you want it to go round, or wiggly, or slanting, or any of those  
 896 S: which one do you want V?  
 897 V: Anything you fancy  
 898 V: I just gonna say (something like)  
 899 S: Tha one or tha one?  
 890 ((indicates with cursor))  
 891 V: Now then, try that one is it, yeah

Followed by evaluation:

985 S: Eh, that's nice innit?  
 986 T: Not very readable though, that's the only trouble with that one

Task/Subtask name	Subtask level	Begin	End	Duration
Create a publication	0	201	1248	1047
Get ideas	1	223	291	68
Research	1	293	393	100
Get images	1	410	613	203
Get image of no-smoking sign	2	407	458	51
Get image of smoke alarm	2	463	482	19
Get image of chip pan	2	504	554	50
Get image of child w/matches	2	556	594	38
Make layout drawings	1	624	707	83
Draft text	1	645	707	62
Produce publication	1	712	1098	386
Insert no smoking picture	2	712	802	90
Create a heading 1	2	810	962	152
Create a heading 2	2	962	1001	39
Choose heading color	3	837	865	28
Choose font	3	866	893	27
Enter text	2	1002	1094	92
Evaluate project (V only)	1	1158	1187	29
Collect copies	0	1274	1276	2

Figure 6. Task decomposition from video transcript.

Figure 6 shows the decomposition of activity during the short field study from the point of view of one sub-group, using transcript line numbers as a reference.

## 5.2 Classification of Tasks

Once tasks have been identified it is then possible to classify them according to various criteria. Distinctions between task types are relative rather than absolute, and are based on assessing the varying degrees of freedom of performance associated with individual tasks. Those tasks which are highly structured by the artifacts in use (e.g. data entry) may be considered as predominantly deterministic, or deterministic-algorithmic; that is, they require a standardized sequence of actions for successful completion. Probabilistic-algorithmic tasks are those which involve choices at some stages. Depending on the interaction between task requirement and conditions, such tasks often become problem-solving tasks for the subject. In cases where task uncertainty is even greater, tasks are termed heuristic or semi-heuristic, the major criterion in these categorizations being the extent to which the task presents an undefined field of solution. Probabilistic-algorithmic tasks often also include non-algorithmic problem-solving components; in the same way, semi-heuristic task-problems

may also include algorithmic and semi-algorithmic sub-tasks (Landa, 1983; Bedny & Harris, 2004).

Classification of tasks will clearly be more or less relevant depending on the activity under study and the stage and nature of the IT-design process. Task typing may prove useful during initial project scoping, as it can be significantly more difficult to design work processes that involve a high levels of freedom of performance, creativity, and unpredictability. In requirements gathering, task classification can underpin initial design specifications. In evaluation, applications, prototypes, or designs may be assessed against the types of tasks in which they are expected to be, or are actually, used. For example, in our longitudinal study the video editing application in use employed a sequential metaphor and modal functionality. This design influenced users to attempt a step-by-step approach to task solution. However, most of the video-editing tasks in which the application was employed could be classified as semi-heuristic, with no clear order or definition between many steps. These real-world tasks required greater operational flexibility than was readily available from the application interface. Noting this divergence between design and use at the task level provided a useful way of thinking about the large number of interaction breakdowns experienced with this application (Harris, 2004).

## 6. MORPHOLOGICAL ANALYSIS OF VIDEO DATA

A fundamental principle of activity theory is the unity of cognition and behaviour (Bedny, Karwowski, & Bedny, 2001). The structure of activity during task performance is understood as formed from a logically organized system of external-behavioral and internal-mental tool-mediated actions and operations, a structure which is continually changing in response to internal and external conditions. Activity-theoretical IT-design can be said to be concerned with the effective alignment of the external and internal tools of activity. A key aspect of this alignment effort is the identification of, and attempt to reduce (or design support for), those aspects of task performance which demand the most time or attention, and which involve the highest levels of complexity.

This section discusses the morphological (structural) analysis of video data, using action as a primary analytical unit, and focusing on the interconnectedness of, and transition between, mental and motor actions. It describes the morphological analysis of video data through (1) the isolation and classification of discrete actions, and (2) the generation of algorithmic descriptions of the logical structure of activity. Structural analyses of activity can provide insights into how external tools such as controls, displays, screens, instructions etc. interact with subjects' internal tools such as conceptual models, skills, knowledge etc. (Bedny & Karwowski, 2004a). This approach allows comparisons between the physical and logical configuration of the equipment in use (using descriptions developed from the objectively-logical perspective, see Section 1.2 above)

and the temporal, spatial, and logical organization of subjects' (real or idealized) actions.

### 6.1 Isolation of Actions

Table 1. Example of task isolation & classification.

Action	Goal	Object	Tools	Type
1	Reach & grasp mouse	mouse		Object-practical
2	Locate display pane	Interface	Graphical interface elements	Sign-practical/ Simultaneous - perceptual
3	Move cursor over hyperlink	cursor	mouse	Object-practical
4	Activate link	hyperlink	Cursor, left mouse button	Sign-practical

In order to isolate discrete actions it is necessary to identify the goal, object, and tools involved in *each* action. The nature of an action is dependent on the interrelation of these components in any particular situation. A useful approach to isolating individual actions in a sequence of video-recorded activity is to begin by setting out a basic sequential description of the technical steps involved in the task under study. Figure 7 shows such a description, of a Web authoring task recorded during the longitudinal study. Analysis can then proceed from the sequential description to the isolation of individual goal-oriented actions. For example, Table 1 shows the isolation of actions involved in Step 2c of Figure 7, "in the display pane, move cursor to hyperlink being checked, left-click on link" and illustrates their classification (according to the two typologies outlined in the following section). It can be seen that some actions involve several external tools. Where tools are not defined (as in action 1) this indicates motor activity not involving external instruments. However, as noted above, activity theory always assumes that motor actions contain cognitive components and, as such, may involve the use of internal psychological tools. Such tools can be assumed in action 2, which implicates not only the perception of signs visible on the interface but also their interpretation using concepts and images.

1. Launch or restore focus to Web browser (IE)
  - a. If application already running move cursor onto window area and left click *or* move cursor onto taskbar icon and left click.
  - b. If application not running, move cursor to application icon and left-click *or* move cursor to start button, left-click, navigate to appropriate menu, choose application icon or label and double left-click.
2. Load or refresh appropriate HTML document
  - a. If document is already being displayed, move cursor to refresh icon (or select command from View pull-down menu) and left-click
  - b. If document not displayed, load into browser by either selecting File>Open from pull-down menu, then typing file path or browsing to file location in Open dialogue box *or* use left mouse button and cursor to drag file icon from desktop or other location and drop on browser display pane by releasing mouse button
  - c. In the display pane, move cursor to hyperlink being checked, left-click on link
3. View resulting display and assess
4. Make decision on whether link needs to be removed or altered
5. Launch or restore focus to text editor (Notepad)
  - a. If application already running move cursor onto window area and left click *or* move cursor onto taskbar icon and left click.
  - b. If application not running move cursor to application icon and left-click *or* move to start button, left-click, navigate to appropriate menu, choose application icon or label and double left-click.
6. Load or review appropriate text (.html) file
7. Identify section of markup that corresponds to link being checked
8. Cut, delete, or modify link markup
9. If moving location of link:
  - a. Identify new location in appropriate text file
  - b. Navigate text editor to appropriate file and location
  - c. Insert cursor and paste link markup
10. Return to Step 1.

Figure 7. Sequence of basic technological procedures in the task "Update Web Pages".

It should be emphasized that in systemic-structural theory, the concept of "tool" is tightly associated with the concept of action; outside of a specific task, it is often not possible

Table 2. Fragment from the algorithmic description of a Web authoring task.

Member of algorithm	Description of Members of Algorithm
$7^{11(1)} \downarrow \downarrow O_1^\alpha$	Identify appropriate Web browser window
$1 \downarrow \uparrow l_1$	If correct Web browser window has focus go to $O_{-3}$ ; if not go to $O_{-2}$ .
$O_2^\epsilon$	Bring browser window to foreground by moving cursor onto window area and left-clicking <i>or</i> moving cursor onto taskbar icon and left-clicking.
$1 \downarrow O_3^\alpha$	Check to see if correct Web page is displayed
$2 \downarrow \uparrow l_2$	If appropriate Web page is displayed go to $O_{-4}$ ; if not go to $O_{-5}$
$O_4^\epsilon$	Refresh browser display by moving mouse cursor to refresh icon and left-clicking <i>or</i> selecting command from View pull-down menu and left-clicking
$2 \downarrow O_5^\epsilon$	Open correct HTML document by either selecting File_Open from pull-down menu, and browsing to file location using Open dialogue box <i>or</i> use left mouse button and cursor to drag file icon from desktop or other location and drop on browser display pane by releasing mouse button
$4 \downarrow O_6^\alpha$	Identify next hyperlink to be checked
$O_7^\epsilon$	Activate link by moving mouse cursor to hyperlink being checked and left-clicking
$O_8^{\alpha th}$	Look at new page display in browser window and compare with expected result
$3 \downarrow \uparrow l_3$	If browser display is appropriate go to $O_{-9}$ ; if “The page cannot be displayed” HTML file is displayed <i>or</i> if link works but page inappropriate go to $O_{-11}$

to precisely determine whether material or ideal artifacts are acting as tools mediating a specific action, or as objects of that action. This suggests that personal computers, or software packages, cannot simply be classed as “tools” of activity except when conducting broad macro-analyses. Rather, they should be considered as means of work that present (or create) a variety of material and symbolic objects that, at different stages during the performance of computer-based tasks, may either mediate actions as tools, be the object of actions, or simply provide the conditions under which actions are performed (cf. Bødker, 1991, 1999).

## 6.2 Classification of Actions

A number of different approaches to the classification of actions have been developed in system-structural theory. Two were used in the studies reported here. The first differentiates between types of mental action based on two considerations:

1. the degree to which they require deliberate examination and analysis of the stimulus (their direct connection with, or transformation of, the input);
2. the dominating psychological process during action performance: sensory, simultaneous perceptual, imaginative, mnemonic, etc.

The second classification scheme is more generalized. Actions are categorized according to the nature of their object (either material or symbolic), and according to their method of performance, (either practical or mental). This scheme thus distinguishes between:

- *object-practical actions* performed with material objects;
- *object-mental actions* performed on mental images;
- *sign-practical actions* performed with external signs;
- and *sign-mental actions* performed through the mental manipulation of signs or symbols.

Table 1 provides examples of classification using both schemes. When required, standardized motor actions may also be identified and categorized using established measured time and motion systems such as MTM-1<sup>4</sup>. This approach can be helpful in determining the interrelationship of mental and motor actions (in Bødker’s terms, analyzing the impact of the *physical* aspects of the computer artifacts in use on the structure of activity) and when using time-structure analysis (see Figure 8 and discussion in next section).

### 6.3 Video Data as a basis for Algorithmic Analysis

The identification and classification of actions provides a basis for developing formal descriptions of the logical structure of activity during task performance from the individual-psychological perspective. These descriptions take the form of symbolic representations known as human algorithms. A completed algorithm is presented in tabular form and consists of a specialized notation accompanied by explanatory text. Table 2 above gives an example.

Video analysis using human algorithms involves:

- the subdivision of observed activity during task performance into qualitatively distinct units, called *members* of an algorithm;
- determination of the logic of the organization and sequence of those units in the structure of activity.

There are two main types of members of an algorithm. *Operators* are composite units, formed from 3-5 discrete goal-oriented actions organized by a supervening goal. They denote clusters of efferent (O epsilon) and afferent (O alpha) mental and motor actions that transform objects, energy, and information. Members of an algorithm called *logical conditions* (*l*) determine the logic of selection and realization of different members of an algorithm, and indicate the decision-making processes involved in task performance. Logical conditions may be complex (L) or simple (*l*). Operators and logical conditions are labeled in series with subscript numbers to facilitate reading of the algorithm. The algorithm is read from top to bottom; arrows with superscript numbers designate the logic of non-sequential (forward or backward) transitions from one algorithm member to another.

Developing human algorithms can help identify those aspects of task performance where design intervention might prove most effective: for example, they can highlight instances where tools require users to perform unnecessarily repetitive or demanding actions. In the longitudinal study, algorithms were used to support the identification of how, and at what point, aspects of computer artifact design contributed to interaction breakdown. Table 2 shows a fragment from the algorithmic description of the Web authoring task discussed earlier (Sections 6.1 & 6.2). Here, the subject was using multiple applications (text editor, browser, file manager), each with different functionality and interface features. Examination of the algorithm helped to identify those points in the task where complexity became maximal in terms of perceptual and decision-making actions. In this case, this led to the identification of differences in the ways in which the applications in use handled windowing as a major contributor to interaction breakdown.

The construction of human algorithms supports detailed examination of the performed actions and logical relationships in a given task or sub-task from an individual-psychological perspective. When analyzing collaborative activity, algorithms can be constructed from the point of view of each involved user, then compared,

contrasted, or combined. Having mapped the logical structure of task performance also makes it possible to describe the temporal structure of activity in terms of performed actions in more detail, supplementing the timeline charts used for task analysis as described earlier. Time measurements derived from the video data can be used to specify the duration of individual elements of activity, with particular attention being paid to the structure of sequential and simultaneous performance of mental and motor actions. Figure 8 shows a fragment from the time-structure of an HCI task using a word-processing application developed in a lab-based study by Sengupta & Jeng (2003).

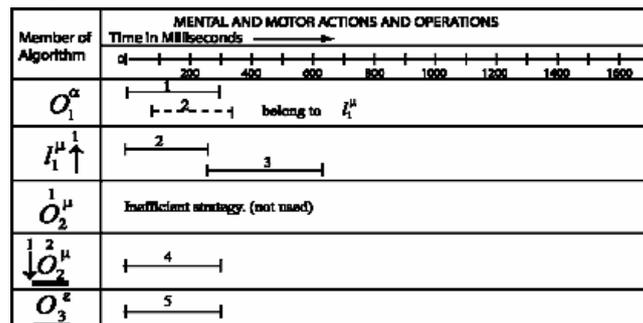


Figure 8. Fragment of time structure of computer graphics task. From Sengupta & Jeng (2003).

Methods of algorithmic and time-structure analysis can also be further supplemented through the use of standardized classifications of motor actions, such as those developed within the MTM-1 system (see Note 4 and Bedny & Meister, 1997 pp. 252-262). While an extended discussion of algorithmic and time-structure analysis is beyond the scope of this paper, further information can be found in Bedny *et al.*, (2000); Bedny, Karwowski, & Kwon, (2001); Bedny & Karwowski, (2003b); Sengupta & Jeng, (2003); and Sengupta, 2004.

### 7. FUNCTIONAL ANALYSIS OF VIDEO DATA

As noted earlier, SSTA understands human activity as a complex, multi-level, psychologically self-regulating system. Functional analyses trace various aspects of the self-regulation process, at different levels of detail and different stages of activity. They produce functional descriptions which, firstly in conjunction with other methods of individual-psychological analysis such as parametrical and morphological analyses, and then in coordination with social-historical and objectively-logical descriptions, can be used to produce a holistic understanding of activity (Bedny & Karwowski, 2004b).

The functional analysis of video data proceeds by using functional models of activity self-regulation as “lenses” with which to scan the activity captured on video tape. Self-regulation models depict components of activity called *function blocks*, and their interrelationships. Function blocks represent coordinated systems of sub-functions with a specific purpose within the structure of activity (Bedny & Karwowski, *ibid.*). They mutually affect each other through feed-forward and feedback influences; while remaining functionally invariant, their specific content changes as

activity unfolds. Self-regulation models may be used for the analysis of observed activity at various levels of detail, according to the purpose of the study and the nature of the data available. Functional analysis of video data is used to study how factors such as users' goals, motivation, and experience affect their interaction with the systems in use.

### 7.1 Using Video Data to Trace the Goal Formation Process

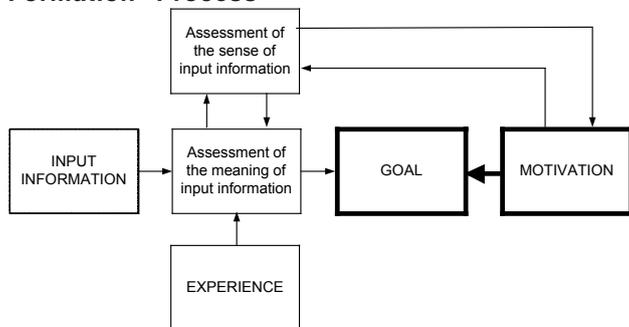


Figure 9. Functional model of goal formation. After Bedny (1997).

In analyzing data from our short field study (see Section 2.3) functional analysis has been used to investigate how the computer artifacts involved supported subjects' goal formation. As a cognitive mechanism, the task- or action-goal can be more or less clear and precise, and either totally or partly conscious. In the type of explorative activity which can constitute a significant part of IT use – especially for less experienced users - subjects' goals are initially vague, gradually becoming more precisely formulated as interaction proceeds. As they perform exploratory actions, subjects produce and analyze a sequence of trials-and-errors, leading to the creation of a hypothesis about the situation, and the formulation of a preliminary goal for activity. Such preliminary goals can be considered as corresponding to particular level of aspiration, inasmuch as they are tied up with evaluations of task difficulty that include elements of self-evaluation. In the research project in which these methods have been used, which is concerned with IT use by individuals from low-income, low-education backgrounds, it has been of fundamental interest to find methods of assessing how the design of computer artifacts in use affects this process of goal-formation, which emerges as a critical factor in the development of fluent technology use by study participants.

The example described here examines the collaborative computer-mediated activity of two subjects, S and V, as they undertake the loosely defined desk-top publishing task. Successful completion of the task was dependent on their formation of a “shared vision” (a sufficiently aligned task-goal) of the form and content of the publication. This required the continual adjustment of their individual goal-images and strategies for action, through dialogue and examination of actual outcomes on the screen of the PC in use. S & V engaged in extensive explorative activity, investigating various functions of the DTP software, and searching for images and information on the Internet. Their

individual, object-oriented, activity was regulated and coordinated through ongoing discussion. Once sufficient precision and alignment of their individual goal-images had been achieved, a task-problem solution adequate to each participant's subjective assessment of the task requirements was achieved.

Figure 9 shows the self-regulation model used for the functional analysis of the participants' activity. This simplified model<sup>4</sup> includes (macro) function blocks representing goal, motivation, experience, and the assessment of sense and meaning of input information in the activity under analysis. Analysis proceeded through systematic examination of the video data, transcripts, and other evidence in order to trace the changing contents of the different function blocks in relation to each individual. In the example here, input information was generated by the graphical display presented by the (shared) computer monitor, verbal instructions from the tutor, and the collaborating subjects' dialogue; these sources were studied using the video footage and transcripts. Data on long-term aspects of the subjects' experience and motivation (individual-psychological profiles, social background, previous observed behavior, etc.) were derived from sources such as interviews and attendance and achievement records. Assessments of how subjects' experience, motivation and interpretation of the sense and meaning of input information developed during task performance came from examining the record of motor and verbal actions.

### 7.2 The Time Structure Outline

Line Nos.	Description
291-292	V signals end of 'get ideas' subtask, T confirms.
293-393	Sub-task 'research'. PC becomes major means of work, Web browser, Search Engine (text search) main material tools mediating activity. S becomes main tool operative. List artifact created by V also supports activity.
297-300	S encounters some physical/behavioral difficulty in assuming the role as PC operator due to physical environmental (space, configuration) and individual (ambivalent handedness) constraints.
367-379	Breakdown in interaction with Web Browser as S & V wish to use text item on Web page perceived as hyperlink that does not respond to left mouseclick. This slows hitherto fluent interaction with site, as users become unsure how to read the semantic properties of the page. Goal-directed action becomes more exploratory, as users pursue what the page can give them rather than what they purposively seek—page becomes object of actions. On return to group, T confirms nohyperlink status of (underlined?) text and activity moves forward again.
382-392	T negotiates cessation of sub-task 'research', although S & V have not indicated that they feel the goal has been achieved—the subtask goal is too imprecise to clearly guide activity. The tutor, T, is regulating the activity from the point of view of the task goal 'project to be completed in one 2 hour session'.
393	V confirms willingness to end subtask 'research'

Figure 10. Fragment of a time-structure outline.

A tabular form based on the transcript sequences was developed in order to support of the process of functional video-analysis. The Time Structure Outline (TSO) correlates durations (indicated by transcript line ranges) with brief descriptions of task-related events depicted by the video and transcript data. A fragment of one TSO is shown in Figure 10. The TSO provides a cross-cutting view into the dataset, facilitating the systematic interrogation of the data from the point of view of each of the function blocks in the self-regulation model in use. As can be seen in Figure 10, the table can also be graphically annotated as an aid to interpretation, using e.g. shading and simple symbol systems. In the example here annotations indicate incidences of breakdown, and mark passages of interest with regard to goals, tool use, and usability issues. Shading indicates the start and end of subtasks, facilitating

the integration of functional analysis with the methods described in previous sections.



Figure 11. Collaborative production of a DTP document - frame from camera video-recording.

During the functional analysis of S and V's task-solving activity, the TSO helped to identify and describe various stages of the goal-formation process as evidenced in the data. For example, the TSO entry below:

S repeatedly passes her hands across the screen as she speaks, indicating (to herself & V) where the block elements of the page layouts should go. The image-goal is now precise enough for her to implement it directly; at 704, 705, 706, S & V exchange utterances that indicate that they are now both visualizing a similar image & text.

Refers to the following transcript extract:

691 S: Erm, that smoking one might fit if you have it up on  
 692 like on trains and buses, so what would be good for that  
 693 one is erm, smoking, smoking over there.  
 694 V: See if I can get this 26 percent  
 695 S: Ah. Ah you could put that right... 26 percent ...of  
 696 people in their house, right, cause this is the house  
 697 one innit - that's right - then you could put erm  
 698 ((Gestures across screen without speaking to  
 699 indicate layout and text))  
 700 ((indicator of goal-image precision))  
 701 S: 26 percent of people in the house..and  
 702 V: How about if you put "smoking kills 26 percent"  
 703 S: In houses  
 704 V: Smoking in bed (S: ah yes) when a cigarette..  
 705 S: Falls out the ashtray  
 706 V: and it goes down the side of a chair  
 707 S: Go on then.

The relevant frames of camera video (Figure 11 shows an example) show the collaborators leaning in toward the screen, and S using gestures to indicate the position of text and image elements in the document being created. Subsequent video segments and screen recordings show the process of attempting to place screen elements in the indicated positions. Using these, transcripts, and a copy of the artifact produced by the activity, it becomes possible to identify which aspects of the applications in use supported or hindered the more precise formulation of the subjects' task-goal.

## 8. CONCLUSION & CLOSING ACADEMIC REFLECTION

This paper has described systemic-structural activity analysis using HCI video data, and suggested ways in which this approach can be used to inform IT-design. SSTA emphasizes that human activity is complex and multi-dimensional, and should therefore be analyzed using multiple methods and from a variety of mutually supporting perspectives. The methods described here focus on the individual-psychological aspects of computer-mediated activity; their primary object of study is each participant's cognition, motivation, and behavior during task performance. The outcomes are structural and functional descriptions of individuals' recorded activity, in a variety of forms and of varying degrees of detail and abstraction. When studying collaborative task solution, models of individual activity can be compared and correlated with each other on the basis of the time-structure of task performance. Using these methods, HCI video data analysis can support identification and evaluation of those aspects of the (actual or proposed) use situation - computer artifacts, task conditions, and user skills and experience - where (re)design intervention is required, possible, and may prove effective in supporting improved task performance.

We have suggested that the individual-psychological perspective is the one most likely to produce the kind of detailed information needed to support design projects "close to technology" (Bertelsen & Bødker 2003 p. 323) such as interface design. However, the structure of a person's work activity, and hence the design of means to support that work activity, is clearly not only affected by their own individual characteristics but also by their physical and technical environment, and the norms, standards, rules of behavior etc. that form the sociocultural context of their work. Therefore, individual-psychological activity analyses will almost always need to be used in conjunction with descriptions of observed activity in objectively-logical terms (identifying the basic technological procedures involved in a task, the necessary input information, results produced, etc.) and social, cultural and historical descriptions of the communities and work practices under study. The systemic-structural theory of activity provides a conceptual framework within which these differing perspectives can be effectively integrated.

### 8.1 The Problem of Objects of Study and Units of Analysis

In recent years progress in developing practical IT-design methods based on activity theory has been disappointingly slow. Reflection on the current status of the field in the light of the approach discussed in this paper suggests that at least part of the problem may be due to a preponderance of studies that feature macro-level, structural activity analyses. Much of the currently available English-language activity-theoretical IT-design literature identifies as a basic object of study the activity (e.g. Kuutti, 1996); an activity system (e.g. Bellamy, 1996; Engeström & Escalante, 1996); or an activity network (Korpela *et al*, 2000, Mursu, this volume). Such studies must necessarily emphasize sociocultural and objectively -logical perspectives on

activity study; correspondingly, they will often also tend to neglect to analyze activity from an individual-psychological perspective. In many of these cases, the basic object of study - e.g. activity - is also identified as the “basic, irreducible unit of analysis” (e.g. Bertelsen & Bødker, 2000, p.6).

But what are the implications - in practical terms - of using activity as a unit of analysis when what is being analyzed *is* activity itself? Firstly, problems can arise when analysts seek to delimit the scope of the context to be considered. The question as to where one system of activity, or an activity network, begins or ends can be difficult to determine with any precision, a difficulty that increases as the temporal and spatial scale of the study widens. Secondly, findings from such studies will tend to be mainly in the form of descriptions and historical narratives of the tools, practices, and communities under study. While - as has been noted - such descriptions are clearly both useful and necessary, they may not be sufficiently specific to guide the design process in any other than the most general terms, while also proving resistant to any further formalization or quantification if this is required.

A third difficulty arises from the way in which analysis at the level of activity is often associated with the use of over-simplified models of the structure of activity. This has encouraged imprecision in the use of activity-theoretical terms and concepts in the IT-design literature, one consequence of which has been the production of inconsistent, and sometimes self-contradictory findings. Consider the well-known three-level schema of the structure of activity as activity-action-operation (e.g. as used in Kuutti 1996). Used in isolation, this schema provides little support for detailed analysis of the “middle ground” of activity that falls between the ongoing flow of activity and the mainly unconscious mental and motor acts (operations) that make up its performance from moment-to-moment. The middle term (action) must “cover” a very wide range of temporal durations, hierarchical goals, and multiple levels of motivation, presenting obvious difficulties for checking, replicating, comparing, or applying the findings of studies which will tend to interpret and apply the concept of action in different ways.

Similarly, attempts to use simplified models of collective action as an activity system - such as the well-known triadic schema developed by Engeström (Engeström 1987 etc.) - often obscure some of the most essential, and potentially useful concepts of activity theory, such as the clear distinctions between the goals, objects, and actual results of object- and subject-oriented actions which are fundamental to understanding the self-regulation and coordination of activity.

## 8.2 Task and Action

In attempting to make a contribution toward addressing some of the problems outlined above, this brief academic reflection has three principal aims. The first is simply to suggest that the further development of practical IT-design methods may be usefully informed by drawing on those traditions and schools of thought, both within activity

theory and the wider “activity approach”<sup>1</sup> in general, which differ in their emphasis from the “cultural-historical activity theory” that has mostly informed developments in the field so far. The pragmatic basis for this suggestion is that some of these approaches, such as SSTA, have already accrued extensive experience and knowledge of applied design in domains such as work psychology, ergonomics, training etc. Furthermore, such a move might help to redress the current imbalance between the three perspectives of activity theory (outlined in Section 1.2 above) evident in much of the current work in activity-theoretical IT-design. The focus in this paper has been on what can be learned from the modern systemic-structural synthesis within activity theory; other researchers (e.g. Fjeld *et al.*, 2002, Lauche this volume, Törpel, this volume) have also begun to draw on other activity traditions such as action-regulation theory and critical psychology.

A second aim is to reinstate the notion of the *task* as an important object of study. As has been noted, systemic-structural activity analysis from the individual-psychological perspective specifically focuses on the structure of activity during task performance, where a task is defined as *a sequence of cognitive and behavioral actions and operations organized around a supervening goal* (see also Section 5). The practical usefulness of using the task as an object of study is the way in which it a) delimits the scope of activity analysis and b) (re)focuses attention on the hierarchical and differentiated nature of goals and motivations in activity. As was suggested in Section 5.2, the classification of tasks from the point of view of the way in which their objective qualities structure the space of a subject’s possible strategies of action can offer useful support for the IT-design process.

There can, of course, be problems with IT-design practice based on task analysis. There have been many critiques, from various epistemological viewpoints, of those task-analysis methods - such as the GOMS family (John, 2003) - developed within the framework of human information-processing psychology, typically on the grounds that such “additive models” embody simplistic and mechanistic assumptions which tend to ignore the context within which interaction is embedded (e.g. Ehn & Kyng 1984; Draper, 1993; Bertelsen 1994). This has led some IT researchers attempting to use activity theory to reject the notion of task as unhelpful (e.g. Holland & Reeves, 1996; Nardi, 1996b). Yet, within the tradition of applied AT, concepts of the goal-oriented task, task-solving processes, and task analysis have always been central (Bedny & Meister, 1997, pp. 18-25). Without some conceptual differentiation between the hierarchically nested goals and sub-goals present in work activity, and their correspondingly differing levels of motivation, detailed and consistent analysis can be extremely difficult.

Using the task as an object of study suggests a third aim: clarifying what is meant by *action* as a basic unit of activity analysis. The activity-theoretical concept of goal-oriented action bridges the cognitive, behavioral, and motivational-affective domains, making it possible to overcome the problems associated with the reduction of the

task to an information-processing concept. Yet the clear and consistent design-oriented analysis of activity in terms of a system of spatially, temporally and logically related mental and motor actions requires a unified and standardized understanding of action currently lacking in activity-theoretical IT-design. The definition of action given in Section 2.1 of this paper describes a considerably smaller (i.e. of shorter temporal duration), and more precise unit than that generally encountered in the current literature. Furthermore, SSTA (re)emphasizes that the identification and typing of discrete actions always also entails identification of the goal, tools and object of *each* action. This requirement necessitates, and enables, the detailed description of activity as it unfolds from moment-to-moment within the context of some particular task.

## ACKNOWLEDGMENTS

Many thanks to staff and students at Coleg Morgannwg Open Learning Centre, Pontypridd, Wales; to G. Z. Bedny, N. Avouris, and B. Törpel for helpful comments on earlier drafts of this papers; and to T. Sengupta & O-J. Jeng for permission to reproduce the illustration used as Figure 10.

## NOTES

1. The term "activity approach" is used here to indicate a movement within the human sciences whose origins were in post-revolutionary Russia during the early part of the 20<sup>th</sup> Century; it has subsequently been developed throughout the former Soviet sphere of influence and beyond. The activity approach encompasses such interconnected and overlapping traditions as Cultural-Historical Psychology, Bakhtinian semiotics, Sociocultural Theory, the various schools within Activity Theory (AT), German Action-Regulation Theory, and German Critical Psychology (CP).
2. In common with other AT approaches SSTA grounds dialogic analysis of intersubjective interactions in the work of Bakhtin. See Bedny and Karwowski (2004a p. 139).
3. Transana was developed by the Wisconsin Center for Education Research and is available for free download from <http://www.transana.org>.
4. MTM (Methods, Time, Measurement) 1 is the most basic form of a family of systematic work study methods. It is a procedure which analyses manual operations and methods into basic motions which are assigned a predetermined time standard.
5. Examples of more detailed models of self-regulation are presented in Bedny & Meister (1997 p. 77), and Bedny & Karwowski (2003, 2004b).

## REFERENCES

Anokhin, P. K. (1969). Cybernetics and the Integrative Activity of the Brain. In M. Cole & I. Maltzman (Eds.), *A Handbook of Contemporary Soviet Psychology* (pp. 830-856). New York: Basic Books.

Anokhin, P. K. (1962). *The Theory of Functional Systems as a Prerequisite for the Construction of Physiological Cybernetics*. Moscow: Academy of Science of the USSR.

Bauersfeld, K., & Halgren, S. (1996). "You've got three Days!" Case Studies in Field techniques for the Time-Challenged. In D. Wixon & J. Ramey (Eds.), *Field Methods Casebook for Software Design*. New York: John Wiley & Sons.

Bedny, G. Z. (2004). Preface, special issue on activity theory. *Theoretical Issues in Ergonomics Science*, 5(4), 249-254.

Bedny, G. Z. (1998). Activity Theory. In W. Karwowski (Ed.), *International Encyclopedia of Ergonomics and Human Factors*. New York: Taylor & Francis.

Bedny, G. Z., & Harris, S. R. (In Press). The Systemic-Structural Theory of Activity: Applications to the Study of Human Work. To appear in *Mind, Culture and Activity*, 12(2), 2005.

Bedny, G. Z., & Karwowski, W. (2004a). Activity theory as a basis for the study of work. *Ergonomics*, 47(2), 134-153.

Bedny, G. Z., & Karwowski, W. (2004b). A functional model of human orienting activity. *Theoretical Issues in Ergonomics Science*, 5(4), 255-274.

Bedny, G. Z., & Karwowski, W. (2003a). Functional Analysis of Orienting Activity and Study of Human Performance. In *Proc. Ergonomics in the Digital Age, IEA Congress 2003* (Vol. 6, pp. 443-446). Seoul: The Ergonomics Society of Korea.

Bedny, G. Z., & Karwowski, W. (2003b). A Systemic-Structural Activity Approach to the Design of Human-Computer Interaction Tasks. *International Journal of Human-Computer Interaction*, 16(2), 235-260.

Bedny, G. Z., Karwowski, W., & Bedny, M. (2001). The Principle of Unity of Cognition and Behavior: Implications of Activity Theory for the Study of Human Work. *International Journal of Cognitive Ergonomics*, 5(4), 401-420.

Bedny, G. Z., Karwowski, W., & Kwon, Y.-G. (2001). A Methodology for Systemic-Structural Analysis and Design of Manual-Based Manufacturing Operations. *Human Factors and Ergonomics in Manufacturing*, 11(3), 233-253.

Bedny, G. Z., Seglin, M. H., & Meister, D. (2000). Activity theory: history, research and application. *Theoretical Issues in Ergonomics Science*, 1(2), 168-206.

Bedny, G. Z., & Meister, D. (1997). *The Russian Theory of Activity: Current Applications to Design and Learning*. Mahwah, NJ: Lawrence Erlbaum Associates.

Bellamy, R. K. E. (1996). Designing Educational Technology: Computer-Mediated Change. In B. M. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction* (pp. 123-146). Cambridge, MA: MIT Press.

Bernshtein, N. A. (1967). *The Coordination and Regulation of Movements*. Oxford: Pergamon Press.

Bernshtein, N. A. (1969). Methods for Developing Physiology as Related to the Problems of Cybernetics. In M. Cole & I. Maltzman (Eds.), *A Handbook of Contemporary Soviet Psychology* (pp. 441-451). New York: Basic Books.

Bertelsen, O. W. (1994). Fitts' law as a design artifact: A paradigm case of theory in software design. In B. Blumenthal & I. Gornostaeve & C. Unger (Eds.), *Human Computer Interaction. 4th International Conference, EWHCI '94 St.Petersburg, Russia, August 1994. Selected Papers*. (Vol. 1, pp. 11-18). Berlin: Springer Verlag.

Bertelsen, O. W., & Bødker, S. (2003). Activity Theory. In J. M. Carroll (Ed.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science* (pp. 291-234). London: Morgan Kaufmann.

Bertelsen, O. W., & Bødker, S. (2000). Introduction: Information Technology in Human Activity. *Scandinavian Journal of Information Systems*, 12(1), 3-14.

Blomberg, J. (1993). Ethnographic Field Methods and Their Relation to Design. In D. Schuler & A. Namioka (Eds.), *Participatory Design: Principles and Practices* (pp. 123-157). Hillsdale: Lawrence Erlbaum Associates.

- Bødker, S. (1999). *Computer Applications as mediators of design and use - a developmental perspective*. Unpublished Doctoral Dissertation, Aarhus University, Aarhus.
- Bødker, S. (1996). Applying Activity Theory to Video Analysis: How to make Sense of Video Data in Human-Computer Interaction. In B. M. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction* (pp. 147-174). Cambridge, MA: MIT Press.
- Bødker, S. (1991). *Through the Interface: A Human Activity Approach to User Interface Design*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Card, S., Moran, T., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Erlbaum.
- Cole, M. (1996). *Cultural Psychology*. Cambridge, Ma.: Harvard University Press.
- Diaper, D., & Stanton, N. A. (Eds.). (2003). *The Handbook of Task Analysis for Human-Computer Interaction*. London: Lawrence Erlbaum Associates.
- Draper, S. (1993). The notion of task. In S. Ashlund & K. Mullet & A. Henderson & E. Hollnagel & T. White (Eds.), *Bridges between Worlds, INTERCHI'93 Conference Proceedings* (pp. 207-208). Reading, MA.: Addison-Wesley.
- Ehn, P., & Kyng, M. (1984). A tool perspective on the design of interactive computer support for skilled workers. In M. Sääksjärvi (Ed.), *Proceedings from the Seventh Scandinavian Research Seminar on Systemeering*. (pp. 211-242). Helsinki: Helsinki Business School.
- Engeström, Y. (2000). Activity Theory as a Framework for Analyzing and Redesigning Work. *Ergonomics*, 43(7), 960-974.
- Engeström, Y. (1991). *Developmental Work Research: Reconstructing expertise through expansive learning*. Amsterdam: Elsevier.
- Engeström, Y. (1987). *Learning by Expanding*. Helsinki: Orienta-Konsultit.
- Engeström, Y., & Escalante, V. (1996). Mundane Tool or Object of Affection? The Rise and fall of the Postal Buddy. In B. M. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human Computer Interaction* (pp. 325-373). Cambridge, Mass: MIT Press.
- Fisher, C., & Sanderson, P. M. (1996). Exploratory sequential data analysis: exploring continuous observational data. *interactions*, 3(2), 25-34.
- Fjeld, M., Lauche, K., Bichsel, M., Voorhorst, F., Krueger, H., & Rauterburg, M. (2002). Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware. *Computer Supported Cooperative Work*, 11(1-2), 153-180.
- Harris, S. R. (2004). *Design, Development, and Fluency: An Activity-Theoretical Approach to Human-Computer Interaction Research*. Unpublished PhD. Thesis, University of Glamorgan, Wales, UK.
- Harris, S. R. (2003). *Motivation, Participation and Co-location: Key aspects of Technology-based learning in Adult Basic Education*. (CASE Technical Report CTR0041). Wales: University of Glamorgan. Available for download from <http://case.glam.ac.uk/CASE/StaffPages/SteveHarris/SRH.html>
- Harris, S. R. (2002). PD in Ponty: Design-by-Doing in Adult Basic Education. In T. Binder & J. Gregory & I. Wagner (Eds.), *PDC 02 Proceedings of the Participatory Design Conference* (pp. 278-283). Malmö, Sweden: CPSR.
- Harris, S. R., & Shelswell, N. (2001). Building bridges across the digital divide: supporting the development of technological fluency in Adult Basic Education learners. In A. Mason (Ed.), *Proceedings of the FACE Annual Conference 2001* (pp. 42-51). Southampton: FACE.
- Holland, D., & Reeves, J. R. (1996). Activity Theory and the View from Somewhere: Team Perspectives on the Intellectual Work of Programming. In B. M. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction* (pp. 257-281). Cambridge, Ma.: MIT Press.
- John, B. E. (2003). Information Processing and Skilled Behaviour. In J. M. Carroll (Ed.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science* (pp. 55-101). San Francisco: Morgan Kaufmann.
- Jordan, B., & Henderson, A. (1995). Interaction Analysis: Foundations and Practice. *The Journal of the Learning Sciences*, 4(1), 39-103.
- Kaptelinin, V., Nardi, B. M., & Macaulay, C. (1999). The Activity Checklist: A Tool for Representing the "Space" of Context. *interactions*, 6(4), 27-39.
- Korpela, M., Soriyan, H. A., & Olufokunbi, K. C. (2000). Activity Analysis as a Method for Information Systems Development. *Scandinavian Journal of Information Systems*, 12(1-2), 191-210.
- Kuutti, K. (1996). Activity Theory as a potential framework for human-computer interaction research. In B. M. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human Computer Interaction* (pp. 17-44). Cambridge, Mass: MIT Press.
- Landa, L. M. (1983). Descriptive and Prescriptive Theory of Learning and Instruction. In C. M. Reigeluth (Ed.), *Instructional Design, Theories, and Models: An Overview of their Current State* (pp. 55-74). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Leont'ev, A. N. (1977). Activity and Consciousness (R. Daglish, Trans.), *Philosophy in the USSR: Problems of Dialectical Materialism*. Moscow: Progress.
- Leont'ev, A. N. (1978). *Activity, Consciousness and Personality* (M. J. Hall, Trans.). Englewood Cliffs, N. J.: Prentice-Hall.
- Leont'ev, A. N. (1981). *Problems of the Development of the Mind*. Moscow: Progress.
- Nardi, B. M. (1996a). Activity Theory and Human-Computer Interaction. In B. M. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction* (pp. 7-16). Cambridge, Ma.: MIT Press.
- Nardi, B. M. (1996b). Some Reflections on the Application of Activity Theory. In B. M. Nardi (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction* (pp. 235-246). Cambridge, Ma.: MIT Press.
- Nielsen, J. (1994). *Usability Engineering*. London: Academic Press.
- Norman, D. A., & Draper, S. (Eds.). (1986). *User Centered System Design : New Perspectives on Human-Computer Interaction*. Mahwah, NJ: Lawrence Erlbaum Associates.

- Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-Computer Interaction*. Wokingham, UK: Addison-Wesley.
- Rubinshtein, S. L. (1935). *Foundations of Psychology*. Moscow: Education Press.
- Rubinshtein, S. L. (1973). *Problems of General Psychology*. Moscow: Academic Science.
- Sengupta, T. (2004). *Investigating Graphical User Interface Usability on Task Sequence and Display Structure Dependencies*. Unpublished Doctoral Thesis, New Jersey Institute of Technology.
- Sengupta, T., & Jeng, O.-J. (2003). Activity Based Analysis for a Drawing Task, *In Proc. Ergonomics in the Digital Age, IEA Congress 2003* (Vol. 6, pp. 455-458). Seoul: The Ergonomics Society of Korea.
- Shelswell, N. (2004). *A Critical Investigation into Uses of New Technologies in the Teaching and Learning of Adult Literacy*. Unpublished MPhil Dissertation, University of Wales Institute, Cardiff, Cardiff.
- Shneiderman, B. (1998). *Designing the User Interface* (3rd Edition ed.). Harlow: Addison-Wesley.
- Suchman, L. A., & Trigg, R. (1991). Understanding Practice: Video as a Medium for Reflection and Design. In J. Greenbaum & M. Kyng (Eds.), *Design at Work: Cooperative Design of Computer Systems*. New Jersey: Lawrence Erlbaum Associates.
- Thorn, C. (2002). *Creating New histories of Learning for Math and Science Instruction: Using NVivo and Transana to manage and study large multimedia datasets*. Paper presented at the Conference on Strategies in Qualitative Research, Institute of Education, University of London.
- Trigg, R., Bødker, S., & Grønbaek, K. (1991). Open-ended interaction in cooperative prototyping: A video-based analysis. *Scandinavian Journal of Information Systems*, 3, 63-86.
- Vygotsky, L. S. (1978). *Mind in Society*. Cambridge, Ma.: Harvard University Press.
- Vygotsky, L. S. (1986). *Thought and Language* (A. Kozulin, Trans. Revised Ed.). Cambridge, Ma.: The MIT Press.

# Shared Object – Asymmetrical and Conflicting Objectives

**Stefan Holmlid**  
Santa Anna IT Research Institute  
Sweden  
holmlid@nada.kth.se

**Henrik Artman**  
IPLab, NADA, KTH  
Sweden  
artman@nada.kth.se

## ABSTRACT

In this paper, we describe, motivate and discuss a method that helps actors or stakeholders to identify objectives behind a shared object. In turn, this allows for identifying asymmetries and conflicts that may hamper the result of the process, or cause unnecessary friction during the process itself. When identified the asymmetries or conflicts can be bridged, understood or dealt with in order to prevent hidden conflicts to have effect on the outcome.

## Author Keywords

Activity theory, conflict, asymmetry, procurement, usability, objectives

## ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *theory and methods, user-centered design*.  
K.6.1 [Management of Computing and Information Systems]: Project and People Management – *strategic information systems planning, systems analysis and design, systems development*.

## BACKGROUND

Differences and asymmetries in goals and competencies influence the relationships between different actors in a system development project. Activity theory provides a framework for analysing and uncovering conflicts between actors and activity systems.

In the project *Procurement Competence* we needed to describe and analyse asymmetries and conflicts when two activity systems meet. It is very common to utilize the framework to analyse conflicts within an activity system. It is fairly common to utilize the framework for making these analyses of conflict between, e.g., the tools of two activity systems (Artman, 2002; Borgström, Artman & Holmlid, 2001).

It is uncommon to see an analysis based on conflicts between the objectives of two activity systems. We suggest that such an analysis is fruitful during the establishment of a system development project when defining the relationship between its different actors and stakeholders. It may also be helpful in order to understand and overcome conflicts during a system development project.

Asymmetric distribution of power and asymmetric distribution of knowledge, are common states of the business in systems development. Cooperative design methods and techniques may support the equalization of such asymmetries (Vimarlund & Timpka, 1998). The approach suggested here equips cooperative design with a technique to elicit and make evident at the outset of a project the potential conflicts that arise from differences in objectives between actors.

## INTRODUCTORY MATTER

Activity theory describes a framework for activity systems (see fig 1). Activity theory has a long history within user centred design.

Before establishing a system development project, there has been a chain of analysis within current work and business practice by the customer/procurer. Most of the time, the analysis consists of a period of implicit or informal analysis, which leads to a decision to make an explicit or formal analysis of needs and ideas. The total analysis period is often performed over an extended period of time before the decision to set up a project as a means of meeting goals is made. Moreover, the analysis is situated within the activity system of the procurer/customer (Artman, 2002;

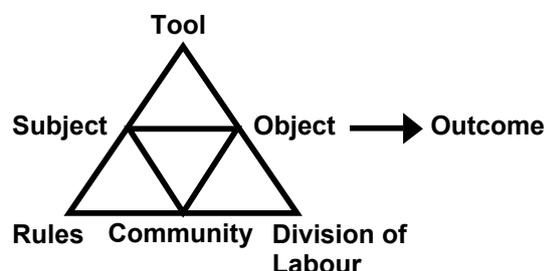


Figure 1. A single activity system.

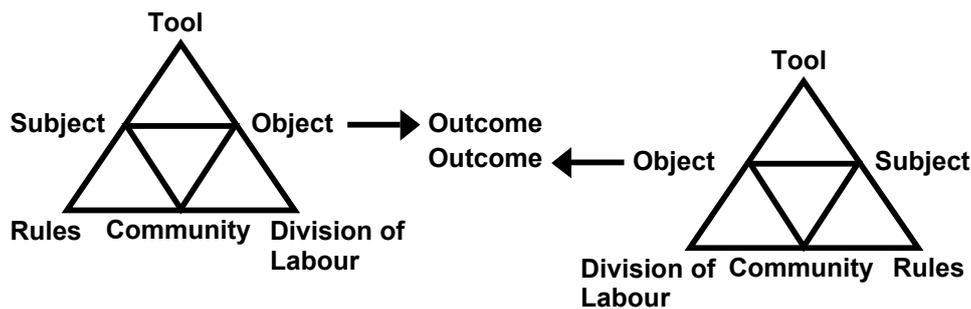


Figure 2. Two activity systems.

Borgström, Artman & Holmlid, 2001; Markensten, 2003; Holmlid & Artman 2003, Holmlid, 2004).

Similarly, a system development practice has its own work and business practice, which has its own activity system. This activity system interacts and changes with other activity systems mostly on a project basis, and undergoes changes in parallel with these projects (Hjalmarsson 2004).

When establishing the system development project, the two activity systems meet (see fig 2). In cooperative design approaches, one would traditionally say that the system developer activity system should aim at producing a system while emancipating the user within the procurer's activity system, and that the developer's role is to mediate between users and management (Greenbaum, 1993; Grudin, 1991; Näslund, 1997; Gärtner & Wagner, 1996). In UCD approaches one would say that the system developer activity system should aim at developing a usable system, according to established process and product standards within the UCD community (Gulliksen & Göransson, 2003; Löwgren & Stolterman 1997).

Activity theory suggests two strategies for analysing activity systems for system development. Either one analyses a single activity system describing the development project, or one analyses a set of different activity systems that together form a larger activity system context for the development project. In the former case, one will encounter problems stemming from the process and result of fusion of systems. In the latter case, one will encounter problems with the selection of which systems to include, or to decide the basis for delimiting a system. Either or, the set of assumptions will potentially have substantial effect on the understanding of the activity system/s and what is possible to perform within these as well as the methods to perform. That is; the construction of activity systems is socio-culturally situated in a reflective and critical community of practice.

#### Short case description and analysis

Taking the idea of multiple activity systems as a point of departure, we analysed the conflicts between and within two activity systems (Artman, 2002; Borgström, Artman &

Holmlid, 2001). In the study the object, system development, was shared between procurer and developer. One of the conflicts between the activity systems was between tools, where the procurer saw user orientation, and the developer saw customer satisfaction. One interpretation of this conflict is that it was rooted in dissimilar objectives, while the object was shared. The procurer's objective was usability, while the developer's objective was lasting relationship. One of the conclusions from the study was that there were conflicts in objectives for the different activity systems involved. Therefore, when looking at the execution of a system development project as a single activity system the conflict in objectives is either assumed or inherited. Analysing conflicts when establishing relationships, and not just assuming the surface conflicts, or inheriting the intrinsic and complex conflicts, could potentially lead to better projects.

We identified a need to structure the objectives and conflicts of two activity systems that meet. It lends its systematic framework from activity theory and design methods (Jones, 1992; Cross, 2000).

#### AIM AND PURPOSE

The aim is to provide a systematic method to describe and structure differences and asymmetries behind shared objects (or aims) across several actors.

#### Initial comments

The aim is to provide a systematic method to describe and structure differences and asymmetries behind shared objects (or aims) across several actors in a system development project. The method is best suited to be used as a means for establishing a sound relationship between two or more different actors. Our experience comes from the asymmetries between procurers of system development and system development companies.

It can be utilized as a basis for dealing with conflicts and asymmetries early in projects, by clarifying them and discussing them, by providing training where such could bridge gaps, or by providing grounding for selecting between potential partners depending on their fit. It should be pointed out that conflicts and asymmetries is not bad in

themselves, but there might be unwanted effects of asymmetries, and assumptions behind unanticipated conflicts.

The basis for using the method is that two activity systems meet and the object (or aim) for the subjects are shared or similar. If the activity systems will stay divided or if they will merge is of subordinate importance. The analysis needs to be performed before such a merge, to uncover inherited conflicts and asymmetries.

Parts of the analysis can be performed by the possibly two organizations separately, but to be useful at least some cooperative work with participants from the different actors is needed.

## OUTLINE OF THE METHOD

In cooperation between actors

1. State the shared object

Individually

2. Clarify objectives based on that they are means to achieve the shared object
  - a. Prepare a list of objectives.
  - b. Order the list in higher-level and lower-level objectives
  - c. Compose an objectives tree
  - d. Add objectives and links between objectives while composing the tree
3. Work through the objectives tree by asking some questions
  - a. Ask the question “How do you achieve/perform <object>” (referred to as <a> in the following)
  - b. Ask the question “How do <a> contribute to <object/ive>”
  - c. Ask the question “How <a> is achieved through <object/ive>”
  - d. If necessary ask the question “Why do you <shared object>”
4. Construct an activity system based on the identified objectives
  - a. Start with the shared object
  - b. Identify objectives that act as *tools* to mediate between subject and object, etc.
  - c. Identify activity system aspects and reformulate them as objectives
  - d. Draw the tree links between objectives
5. Redraw the objectives tree based on 4c

In cooperation between actors

6. Construct an ensemble objectives tree
  - a. Put trees side by side
  - b. Mark all objectives occurring in both especially
  - c. Draw cross links between objectives in different base trees
  - d. Identify conflicting objectives and means
7. OPTIONAL: Construct an ensemble activity system
  - a. Put systems side by side
  - b. Mark all identified cross links
  - c. Mark all identified conflicts

## EXAMPLE

### Web-based questionnaire application

The procurement organisation works within a branch of organisational change and they have developed a product that analyse employees attitudes and situation within the organisation. The user answers several questions within different organisation areas, from physical to socio-psychological issues. The user is then presented a table of how s/he is in relation to different norms. The procurement organisation is not involved in the actual organisational changes they do only provide the product to take temperature of the organisation.

#### Step 1 – state the shared object

The shared object is to “develop a good system”.

#### Step 2 – clarify objectives

For the procurer the list of objectives is:

- user orientation
- prototype construction
- communicate with and involve users
- usability student
- requirements formulation

For the developer the list of objectives is:

- customer satisfaction
- adhering to customer requirements
- hiring consultant for usability work

From this list an objectives tree is drawn.

#### Step 3 – work with objectives tree

This step is performed in order to criticize the constructed trees. It is a variation to the “Why? Why? Why?”-technique that can be used to find reasons behind objectives, suggestions or decisions. The How-questions moves the scope of analysis down the objectives tree, and identifies

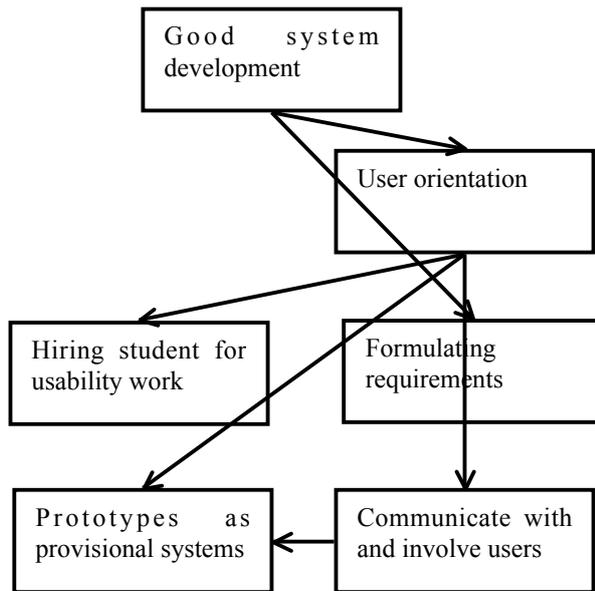


Figure 4. The procurer's initial objectives tree

links between objectives. In this step one objective is added for the procurer

- communicate with and involve users

See figure 4 for the procurer's objectives tree.

The last question can be valuable to answer although it creates a new degree of complexity to the further analysis, which might be unwanted. Still, the answers to the Why-question here could be the source of many future conflicts.

**Step 4 – construct activity system**

The objectives tree process highlights specific kinds of objectives. Performing an activity system analysis provide a framework to formulate other kinds of objectives. In step 4c structural considerations play a role in formulating objectives. Issues of division of labour are easier to understand and formulate within an activity systems analysis, than as a process of formulating objectives.

The analysis clarifies the relationship between the mediating roles and the hierarchical structure of objectives.

See figure 5 and 6 for individual activity systems.

**Step 5 – ensemble objectives tree**

See figure 7 for ensemble objectives tree.

The shared object is put in the middle, to point out that it is shared.

Links between trees are drawn. There is a link between “Formulating requirements” and “Adhering to customer requirements”. If necessary one might draw an arrow to clarify which of the objectives is a means for achieving the other. In this case they construct each other mutually, and one is not the means for the other, or vice versa.

We note potential conflicts between user-orientation/customer-satisfaction and usability student/hiring consultant for usability work.

**Step 6 – ensemble activity system**

This is an optional step, conflicts and potential conflicts identified in step 5 may be enough to deal with to sort out the most obvious obstacles for a good cooperative project with acknowledged, accepted and discussed differences in overall objectives while preserving the super-ordination of the shared object.

If one is performing step 6, here follows an example of what might be found.

With this view it becomes quite clear that the potential conflict user-orientation/customer-satisfaction is a conflict with effect on the different activity systems overall objectives. Thus, it effects the outcome of the project.

The developer's view of system development is mediated by customer satisfaction and division of labour solves usability concerns. The division of labour restricts the participation of other consultants in the customer process. The overall objective for the developer then becomes an issue of keeping a good relationship with the customer.

The procurer's view of system development is mediated by user orientation, and the fact that the procurer hires the developer to develop a good system for her community of users. The overall objective for the procurer then becomes an issue of usability.

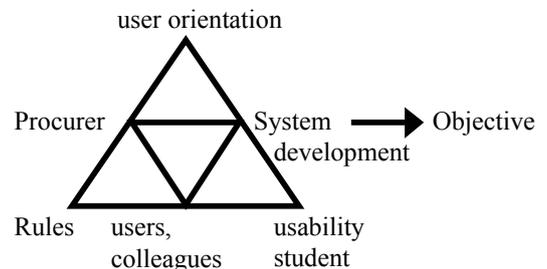


Figure 5. The procurer's activity system

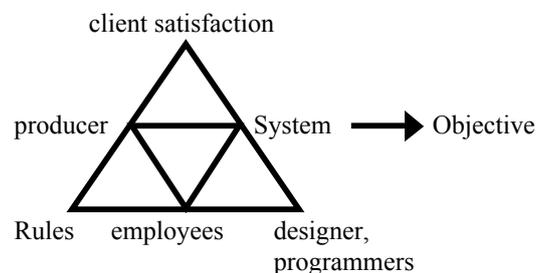


Figure 6. The developer's activity system

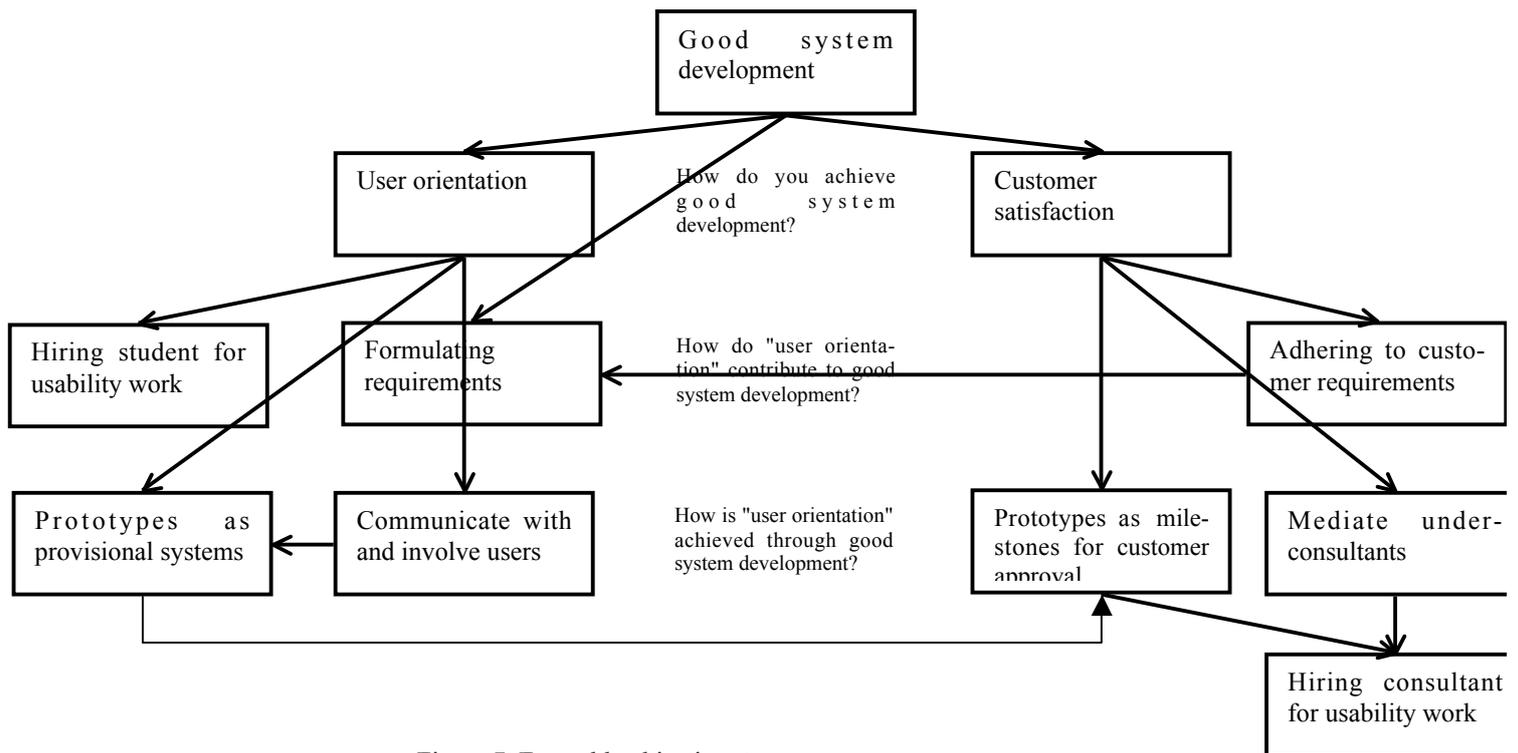


Figure 7. Ensemble objectives tree.

Other conflicts are available for analysis, such as those based on the ways division of labour is set up within the two systems.

**COMMENTS**

The method lends a set of techniques from Jones (1992) and Cross (2000). The methods have been slightly adapted to be applicable within an activity theory framework.

Method 3.1 Stating objectives (Jones, 1992 pp 194-200), as well as Clarifying objectives (Cross, 2000 pp 61-76), assume that the developer should adapt to the sponsor's expectations and her reasons for these. In the method presented here we assume that the developer and sponsor have a socio-cultural history which construct their different interpretative perspectives. We also assume that the developer does not enter into the procurer's activity system but rather strives for preserving his own.

Stating objectives and Clarifying objectives focus on objectives for a design problem that need a solution. In the method presented here the goal is not to find a solution, but to find discrepancies and conflicts. These might need to be solved, or just to be identified and better understood.

The overall process is collected from Clarifying objectives, but variations might be imported from Method 3.1.

Why-why-why (Jones, 1992 pp 318-319) in the context of objectives trees, lends itself to a variation that pushes the answers in the other direction than originally intended. The result is How-how-how. But there is also an interesting

aspect to this seemingly linear figure of thought, which is reflected in the questions of step 3 above. There can be asked two quite different How-questions under the circumstances described here. The direct How-question, but also a How-question that connects the shared object with a stated objective (see figure 7 for examples).

There is an assumption behind the activity systems; the object is similar in all activity systems, and the subject is defined. This means that the object-subject axis is fixed, and the analysis circles around this fixed axis.

The ensemble activity system is optional. One reason is the complexity it would exhibit, with connections between different objectives. It requires a trained eye to consider the possibilities of such a structure. Still, it is useful, and given that the objective trees at some stage are drawn with a powerful structured drawing tool, the transformation as such from a tree to the two triangular forms need no the complicated.

**Application**

Careful statement of objectives and conflicts is a valuable asset in all kinds of cooperative situations. In such situations it is important to acknowledge the fact that actors have different socio-cultural backgrounds, understand and act on this in good time before conflicts arise.

**Learning**

Working with this method is a learning process in itself. First of all it is reflexive learning, trying to understand ones

owns objectives. Secondly, it is cooperative learning within the scope of a joint project, establishing a shared understanding of each other's objectives, the gains and conflicts that these may harbour.

The basic steps of the method are not hard to learn to perform, although it might require some tediousness.

### **Cost and time**

The research and effort needed to work through the steps of identifying conflicting objectives may take longer than just writing the usual documents, and altering requirements along the way of the project. The possible increased cost for performing this analysis is small in relationship to the pitfalls it is intended to avoid; projects that fail because objectives have been misunderstood, potential conflicts have been disregarded, or the effect of differences in objectives could not be anticipated due to interpretative blindness.

## **DISCUSSION**

The provisions for a method as the one suggested will be discussed in three parts: the lack of history in design methods, the question of system fusion, and whether it scales or not.

### **Design methods lack history**

Many design methods assume that the designer and her methods are neutral with respect to the needs and goals of the design client. In Gulliksen & Göransson (2003) and Jones (1992) this assumption shows through in the way the author implicitly formulate the role of the designer. In very few places the designers role as an agent or her interpretative role is touched upon. This is especially alarming in the parts of the relationship where the groundwork for design is laid; when clarifying the objectives.

Activity theory, on the other hand, teaches us that our socio-cultural history affords us to act as interpretative agents in all situations, more or less consciously. This is of course true for designers and clients alike. When people act within the same activity system such agency is part of the everyday practice. It can be hard to see, understand and act well under such conditions.

This suggests that it will not be less complex to understand when two different activity systems meet under the assumption that one shares the object or aim. So, when clarifying objectives the two different activity systems will provide very different grounding for the interpretations of what the objectives are, which objectives that are relevant, as well as which objectives that belong to one of the actors and which are shared.

Acknowledging these socio-cultural differences when clarifying objectives could provide a better understanding of the objectives stated, and the reasons for including them, as well as understanding the objectives that are not shared.

### **To fuse or not to**

Shouldn't we strive for fusing the two activity systems and form a single activity system for the system development activity?

There are two main objections to this. The first concerns the fact that the formation of such an activity system is not history free in itself; so it inherits conflicts. The second is that the legacy of the two original activity systems will dominate for a long time before the new activity system has formed a life of its own.

But isn't activity theory analyses aim to uncover such conflicts? Why complicate the matter by contrasting two systems with each other? The main objection to this relates to the second objection above. The formation of an analysable activity system for a specific development project grows during a period of time where two different systems meet and co-develop. The effect this period of formation has on the resulting activity system is important in order for us to understand the reasons behind conflicts. By actively and early uncovering conflicts between the two systems, one can systematically understand and lead the formation process, in order to create a better-suited project.

### **Does it scale**

One last point to discuss is whether a technique such as the one suggested scale. Three aspects of scaling may be relevant; thousands of objectives, tens of activity systems, scaling over time/projects.

In the first case this method is probably as hard to work with as requirements engineering and prioritisation (Carlshamre, 2001). The more objectives identified the more is required of the well-trained eye and gut-feeling, if there is not a competent and powerful tool to support the process.

In the second case this method gets close to linearly harder. Working with a lot of activity systems makes it possible to look at chunks of objectives or chunks of systems at a time, and relate these parts to the whole.

In the third case, the scaling over time and projects actually strengthens the applicability of the method. The more one learns to use the method, the better one will be at performing the different steps. The learning factor is also important when it comes to re-using structures of objectives over time and developing them as the organisation develops. IT is also possible to relate objectives between projects and thus find support for objectives in finished projects.

## **CONCLUSION & CONTINUATION**

Given the process of establishing relationships before executing a system development project and that this is exactly the point where activity systems meet, conflicts and asymmetries in objectives, needs, goals and competences occur, despite a shared object. In order for these not to be assumed, inherited or in other senses reinforced while

executing the project, they need to be scrutinized during the establishment of the relationship. If not fleshed out between actors and stakeholders such conflicts might lead to unwanted breakdowns in processes, further conflicts, or misinterpretations of intents.

We have proposed one way of eliciting these possible conflicts, and pointed towards an application. We conclude that the conflicts between different possible actors in a project could potentially be of the kind that it would be positive for a project to form an understanding of these conflicts and asymmetries during the establishment of project stakeholder relationships.

Given the possible conflicts that might be elicited with a multiple activity system approach, it is important to understand whether such an analysis can provide the stakeholders with a means to perform better together. To be able to do that the method needs to be further developed.

#### Acknowledgements

This research was made possible by a research grant from the Swedish Innovation Agency, Vinnova. We wish to acknowledge the participants in the referenced study, and the invaluable efforts of Emma Borgström, and research colleagues Erik Markensten and Anna Stockhaus.

#### REFERENCES

- Artman, H. (2002). Procurer Usability Requirements: Negotiations in contract development. *Proceedings of NORDICHI 02*. pp 61-70.
- Borgström, E., Artman, H., Holmlid, S. (2001). Beställar- och leverantörsinteraktion i systemutveckling. *In proceedings from STIMDI '01*, Stockholm [Interactions between procurers and producers in system development].
- Carlshamre, P. (2001). A usability perspective on requirements engineering – from methodology to product development. *Linköping studies in Science and Technology, dissertation no 726*. Linköping: Linköpings universitet.
- Cross, N. (2000). *Engineering Design Methods: strategies for product design* (third edition). Chichester: John Wiley and Sons Ltd.
- Gärtner, J., Wagner, I. (1996). Mapping Actors and Agendas: Political Frameworks of Systems Design and Participation. *Human-Computer Interaction*, vol. 11, pp. 187–214.
- Greenbaum, J. (1993). A Design of One's Own: Towards Participatory Design in the United States. In Schuler, D., Namioka, A, eds (1993). *Participatory Design: Principles and Practices*, pp 27-37. Hillsdale, NJ: Lawrence Erlbaum Associates
- Grudin, J., 1991. The Development of Interactive Systems: Bridging the Gaps Between Developers and Users. *IEEE Computer*, 24, 4, 59 –69.
- Gulliksen, J., Göransson, B (2003). *Användarcentrerad systemdesign*. Lund: Studentlitteratur.
- Hjalmarsson, A (2004). Att etablera och vidmakthålla förbättringsverksamhet - Behovet av koordination och interaktion vid förändring av systemutvecklingsverksamheter. Linköpings universitet.
- Holmlid, S (2004). Issues for cooperative design: A procurement perspective. In Proceedings from PDC 04.
- Holmlid, S., Artman, H. (2003). A tentative model for procuring usable systems. *HCI International 2003*.
- Jones, J., C. (1992). *Design methods. Second edition*. New York, NY: Van Nostrand Reinhold.
- Löwgren, J., Stolterman, E. (1998). *Design av informationsteknik*. Lund: Studentlitteratur.
- Markensten, E. (2003). Procuring Usable Systems - An Analysis of a Commercial Procurement Project. In Proceedings from *HCI International 2003*.
- Näslund, T. (1997). Computers in Context – But in Which Context? In Kyng, M., Mathiassen, L. (eds), *Computers and Design in Context*, pp 170-200. Cambridge, MA: MIT Press.
- Vimarlund, V., Timpkat (1998) Participatory Design in Economic Terms. Proceeding of PDC-98. The fifth Biennial Participatory Design Conference pp: 11-17.

# RUBICON: Transforming work practice descriptions into IT requirement elicitation via AT modelling

**Kristina Lauche**

School of Psychology

University of Aberdeen

Aberdeen AB24 2UB, Scotland, UK

+44 1224 272280

k.lauche@abdn.ac.uk

## **ABSTRACT**

This paper describes RUBICON, a combination of research methods aimed at the early stages of requirement analysis that bridges the gap between social science descriptions and IT systems design. It involves a three-stage process of observations, interpretation and a special task analysis for requirements. The elements of the method are described and examples given from the development of a multi-user tabletop system are provided.

## **Keywords**

Activity theory, task analysis, requirement analysis

## **INTRODUCTION**

RUBICON is a combination of research methods that bridges the gap between the mostly descriptive accounts of work practices and the context of use for new IT system on the one hand, and the need to specify requirements on the other hand. There is a general academic consensus that one needs to understand the context of use by studying work activities in order to design supporting IT systems [16]. Yet in practice this is still far from common. One underlying reason seems to be that social scientists and system designers live in epistemologically different worlds. Social scientists produce descriptions and explanations of reality but not necessarily a basis for design interventions. There is no straightforward methodology to transform these results of workplace studies into system design [20, 27]. At some point system developers need to cross this Rubicon to make design decisions. The aim of the RUBICON method is to link the work of social scientists to that of system developers under an activity theory framework. The method proposed in this paper enables researchers to collect context information, model its implications for the design process and then deduct and elicit specific requirements.

## **Activity theory as a framework**

The RUBICON method is based on the idea that in order to understand and design artefacts such as IT systems in use, one needs to conceptualise the activity system that forms the context of use [4, 16, 23]. Activity theory [9, 10, 19] provides a framework to understand real-world activities and the existing division of labour from a cultural-historical perspective. The activities of an individual or a team and the object of their work are seen in the social (technical, organisational, political) context in which they

originated and developed. Artefacts form part of the context that humans create for themselves and contribute to the cultural history. Activity theory therefore conceptualises artefacts (including information technology) as an integral part of the development of an activity itself [3, 16]. So far, there has been little methodological guidance on how to analyse artefacts beyond individual use. The tools introduced here combine ethnographic approaches and document analysis. The artefacts are used as a trigger for an interview conversation with potential users of an IT system as a key for understanding their work.

## **Outcome and scope of method**

The outcomes of the RUBICON method are insights into the application domain and requirement specifications for IT development. Both are typically recorded in the form of a report. However more importantly RUBICON describes a procedure that requires IT developers to familiarise themselves with the context they are designing for, and social scientists to specify the design implication of their empirical findings. The main outcome is therefore an interdisciplinary collaborative work experience that will materialise in a report and eventually an IT system.

The method is aimed at the early requirement analysis prior to the engineering and systematisation of a list of specifications. It is mainly concerned with what is desirable for the task and the activity system in question. It does not answer the question what is technically feasible or economically viable. The approach can be applied for complete innovations but also for re-design of existing systems. RUBICON was developed for collaborative planning tasks in architecture and design but the general procedure can be adjusted to other CSCW settings.

## **Intended users and required competencies**

The RUBICON method is aimed at designers and researchers who study work practices and transform their findings into requirements. The approach is of particular value if there is limited direct interaction between system designers and intended users. Then the methods can be used to introduce a degree of participation into otherwise segregated worlds of users and designers. The method can also be used by designers and users themselves but it requires that people assume the perspective of an observer and model user needs empirically before jumping to conclusions about specific requirements.

The competencies required for applying the RUBICON method are an understanding of AT, or appreciation of the role of work practices and the context of use for systems design. The terminology is explained as needed, and experience so far has shown that designers with no prior background in AT developed at least a basic understanding of the concepts with relative ease. Social scientists require an understanding of the design process and the importance of artefacts in activity systems. The methods require observation and interviewing skills and quantitative as well as qualitative analysis of empirical results. The procedure as documented here provides guidance for data collection and analysis.

**PROCEDURE**

The RUBICON method consists of three iterative stages:

**Stage I** aims to create a basic understanding of the work practices and the context of the proposed IT system. It is therefore mainly descriptive with field observations, analysis of artefacts and subjective data on the activity system (such as questionnaires about the scope of collaboration and work environment). The purpose is to conceptualise the task to be able to derive requirements for IT system, not to merely elicit wish lists from prospective users. Preferably these observations are carried out as a collaborative effort of IT developers and social scientists who both take individual field notes. If both cannot be present in the field, they should review each other’s notes and ideally record the events on video for joined discussions.

**Stage II** consists of interpretation and modelling of the findings in terms of the activity system to be designed for. This stage is less formalised than the data collection in Stage I as it is an iterative, hermeneutic process from the observations to the model. It is indispensable that stage II is conducted as co-construction among developers and social scientists, so that both can theorise about the activity system, mutually challenge their assumptions about it and jointly extract design implications for the proposed IT system. The purpose of this stage is to narrow down the scope of the IT design and to decide on a feasible roadmap for the development. The result should be communicatively validated with members of the activity system.

**Stage III** focuses on eliciting specific requirements for the system development. By this time, the developing team will have a broad concept of the system to be designed and can start to integrate specific requests from the prospective users. The method suggests a number of standard questions for a semi-structured interview, which can be adapted or extended to suit the purpose of the investigation. Depending on the scope defined in stage II, the sample of prospective users, tasks and activity systems may be different or identical to the one from stage I.

The three stages are connected through the interpretation efforts and form an iterative rather than strictly sequential process. The purpose of the stages is to emphasize that no listing of requirements should be attempted without any background research, and that a theoretical analysis is necessary to proceed from data to design recommendations.

The following sections describe the use of the RUBICON method as part of the development of a multi-user table-top system.

**STAGE I: FIELD OBSERVATIONS**

The RUBICON methods have been triangulated to avoid common methods variance, i.e. problems related to using a single method such as a questionnaire will affect the entire study if only that one method is used, and correlations found may be a result of the instrument rather than reality. Table 1 provides an overview of the methods and their rationale.

Research method	Rationale for inclusion
Observation of collaborative activity	Understanding of work practices, collaboration in context, ecological validity, assessment of potential for IT systems
Analysis of artefacts	Artefacts are theoretically important in activity theory as mediators between actors and their objects. They are practically important for system design as they may be replaced or augmented by the new system
Questionnaire ProTeam	Validation of work practices, scope of collaboration and technology use in larger sample, subjective perspective on activity system

Table 1: Research methods for stage I of RUBICON

**Definition of sample**

Prior to the start of the investigation, the activities to be supported need to be broadly defined. Irrespective of whether the IT development is initiated by a technological possibility or driven by a practical need, the potential application should be defined in terms of tasks and user groups. It is advisable to follow an explorative, qualitative approach at this stage and purposefully sample a range of activities as part of the result of the descriptive stage may be that the nature of the task is different from what was assumed at the start.

Following the definition of potential tasks and user groups, access to a sample of projects and sites should be negotiated. We have experienced little difficulty in negotiating access for internal meetings but observing interaction with clients or patients may be more restrictive. Observation and interviewing normally does not subject participants to any undue harm or risk unless it reduces concentration for a potentially dangerous task. The most important ethical considerations at this point are consent to taking part and confidentiality of results. Participants should be briefed and debriefed about the purpose of the research and consent explicitly to be recorded in whatever form. Ideally, personal or sensitive information is not recorded at all or anonymised appropriately.

**Observation**

The potential users are to be observed as part of their normal work practice. The RUBICON method employs

non-participating observation, i.e. the researchers are not directly involved in the activity itself and should not attempt to influence the activity system at this stage. We have found that compared to participant observation, the benefits of having a dedicated person to observe outweigh the potential problems that may arise if users feel uneasy about being observed.

As not all complex tasks can be readily understood by mere observation, it is useful to gather preliminary information from documents such as meetings minutes, and ask for a briefing about the roles and topics from a representative of the activity system.

During the observation of the collaborative activity, the researchers record field notes on a standardised form [17]. The form developed for the observation is simple and straightforward but tried and tested. A4 sheets are prepared with a table as shown below, also indicating date and place of the observation and the overarching topic under discussion.

Time	who	Action, content	Memo

Figure 2: Form for observation notes

The observers note time and actor and the gist of the argument or action. Observers should be trained to distinguish between observations and interpretations, and note them separately. Any comments for the interpretation or ideas for design implication are noted in the “memo” column. Other researchers have typed conversation notes directly on a laptop, or have used online categorisation. Although electronically stored field notes ease the processing of data, our experience has been that this form of note-taking completely absorbs the attention of the observer. Our observers were busy recording but unable to understand and form a mental model of what is going on at the same time.

If understanding the context constitutes a research aim in itself, it may be appropriate to transcribe part of the video recordings selected. The notes can be used to extract critical situations of importance for the progress of the project [2] for further analysis. In this case, the corresponding video record is transcribed and analysed sentence-by-sentence using a category system [17]. The system is based on protocol analysis [6] and coding schemes for meetings from social psychology. The categories differentiate between three basic processes: firstly the cognitive task of problem solving, secondly the social and emotional processes in the group, and thirdly the monitoring of group discussion. However, such a detailed analysis is an immensely time-consuming process and normally not directly relevant to the development of IT systems. The main advantage of a

video record for design purposes is to retain a visual narrative for team members involved in the development of the IT system who have not witnessed the interaction themselves.

**Analysis of artefacts**

For the analysis of artefacts, the information is obtained from documents or and/or elicited in an interview with users or authors of the artefact about its history. The method can also be applied to observing the creation of a common objectification, or to diagnose the state of collective knowledge and its materialisation for practical purposes. It is advisable to decide which artefacts shall be regarded as relevant for the study due to their centrality within the activity system. The interview guideline classifies artefacts according to formal features and their individual, cognitive as well as social and organisational function.

Secondly the method employs the Objectification Inventory [28], an instrument to analyse artefacts and their use in design teams. The Objectification Inventory is based on Leont'ev's [19] concept of collective objectifications as an important mediator within activity systems and similar categorisation schemes used in design research [24, 26]. It is designed to analyse documents and artefacts not generated for the purpose of research but as part of the activity to be analysed. Examples for objectifications in design include knowledge memories (e.g. databases, protocols of team meetings), visualisations in the problem solving process (e.g. physical models, photographs, sketches), documented general practices (e.g. principal solutions for design problems, user hints, heuristic rules), reference books (e.g. catalogues), tools and facilities (e.g. software tools).

The Objectification Inventory covers five dimensions to analyse each artefact in use:

1. Content of the artefact and its formal features: description of the artefact and/or copy for further reference, classification of type (slide, sketch, physical model, software), features (colour, 2/3D, manual/CAD), and document management indices;
2. Context of use and attribution to subtask of the project: when and how is the artefact used? How is it localised in the activity system physically and conceptually? (categories will depend on the task/project)
3. Individual cognitive function of the artefact for the activity (external storage of information, solution finding, evaluation, record of activity);
4. Collective and symbolic function in the activity system: what is the artefact being used for? How often? By whom? What is the degree of collectiveness and diversity of users in the generation, usage, further development, maintenance of document?
5. Usefulness from an organizational perspective: can the artefact be used and shared by others not involved in its creation? Is it transferable and accessible to other users and projects? How much effort was required and was this worth its while?

The use of the Objectification inventory shall be exemplified with one artefact that we observed during its generation and in a second, modified form. We observed an internal and an external software designer discussing their concepts for the interface of a tool for customer driven design. They carried out a process mapping of the proposed online communication between sales personal on client visit and engineering staff via a joint product database to configure a modularised product according to the client's requirements.



Figure 3: The generation of an artefact as part of a collective design process

At first the two software designers developed options and identified where they had different concepts of how the interface should be. They arranged existing material and modified post-it notes on a flipchart during their discussion. The result showed traces of their collective design process and still had the flexible qualities of a sketch. On reporting their results to the project manager, changes were made readily. It scored high for team scores but low for readability to others and retrieval.

Then one of the engineers transformed the artefact into a PowerPoint presentation, which had none of these team sketch qualities: The slide gave the impression of a finished, normative description of the customer driven design process. The collaborative authorship was no longer identifiable.



Figure 4: Artefact transformed into printout of file

In the second picture, an attempt was made to retain the quality of the large size: the PowerPoint presentation has been printed as a poster. However, the printout was less flexible than the original sketch. Other members of the team used the second artefact as a result without developing it further. Hence, the team score went down. The electronic file would have been easy to change but it was not available to the other team members. Therefore, the score for knowledge management was not as good as could have been for an electronically stored document.

### Subjective perspectives

A subjective measure such as the ProTeam questionnaire [18] can be used. The main purpose is to check the findings from the observation about forms of collaboration in a larger sample. Questionnaires are a cost-effective form of data collection but allow no dialogue with those who answer them. Other forms of eliciting subjective views such as interviews, focus groups or open space events are also possible to obtain a description of work practices and gather issues from a user perspective.

The ProTeam questionnaire was originally developed to measure teamwork in product development teams. In its full-scale version it covers the following aspects (see table 2 below):

Scale	Items	$\alpha$
Individual attitudes:		
- Cooperative interdependence	5	.62
- Competitive interdependence	5	.71
- Individual independence	5	.78
- Interdisciplinary collaboration	5	.65
Team characteristics		
- Collective efficacy	8	.78
- Perceived quality of teamwork	11	.82
Organisational factors of teamwork		
- Scope for collective action regulation	10	.38
- Perceived organisational support	6	.65
Technology use, socio-technical issues	11	N/A

Table 2: Scales of ProTeam questionnaire with item numbers and internal consistency (Crombach's  $\alpha$ )

The scale that is most directly relevant to the design process is the scope for collective action regulation (see figure at the end) and the perceived quality of teamwork (see table 3 below). For most scales, the answer format is a 5-point Likert scale from "strongly agree" to "strongly disagree".

1. I have helped to finish work of other members.
2. Other team members were willing to finish my work.
3. We keep each other in the loop about relevant information and changes.
4. Other members of the team let me know what they expect of me.

5. I have checked with other members whether they would be affected by any of my decisions.
6. If other people's action affected my work, I got in touch with them to resolve it.
7. I am clear about the other member's roles and task.
8. Team members are prepared to question the basis of what the team is doing.
9. Our team critically appraises potential weaknesses to achieve the best possible outcome.
10. We share information about new technology or working methods.
11. I often receive helpful advice and hints from team colleagues.
12. We challenged group members who took a laid back attitude.
13. I have shared my knowledge and expertise with others.
14. When I run into difficulties with a certain job, I could always rely on colleagues to help me out.
15. If a colleague notices a mistake, they point me to it discretely and constructively.

Table 3: Items on perceived quality of teamwork

### STAGE II INTERPRETATION AND MODELLING

The process of interpretation is not formalised as such but broadly follows the process of qualitative data analysis. Typically, the observations are used to model the activity system, to content analyse the meetings and to assess the task against criteria of human-centred job design. These criteria are part of normative models about intrinsically motivating tasks, derived from activity and action regulation theory [13, 30]. They postulate the need for meaningful work, a sense of personal control, social interaction and physical activity, and the potential to learn and develop as part of the work. Jobs vary immensely in the degree to which they are designed to suit human needs, and the implementation of IT systems can improve or worsen this situation. The RUBICON method follows this normative approach in its focus on the task and the human actors as the focus of the design.

The first step of the interpretation is a descriptive summary of each observed episode. Those descriptions are then content analysed, ideally by a social scientist/ psychologist and a designer. Each meeting should be summarized with respect to the scope of the design project, the type of collaboration, the use of tools and artefacts, and the design outcome.

By taking the next step of formulating implications from the findings, one crosses the Rubicon from descriptions to design and stipulates needs and requirements for an IT system. The RUBICON method views this first conceptual step as the task of an expert designer who has been trained to formulate requirements in a solution-neutral form and aims to avoid idiosyncratic wish-lists at this stage. The

broad concept is then shaped in a more participatory process in stage III.

### STAGE III: SPECIFIC TASK ANALYSIS FOR REQUIREMENTS

At this stage, the IT system has been broadly conceptualised and specific requirements need to be identified. Potential users are introduced to the concept and interviewed about specific needs and gaps in their current environment. They are asked to speculate how they would use the proposed IT system and generate a description of the new context of use. Typical user input at this stage concerns interface design, connectivity requirements, compatibility to other soft- and hardware and information about user groups.

The third type of instrument used to inform the system development is task analysis. In distinction to the mostly descriptive form of task analysis in the Human Factors tradition [1, 15], action regulation theory has been instrumental in promoting instruments for human-centred task analysis and job design. This approach to task analysis shares its basic principles with socio-technical system design [5, 8] and the evaluation of psychological impact of job design [21]. Among these principles are opportunities for personal control and skill use, variety, clarity of consequences, opportunity for interpersonal contact and physical activity [29, 30]. Various instruments have been developed to assess if a job and the working environment provide a sustainable basis for managing human resources in terms of learning opportunities and resources for coping with potential stressors [7]. From this group of instruments, two that focused on collaboration and technology use were chosen as a basis for a specific interview guideline for this study. The first instrument assesses the scope for collective action regulation and self-management in teams [31, 32]. The empirical results from studies in production teams suggest that a higher level of collective action regulation leads to more shared responsibility and better results at group level [28]. The second instrument is part of a set of guidelines for user-oriented software design [22]. It provides detailed information on technology and information currently available and strategies for change. Based on the theoretical framework, a set of structured questions were formulated, which comprised of the organisation and its objectives, the task and its scope for collaboration and interaction with other designers, other functions and clients, and the current technology in use, including details of the CAD software and specifications for interfaces with other systems.

### EXAMPLE: DEVELOPMENT OF BUILD-IT

The example refers to the use of the proposed method as part of the development of the Build-it system, a multi-user system for co-located collaboration of designers during conceptual design activities and interactions with their clients. Both the empirical research and the concept of the Build-it system were grounded in activity theory [11].

Build-it is a tabletop system where multiple users simultaneously interact on an augmented reality interface. Physical bricks are used to activate the virtual objects on a table and

wall projection and all users have shared access to them. Users can shift between aspects of the configuration task and individual and group work. Data can be incorporated or exported to CAD systems for external work. The interaction takes place on a large-scale projection on a table thereby unifying action space and perception space.



Figure 1: Interaction with the BUILD-IT system

The strategic aim of the Build-it project was to provide an intuitive means of co-operation for users, inhabitants and operators with professional planners or architects. This way, the users of built environment or a new factory should be enabled to participate in a democratic decision making process about the design without having to learn specific software skills. The planners on the other hand would benefit from an output format for these discussions exportable to professional software packages.

The underlying assumption of the design philosophy could be specified as follows:

1. Collaboration was seen as vital in the design process. It would occur co-located in small groups, where a tabletop system would constitute the best interface approach.
2. Interaction with clients and users was seen as limited because it lacked appropriate means of support. If users were given an intuitive system, they would be able to participate more actively, thereby making the design process more democratic.
3. For most situations in collaborative design the task could be reduced to assembling modular objects from a library of standard materials or machinery. The objects could be imported from professional CAD software outside the system boundaries of Build-it.

To test these assumptions, the study explored how and with whom designers in different industrial sectors actually interact and how they use artefacts to support their work.

#### *Results of the observations for Build-it*

In the example of the Build-it system, work practices were observed in architectural design, urban planning and layout planning in the process industry to machine configuration.

They were analysed with regard to when and how designers collaborate and how this collaboration could be supported.

The meetings ranged from regular review meetings of two to five hours and large-scale workshops of three to five days, with a total recording time of 96 hours. The shorter meetings were project reviews with an emphasis on coordination and idea generation meetings for subtasks of a project. These meetings involved on average 7 participants. The longer meetings were strategic retreats of whole R&D teams to define the scope of a new project, with an average of 19.6 participants. All teams were of mixed discipline (mechanical engineering, marketing, software and industrial design), often from more than one company. The participants were highly educated specialists, predominately male with only 10% women.

Participants reported that the majority of their work was computer supported both in individual and collaborative tasks. Email and telephone were used in distributed work across locations and organisational boundaries. Co-located work was also supported by shared access to files from some form of product data management. However, during the meetings observed, virtually all communication was face to face with little IT support. Shared meeting notes were produced as flipcharts or as idea assemblies on pin-boards and mostly captured as digital photographs. These notes were later modified and used for strategy documents, specifications, drawings or prototype. Only unidirectional communication i.e. summaries of the work of a subgroup, were prepared and presented electronically as PowerPoint presentations. Only one company regularly used video conferencing for their reviews to communicate with their North American partners, another one had done so for the time their managing director spent abroad. In two companies a dedicated person kept a verbal record mostly of the secondary task of co-ordinating and decision making as minutes on a laptop during the meeting but usually without a link to the graphical output of the design process.

Overall the observations indicated that the innovation task as such does not start as “born digital” and is only partly supported by adequate IT in the early stages. There seemed to be a potential to support collaboration and the interactive periods of the meetings in a format that could be retrieved and reused in later stages both individually and collectively. However, the results also indicate that the intended functionality of the Build-it system could turn out to be an over-simplification of the variety of activities in the design process.

#### *Results on artefacts from the Build-it development*

The analysis of artefacts for the Build-it project indicated that designers use a broad range of technologies and means of communication. Engineering departments communicated among themselves or with their clients via dedicated networks and internal servers but also used conventional media such as telephone and fax to clarify specific questions and establish internal coordination about sub-tasks. The fax documents contained technical drawings in 2D with handwritten emphasis and comments. 3D CAD models were used during individual engineering work for

detailed design and during project meetings to provide an overview of the issues under discussion. The shared objectifications consisted of both physical and electronic records, which were combined seamlessly as long as the teams had successfully negotiated a co-located arrangement for the duration of the project. Interviewees viewed their means of communication as mostly adequate for distributed work apart from problems with collaborative editing and unintentional overwriting in one software package. However, none provided an interface for collaborative interaction with the design data.

The communication between service personnel and engineering design traditionally used a low-tech approach with Polaroid pictures and handwritten notes with piece identification numbers. The annotated pictures provided a problem description for office staff to solve the specific problem and keep the designers involved in ongoing design improvements. The notes were either faxed or posted. The technology then changed to digital cameras, which increased the speed of the process slightly but mainly improved the quality of the pictures significantly.

Both paper and electronic means were used at various stages during the design process. Objectifications that became central to the collaborative work were regularly referred to and posted on walls generally underwent an evolutionary change from a flipchart or paper sketch to an electronic document with various updates.

The analysis of artefacts identified a number of requirements for a multi-user design tool. It should allow for collaborative editing and version control, objectifications should be easy to change and the system should interface with other forms of representations on paper or as CAD files. The design philosophy of the Build-it system is consistent with these requirements; they are merely technical problems for the software development.

*Questionnaire results in example*

The following section reports the descriptive results of the questionnaire for collective action regulation followed by the modelling of individual, team and organisational influence on perceived quality of teamwork.

The descriptive results on collaboration point out the prevalence of co-located work: 35% responded that most or all team members were located in the same room, and a further 55% said they were located in the same building. For the remaining 10% most team members were situated within one to three hours travelling distance. Only 5% answered that some of their colleagues were located abroad. These frequencies indicate a co-located normality where computer-supported collaboration across distances is only relevant for a minority of people and activities (e.g. one company had outsourced the software development abroad). However the majority of interaction could potentially benefit from a co-located multi-user system like Build-it.

The frequency of meetings varied significantly between the two sampled organisations: In one organisation 50% the respondents met on a daily basis and 36% once a week, whereas in the other organisation the largest proportion (35.4%) met less than once a month, followed by 27%

meeting once a week. The purpose of these meetings was in most cases internal co-ordination (52%) or information (14%); only 34% met for collaborative work. These results paint a less convincing picture for the actual use of a collaborative design tool.

The content of collaborative work was mostly seen in the conceptualisation (76%). Later stages were mentioned less often and only 12% indicated that their team was engaged in four or five of the listed activities together (conceptualisation, structuring of module, design of modules, detailed design and correction). These results are in line with the assumption that it would be most important to support collaborative activities in the early stages.

Compared to the ideal of integrated product development and self-managed teams, the scope for team decision making was rather limited. In most cases, the authority was attributed to the project manager rather than the team (see table 4).

	Technical director	R&D manager	Project Manager	Relevant members	All members
External representation	6%	15%	<b>70%</b>	9%	0%
Project co-ordination	7%	13%	<b>74%</b>	2%	4%
Detailed co-ordination	2%	1%	<b>55%</b>	24%	17%
Responsible for time, quality, cost	9%	2%	<b>47%</b>	25%	17%
Personnel decision	<b>38%</b>	9%	29%	11%	13%

Table 4: Results on scope for team decision making (in percentages)

These results indicate a medium degree of collective action regulation with most of the decisions made by one person, the project manager. Another indication in the same direction is the fact that 98% received a fixed salary with no project or group bonus. The organisational support for teamwork was perceived as mildly positive (scale mean = 3.58). The ideal of democratic, participatory decision making that the designers of the Build-it system adopted as their strategic aim seems to have little resemblance to organisational structures. It remains an open question whether a new technological option would change the allocation of power.

On the individual level, the attitudes towards teamwork were strongly positive. In terms of Deutsch's (1973) concepts participants showed a preference for cooperative interdependence (mean = 4.17) over competitive interdependence (mean = 2.55) and individualistic independence (mean = 2.39). The scale for interdisciplinary cooperation

received a mean of 4.22. These high means may be a result of social desirability if respondents assumed that willingness to collaborate would be regarded favourably. Still the results indicate that despite the fact that the actual scope for team decision making was rather limited, team members have the potential to meet and view teamwork positively. It is however worth noting that for the scales on perceived quality of teamwork and on interdisciplinary cooperation, two items were excluded to improve internal consistency, which assumed equal participation and free exchange of tasks among all members. Based on the observations and informal communication with participants it may be postulated that teamwork in these design teams carried a certain degree of division of labour between specialists rather than a high degree of polyvalence of skills.

*Stage II: Interpretation of example*

In stage II, the results were interpreted and modelled in terms of the activity theory framework. The context could be characterised as follows (see figure 3).



Figure 5: Modelling the context of design tasks

The subject consisted of a variety of team members with different roles and expertise, aiming at the task of creating a strategy and concept for new product(s). Mediating artefacts were flipcharts with meeting notes, presentation slides or handouts, idea banks, records of user and market research, technical drawings and meeting minutes. The rules adhered to in the observed activity systems were procedures of a design methodology or managerial interventions such as innovation justification. Their consistency and uniformity among members varied between different organisations. The community that shaped the activity system can be seen in the industrial sectors with their different priorities and established work practices, the market situation with competition and collaboration, and the organisation with its governance and internal structure. The division of labour occurred both between team members with their distributed core task elements and across different projects. These results identify design collaboration as a rich and sophisticated context with high variation among influential factors. This made developing IT systems for this activity system not impossible but challenging.

There certainly is evidence of collaborative work in the design process although it seems to take place among

different professions and less so with the end-user or operator. The interaction between the actors in the design process normally took the form of information and co-ordination; true collaboration was not the norm but appeared crucial in initial stages. A lot of this interaction and collaboration happens in co-location and it is indeed poorly supported by IT systems. In this sense, the first assumption as to why a multi-user tabletop system would be desirable could be supported from the investigation.

The second assumption referred to interaction between designers and users. The results make it seem doubtful if the proposed form of teamwork among designers and with end-users of build environment and machinery actually exists. The fact that direct manipulation of objects as an instrument for intuitive interaction with clients was not mentioned as much as expected may relate to the different conceptions about collaboration within design activities. The strategic aim of the development of Build-it was to foster and support participatory design processes. The system therefore implemented the bricks as an egalitarian mode of interaction and intuitive, non-specialist menus. However, in the task analysis, interviewees characterised designing as an expert planning activity. They described it as essential to consult and convince the paying client but not necessarily the operator or end-user. The designers themselves felt that theirs was a complex activity requiring expertise and knowledge that could not be fully embedded in an application, which made designing appear so easy. The new technical opportunity to collaborate using a multi-user interface was not as readily embraced by the designers because the logic of the Build-it system assumed a type of collaboration different from their organisational reality.

In terms of the third assumption that designing could be emulated as assembling objects, the results serve to caution on its generalisability as there was a huge variety in design activities and potential requirements between different types of design. The observations showed variations in group size and thematic priority, and the artefact analysis identified a multitude of currently used means of communication. In the task analysis potential users also indicated that the requirements for different tasks may be very different depending on complexity and collaborative structures. It may well be that the results are indeed be very specific to the companies included in the sample. It remains difficult to decide which of these requirements to accept as generic for the design of tabletop systems for design, and which to leave for customisation. The basic functionality of the prototype focuses on configuration of existing elements, which is likely to be too concrete and limited for the more abstract, strategic questions and open ended discussions that some of the design meetings dealt with. So far the results do not present a strong case for a unified system for the various design tasks. Yet economically that would be the most viable option for introducing a new technology.

In conclusion it seems that Build-it would be suitable to support part but not all of the collaborative design process. The potential market appears to be smaller than anticipated as the scope of interaction with the users of design was

limited in current practice. One could argue that a multi-user system may indeed change organisational reality, as the Build-it designers had hoped for. However, the high cost of the system that some described as prohibitive made marketing a product that will only make sense in a different organisational reality not an easy task. Alternatively, Build-it could be used as a mere visualisation tool in client interaction, or as an internal means for collaborative design. In the latter case, users should ideally possess technical expertise, sales skills and competence in manipulating the system, which has implications for organisational and skill development.

### **Task analysis in Built-it example**

In stage III, specific requirements for the Build-it system were identified. The study tested the underlying design philosophy of the Build-it system in terms of the tabletop approach, the interaction metaphor and the specific functionality, for those design activities.

The twenty-two semi-structured interviews with designers as potential expert users were recorded and transcribed as summarising protocols. The following section describes the results about the specific task to be supported by Build-it, the potential benefit and technical requirements for each industrial sector.

#### *Architecture*

In the architectural domain, a total of seven designers from four companies were interviewed. The spectrum of tasks included planning and configuration of office layout and furniture, interior architecture and furniture design and project management of large building sites and removal logistics. Most of the planning was carried out two dimensional in a combination of CAD software and manual manipulations of magnetic clips on paper. The designers reported they felt comfortable planning in 2D but realised that their customers were unfamiliar and often had difficulties to mentally construe the third dimension. The Build-it system was seen as an instrument to combine 2D and 3D views.

The architectural companies differed in size and in their internal division of labour: In some cases, the entire design and planning process was carried out by the same people, in other cases customer contact and detailed design were split between senior architectural staff and junior technicians. Subsequently there was an uneven distribution of technical, design and IT skills that sometimes resulted in communication problems and misunderstandings within the company.

All interviewees described their architectural design work as an expert's task to which the client contributes valuable information but does not and should not participate, as their layperson's understanding of design is limited. For their own work, the Build-it system was seen as nice to have but not essential: It might help to communicate between the customer contact and detailed planning work. The designers also felt the "opportunity to use their hands again" and obtain an overview on the large projection were beneficial. The key advantage however was seen in the

possibility to show and explore specific design options with the client as a means of increasing commitment to the sale and the design. Clients would hopefully be impressed by the new, easy to use technology. However, the client interaction was not perceived as a form of true collaboration as assumed in the original design principle of the Build-it system. One interviewee even warned not to make designing too easy so that people are not encouraged to make decisions they do not understand properly.

The Build-it prototype was seen as sufficiently functional: most design tasks in these interior architecture firms involved configuring arrangement from known modular parts, which was supported by the system. The only missing function identified during the interviews was scanning large plans and drawings: Interior architecture has to work within the existing built environment, and particularly for older buildings, plans normally come on paper. Therefore it was suggested that the system should have a scan function, which would preferably retrieve the technical information about walls, windows etc. from the paper plan and make it available for 3D interaction.

#### *Manufacturing systems*

A total of ten people from two companies were interviewed about the potential use of Build-it in designing production systems and complex machinery. In both companies, the design consisted of customer and project specific configurations of modular elements such as drilling or welding units, automated turning or transport of parts, protection for the safety of operators and control units.

In comparison with the architectural firms, the mechanical engineering companies were designing a more complex product and dealing with a worldwide market. Customer representatives, for example from the automotive industry, were not always experts on all of the technical processes involved and had to be educated about functionality and safety aspects of the system. The key challenge for the design and engineering was to achieve the required precision and quality with the shortest running time and highest output of parts per second.

Although the products and markets were different, the process followed a similar pattern in both companies. Within hours of making contact with a potential client, an experienced sales person provided a first rough estimate of the total prize to secure interest. After closer examination of the requirements and conceptual designing, a first offer is produced and discussed with the client. A report on modifications and additional requirements elicited from or provided by the client informs the development of a second, more detailed offer. The current process was very Tayloristic with complicated communication between sales, conceptual design and engineering. Since tendering is generally not paid for, simplifying and shortening the process was welcomed. The main benefit of a system like Build-it was seen in both cases as an interactive planning tool for the first meeting with the client. Modifications could then be discussed in a user-friendly yet electronically re-useable format, thereby shorting the time required for preparing the final offer. As part of the research

collaboration with the companies, one of the engineers estimated cost savings as a result of less reporting, less misunderstandings, better management of change and better evaluation of variants as 50-80%. His report also mentioned improved interaction and collaboration with customers, more commitment, integration of product data, automation of routine calculations and standard solutions. His analysis was based on the assumption of optimal project delivery with no risk analysis of what could go wrong, the likelihood and consequences of non-optimal projects, and how this might be affected by the new technology.

Over all the Build-it system was seen as functional yet requiring integration with other software systems in use. The intuitive interaction for assembly only covered one aspect of the entire process, which at times could be fuzzy or very precise. The sales interaction was portrayed as intuitive and non-technical. Yet for the specification of some components, the easiest form of interaction was seen as entering numbers, a functionality not supported in the Build-it prototype.

#### *Plant engineering*

For the third type of intended applications of Build-it, five designers from a plant engineering company were interviewed. Their products are large process plants such as incinerators or waste recycling, normally carried out as projects over 2.5 years with the operator, representatives from the local community and an architect. Again, tendering was a cost-intensive, long-winded but unpaid process. More than the previous two industries, plant engineering is essentially as three-dimensional task, which can be very complex and difficult to grasp for anyone without years of experience. The emphasis was therefore less on impression management or lean process but more on reducing complexity for the designer and avoiding costly planning mistakes as early as possible. In this case, the designers could see a lot of benefit in using Build-it for their own internal work. At the time of this study, most of the design was carried out with 2D CAD systems and supported by rather expensive wooden models of the plant. The appeal of the Build-it system was in providing 3D functionality without being so “terribly technical”. The designers envisaged using the system for virtual walkthroughs and collision checks, for the simulation of incidents and for planning of escape routes based on templates for the different legislation in different countries.

This design task also challenged the functionality of the Build-it system. The manipulation of height as the third dimension was not as well as developed as the planar interaction. Some tasks such as planning pipelines through the plant seemed not suited for the interaction with the rather clumsy bricks.

#### *Summary of task analysis*

In the three industrial sectors, different tasks with varying degrees of complexity and different forms of collaboration could potentially be supported with a 2D/3D co-located tabletop CSCW system like Build-it. The interaction with clients and the functionality for designing varied,

depending on the design task and organisational practice. In most cases the system was seen as promising in terms of shorter timescales and increased customer commitment. In each industrial sector, some of the designers favoured the Build-it system for its direct, manual manipulation of virtual objects. However, from the day-to-day use of keyboard and mouse, the intuitive handling of objects via the bricks was not seen as universally suitable for some of the very precise, numerical specification or the design of small, linear objects such as pipelines. Most of the designers’ work required complex numerical manipulations of a design, which would eventually take a physical shape as a result of the work of others; however, it did not involve form giving and direct physical manipulation as such. The unification of action and perception space that Build-it was trying to achieve was still present in the division of labour.

### **ACADEMIC REFLECTION**

The academic reflection describes the relationship the Activity Theory and the development of the method. The central concept that informed the analysis is collective action regulation [31]. The concept is part of the theoretical tradition of action regulation theory, which arose from the need to formulate a theory for goal-directed action in the workplace [12, 14, 29]. It was felt that most psychological theories at the time explained human behaviour only for very restricted contexts as they are normally studied in laboratory settings. Neither motivation and wellbeing nor cognitive functions such as planning and memory could be the same for an artificial task in an experiment and for working life for which people have a professional history and experience, and relate to others in the purpose, the means and the activities of their work. The founders of action regulation theory therefore decided to enrich models of cognitive psychology with a background in Activity theory [19, 25]. One of the main schools of action regulation theory developed at the University of Dresden in the then German Democratic Republic who had active links to the Russian community of Activity Theory researchers.

The action regulation theory tradition maintained the notion that human action has to be studied and understood in the context on the activity that it relates to. In the field of work psychology, it has strongly opposed any attempts to isolate “human factors” from the organisational and socio-economical background and the origin and purpose of an activity. It has a Marxists background in its emphasis on the contingencies of the situation and their power to influence what gets done and is seen as possible, rather than selecting individuals with “the right attitude”.

Its specific contribution is a body of empirical research on the mental regulation of work through the re-definition of a task and the anticipation of outcomes, and a set of normative guidelines and methods for analysis on human-centred work design. These guidelines were traditionally aimed at the design of tasks and work environment to ensure that work offered process control and feedback, time flexibility and a variety of skills with an opportunity to

learn and develop. The guidelines can and have also been used to inform the development of IT systems [22]. Typically the methods are designed for work psychologists to provide an “objective” assessment of the work situation as an outsiders based on the set of criteria. The rationale for this expert centred approach is that the work situation will have already influenced people’s perception: they will take for normal what they experience as normality. A subjective assessment runs the risk of so-called pseudo-satisfaction, which implies that people adjust their expectation to the actual experience. The methods do however include recommendations and complains from those involved in an activity.

More recently the social dimension of action regulation theory has been emphasized. It is the merit of Weber [31] to conceptualise the collective regulation of action in the work place by going back to the original sources of AT. It includes any form of team decision making on its internal organisation (allocation of work, collaborative planning of work and co-construction) and the management of the boundaries to other teams or organisations. A higher degree of collective action regulation will allow a team to self manage their process and own the decisions that influence their work activity. While the RUBICON method follows a generic process of understanding work for systems design, its specific research question is aimed at collective action regulation.

The RUBICON method has evolved from the collaboration of researchers and system developers during two projects, the Build-it system as discussed here and Innoplan, a system for collective idea generation in early stages of product innovation. Although all parts of the method were actually used during those two projects, the RUBICON method as it stands now is the result of my reflection of positive and negative experiences trying to understand work for design.

The positive experiences relate to the interdisciplinary collaboration and the degree of understanding I achieved as an industrial psychologist, both for the design task of the user and the design task of the IT developers. Continuous discourse with various colleagues at ETH Zurich and from industrial partners over the course of four years has provided as rich background for the method. The advice to observe and interpret collaboratively with both social scientists and system developers involved is based on those positive experiences where insights and solutions were achieved that had not been possible without this discourse. My attempts to convey the implications of the context of use were never more fruitful than when my colleagues were present during the interview and could see the setting themselves. Their understanding of what they had seen was also never more structured and reflected than when I had asked the questions. The RUBICON method is an attempt to distil this collaborative learning process into a suggested procedure. It remains to be seen if now much of this learning was based on the personal interaction with the individuals involved and if the description of what I did as a social scientist substitutes for employing that expertise

on the project. My suspicion based on my own experience would be that the procedure does not replace the expertise, particularly not the expertise on social science for designing IT systems. Eliciting requirements is not a novice’s task for either side of the postulated Rubicon between descriptive and design science.

Based on this experience, observations were seen as indispensable as they provided an insight into design activities in context. The rather simple form of structured notes proved to be a user-friendly and efficient format. We have sometimes used the more detailed methodology of transcription, protocol analysis and content coding for research purposes but it is a very time consuming process and not directly relevant to the design of IT systems. We also noticed that the results are likely to be affected by the selection of the sample, in our case early stages of product innovation and meetings, which may have resulted in an overemphasis of abstract, strategic design questions and team interaction. For the purpose of system design, a broader range of observations would have been beneficial.

The analysis of artefacts contributed largely to understanding current use and the potential implications of emulating some of this use in the new system. As such, it forms an indispensable step in analysing work practice for systems design. The Objectification Inventory classifies artefacts in a descriptive form that still requires theoretical analysis. Yet the instrument does not provide much guidance for theorising so unless the results are properly interpreted in stage II it runs the risk of generating a list of artefacts from a sample of convenience.

The ProTeam questionnaire was mainly used as time-efficient supplement to the observations. But in its current format it is not essential for the development of IT systems. As suggested, other instruments such as focus groups or interviews could be used alongside the observations.

The interpretation in stage II occasionally occurred in the collaborative revisiting of observation and theoretical modelling as described here. This positive experience as documented in [11] was given normative status as part of the RUBICON method. In the two projects that led to the development of the method, the interpretation was often the lone effort of an academic writing up findings for reports and publications at a stage when the system development had moved on to other problems.

The task analysis combined general and specific questions on requirements and formed an indispensable step in the design of the Build-it project. The specific task analysis was based on a reasonable sample of 22 designers yet only a limited number of companies from a broad range of industries. Yet the method did not avoid the usual problem of interviewing current users for prospective technology. Users tend to formulate their needs as proposed solutions, rather than abstract requirements. These formulations are often influenced by the available technology and current practice rather than new concepts. Some of the requests interviewees made were inconsistent with the basic design principle of direct interaction. All the interviewees

discussed the Build-it system with very little practical experience of using it, thereby idealising its reliability and functionality. None of them anticipated the practical problems that those who interacted with the prototype got frustrated about.

The method as described here is also a result of the negative experience in both projects: The actual task analysis was not always carried out with the appropriate background research, and not all of the descriptive results were incorporated into the system development. As a result, the research highlighted that some the assumptions were in fact inappropriate at a point when it became clear that the system faced difficulties in getting acceptance with users for commercialisation.

The Build-it system received a lot of praise and interest both academically and from potential users. Yet, like other tabletop systems it has not been commercialised as a product. None of the systems have been taken up by a hardware producer for the main computer market. All share the status of experimental set-ups in universities and laboratories. Is the reason for this inappropriate system design, and if so could a better collaboration between social scientists and system developers have pre-empted that disillusionment?

At least some of the problems could have been avoided or noticed earlier had the process followed the RUBICON method more strictly. More background research at the start and an earlier test of the implicit assumptions about the context of use would have been helpful. Different types of design carried different specific requirements and a unified solution is likely to be inappropriate for many. Also design turned out to be less of a collaborative activity than the system developers anticipated.

One could also question why social scientists always take the entire time of the project to arrive at a descriptive account. The Build-it system postulated the coincidence of action and perception space as a key element of usability to lay people. However, the involvement of lay people in the design process occurred less often than expected. Therefore the basic design philosophy may be interesting from an HCI perspective but the lack of professional feature turned out to be more prominent. To that extent, the Build-it project adds to the existing work of social scientists who, in hindsight, identify the reason why the technology failed or did not reach its potential. In order to improve this situation, the RUBICON method as described here focuses on the relevance for the design of IT systems in an attempt to shorten stage I and to allow for more iterations.

Another option may be to do contingency planning for different outcomes of the observations and task analysis and aims to develop modules rather than all potential applications at once. The more precise yet sobering picture of the potential market could also be used proactively in adjusting the development strategy and external communication.

Does the RUBICON method require a background in Activity Theory? In both projects during which the method was developed, none of my colleagues had any background

in AT, or indeed in any other social science model. So it became part of the collaborative work to develop and understanding of AT concepts in the context of the project. The strength of AT compared to other frameworks such as Distributed Cognition is that it explicitly addresses the social-historical context. The visualisation of the triangles requires one to at least specify the internal rules by which an activity system functions and its division of labour. The normative criteria for human-centred job design derive their meaning from this concept of human activity. If isolated from that humanistic grounding, they become open to any kind of opportunistic dispute that system design may be subjected to in a world that is normally not driven by humanistic values.

Would the RUBICON method work without a background in Activity Theory? It may work but it is likely to encounter difficulties a) during the interpretation and theorising stage, and b) if any aspect of the RUBICON method needs to be adapted to the specific purpose of an investigation. Both cases require an understanding of the underlying concepts of an AT approach to system design that cannot be fully described in this paper. In its present form, the method requires at least one expert user to determine if the instruments are appropriately chosen.

#### ACKNOWLEDGMENTS

The research that led to the development of the RUBICON method was partly funded by a grant from the Swiss Federal Commission for Technology and Innovation and a grant from the Zentrum für Integrierte Produktionssysteme at ETH Zurich, Switzerland. Thanks to all industrial partners for their cooperation and to Morten Fjeld, Kerstin Höger and Alexander Verbeck for developing a shared understanding of how to support design processes.

#### REFERENCES

1. Annett, J. and Stanton, N. (eds.). *Task Analysis*. Taylor & Francis, London, 2000.
2. Badke-Schaub, P. and Frankenberger, E. *Management von kritischen Situationen. Produktentwicklung erfolgreich gestalten*. Springer, Berlin, 2003.
3. Bertelsen, O.W. *Elements to a theory of design artefacts: a contribution to critical systems development research*, Aarhus University, Aarhus, 1998.
4. Bødker, S. *Through the Interface- A Human Activity Approach to User Interface Design*. Lawrence Erlbaum, Hillsdale, NJ, 1991.
5. Cherns, A. *Principles of sociotechnical design revisited*. *Human Relations*, 40. 153-162.
6. Cross, N., Christiaans, H. and Dorst, K. *Analysing design activity*. Wiley, Chichester, 1996.
7. Dunckel, H. (ed.), *Handbuch psychologischer Arbeitsanalyseverfahren*. vdf, Zürich, 1999.
8. Emery, F.E. and Trist, E. *The causal texture of organisational environments*. *Human Relations*, 18. 21-32.
9. Engeström, Y. and Middleton, D. (eds.). *Cognition and Communication at Work*. Cambridge University Press, Cambridge, 1996.

10. Engeström, Y., Miettinen, R. and Punamäki, R.-L. (eds.). *Perspectives on Activity Theory*. Cambridge University Press, Cambridge, 1999.
11. Fjeld, F., Lauche, K., Bichsel, M., Voorhorst, F., Krueger, H. and Rauterberg, M. *Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware*. *Computer Supported Cooperative Work*, 11. 153-180.
12. Frese, M. and Zapf, D. Action as the core of work psychology: A German approach. in Triandis, H.C., Dunnette, M.D. and Hough, L.M. eds. *Handbook of industrial and organizational psychology*, Consulting Psychologists Press, Palo Alto, CA, 1993, 271-340.
13. Hacker, W. Action Regulation Theory - A practical tool for the design of modern work processes? *European Journal of Work and Organizational Psychology*, 12 (2). 105-130.
14. Hacker, W. Action regulation theory and occupational psychology. Review of German empirical research since 1987. *German Journal of Psychology*, 18 (2). 91-120.
15. Kirwan, B. and Ainsworth, L. (eds.). *A Guide to Task Analysis*. Taylor & Francis, London, 1992.
16. Kuuti, K. Activity Theory as a Potential Framework for Human-Computer Interaction Research. in Nardi, B.A. ed. *Context and consciousness. Activity theory and human-computer interaction*, MIT Press, Cambridge, MA, 1996, 17-44.
17. Lauche, K., Ehbets Müller, R. and Mbiti, K., Understanding and Supporting Innovation in Teams. in *International Conference on Engineering Design*, (Glasgow, 2001), WDK, 395-402.
18. Lauche, K., Verbeck, A. and Weber, W.G. Multifunktionale Teams in der Produkt- und Prozessentwicklung. in *Zentrum für Integrierte Produktionssysteme der ETH Zürich ed. Optimierung der Produkt- und Prozessentwicklung. Beiträge aus dem Zentrum für integrierte Produktionssysteme*, vdf, Zürich, 1999, 99 - 118.
19. Leont'ev, A. *Activity, Consciousness, and Personality*. Prentice-Hall, Englewood Cliffs, NY, 1978.
20. McCarthy, J. The paradox of understanding work for design. *International Journal of Human Computer Studies*, 53. 197-219.
21. Parker, S. and Wall, T. *Job and task design*. Sage, Thousand Oaks, 1998.
22. Rauterberg, M., Spinas, P., Strohm, O., Ulich, E. and Waeber, D. *Benutzerorientierte Software-Entwicklung*. vdf, T.G. Teubner, Zürich; Stuttgart, 1994.
23. Redmiles, D. Introduction to the special issue on activity theory and the practice of design. *Computer Supported Cooperative Work*, 11 (1). 1-11.
24. Römer, A., Weißhahn, G., Hacker, W., Pache, M. and Lindemann, U. Effort-saving product representations in design: Results of a questionnaire study. *Design Studies*, 22. 472-491.
25. Rubinstein, S. *Eine Studie zum Problem der Methode*, University of Marburg, 1914.
26. Sachse, P., Leinert, S., Sundin, M. and Hacker, W. Funktionen des Prototyping im Konstruktionsprozess. *Zeitschrift für Arbeitswissenschaft*, 53 (4). 225-236.
27. Schmidt, K. The critical role of workplace studies in CSCW. in Luff, P., Hindmarsh, J. and Heath, C. eds. *Workplace Studies. Recovering Work Practice and Information System Design*, Cambridge University Press, Cambridge, 2000, 141-149.
28. Verbeck, A. and Lauche, K., Analysis and Classification Methodology for Objectifications in Collective Design Processes. in *International Conference on Engineering Design 2001*, (Glasgow, 2001).
29. Volpert, W. Psychische Regulation von Arbeitstätigkeiten. in Kleinbeck, U. and Rutenfranz, J. eds. *Enzyklopädie der Psychologie, Band D III1: Arbeitspsychologie*, Hogrefe, Göttingen, 1987, 1- 42.
30. Warr, P. The study of well-being, behaviour and attitudes. in Warr, P. ed. *Psychology at work*, Penguin, London, 2002, 1-25.
31. Weber, W.G. *Analyse von Gruppenarbeit. Kollektive Handlungsregulation in soziotechnischen Systemen*. Huber, Bern, 1997.
32. Weber, W.G., Lauche, K. and Verbeck, A. Gemeinsame Vergegenständlichungen und kooperatives Handeln in multifunktionalen Teams. in Sachse, P. and Specker, A. eds. *Design Thinking*, vdf, Zürich, 1999, 220 - 251.

1. How often do you meet with your project team?	<input type="checkbox"/> Daily				
	<input type="checkbox"/> Once a week				
	<input type="checkbox"/> Once a fortnight				
	<input type="checkbox"/> Once a month				
	<input type="checkbox"/> Less than that				
2. What is the purpose of these meetings?	<input type="checkbox"/> Mutual information				
	<input type="checkbox"/> Co-ordination				
	<input type="checkbox"/> Co-operation on shared task				
3. Where are people located?	<input type="checkbox"/> In the same room				
	<input type="checkbox"/> In the same building				
	<input type="checkbox"/> Within one hour drive				
	<input type="checkbox"/> Further away but within the UK				
	<input type="checkbox"/> In another country				
4. Project reviews and discussions are scheduled	<input type="checkbox"/> at predefined stages of the project				
	<input type="checkbox"/> Regularly every .... week				
	<input type="checkbox"/> As needed				
	<input type="checkbox"/> Not at all				
5. Within the project we usually carry out together	<input type="checkbox"/> Tendering				
	<input type="checkbox"/> Conceptual stage and task break down				
	<input type="checkbox"/> Risk analysis				
	<input type="checkbox"/> Detailed engineering work				
	<input type="checkbox"/> Documentation				
Please tick the most appropriate box (one only).	The Managing Director	Project Manager	Assistant Project Manager	The relevant project member(s)	All members of the team
6. The project is officially represented by:					
7. The co-ordination of the project is carried out by:					
8. The detailed planning within the team is undertaken by					
9. Interaction with the client are done mostly by:					
10. The responsibility for time, quality and costs lies with:					
11. Staffing decisions within the project are made by:					

# The Activity Theoretical Iterative Evaluation Method

**Amanda Quek**

Faculty of Computing, Engineering and Technology  
Staffordshire University  
Beaconside, Stafford, ST18 0AD, United Kingdom  
a.quek@staffs.ac.uk

**Hanifa Shah**

Faculty of Computing, Engineering and Technology  
Staffordshire University  
Beaconside, Stafford, ST18 0AD, United Kingdom  
h.shah@staffs.ac.uk

## ABSTRACT

This paper describes the Activity Theoretical Iterative Evaluation Method (ATTEM). ATTEM is based on Activity Theory (AT), and aims to concretise the abstract concepts of AT into a practical method for evaluation that can be applied by IS practitioners without requiring in-depth knowledge of the theory. We describe how the method was developed, its links to AT, and the application of ATTEM in a case study. We then discuss our reflections on the method, which are framed as contradictions. Additional issues and recommendations also emerged from the development and application of ATTEM, and these are addressed in the subsequent section of this paper, followed by some conclusions.

## Keywords

Activity theory, evaluation, information systems

## INTRODUCTION

The need for incorporating socio-cultural and contextual issues into IS development has been well recognised. Activity Theory (AT) has been used successfully for this purpose, but it remains difficult for the non-activity theorist to directly apply the abstract concepts of AT. There is a need for practical AT-based methods that can be used without needing much theoretical knowledge. To this end, the Activity Theoretical Iterative Evaluation Method (ATTEM) was developed. This paper presents an academic reflection on the method, and how it was developed from AT. A practical case study is also described, within which the method was applied. Finally, the contradictions that arose during the development of the method are discussed, together with ideas for future work and refinement.

## THE ACTIVITY THEORETICAL ITERATIVE EVALUATION METHOD

ATTEM is based around a framework with 6 main components. The method is founded on extensive academic, theoretical research into AT, within the context of Tracker, a research-oriented software development project. The need for a structured AT-based method for evaluation was realised when we attempted to apply AT to evaluate the Tracker prototype system. ATTEM was developed, and applied to address this need. The method is semi-structured, because it needs to be flexible enough to evolve and be

changed by its user to fit the situation to which it is applied. ATTEM is iterative in nature, and practitioners will find themselves returning to previous results of components to provide additional information. There is no strict 'border' between each component. This is similar to the UML design process, where work on one model may effect changes or refinements to other models.

ATTEM can also be used as a tool for supporting collaborative work between developers as well as for facilitating collaboration between developers and users.

## DEVELOPING ATTEM FROM ACTIVITY THEORY

From the theoretical body of AT, several concepts were identified that we believe are of particular significance and importance to evaluation, and that can help to provide and identify information that is of interest to evaluation. These concepts were used to construct the evaluation components. Each component has a unique purpose, and some components are related, which means they can be carried out concurrently. There are 6 components in the method, in a generally descending order of priority.

The criteria used in the method are based directly on AT, although translated into everyday language. These criteria are not fully comprehensive, and represent only what we have identified as being the key concepts of AT for the purposes of evaluation. These high-level criteria determine the degree to which the tool:

- Is activity focused:
  - Supports the way work is done currently.
  - Enhances / improves current work, because the purpose of any information system is not just to automate, it is to improve.
  - Addresses the original problems faced by the users that led to the need for the tool.
- Supports extended activities (those that are not directly connected with the use of the tool):
  - Supports activities that provide input to the tool or require output from the tool.
  - Supports (if necessary) the surrounding activity systems.

- Supports collective activity:
  - According to AT, all activity is collective. Therefore, the system should support collaborative work between users, both horizontally and vertically as well as with other people with seemingly no connection to the activity.
  - Supports communication between users.
- Supports learning:
  - Supports users in learning to use the system, if it is difficult to learn to use it, it will hinder work.
  - Supports not just initial learning, but ongoing learning – eg the tool needs to support and encourage the smooth transition between stages of proficiency - phases of novice and expert user.
- Supports change and development:
  - Is constructed with an awareness of, and support for not only how work is done currently, but how work has evolved – eg do the developers understand the reasons for current ‘workarounds’, and their implications for the new tool?
  - Allows continuous maintenance and change to support the evolution of work practice over time - are users allowed to create their own settings, save preferences, even modify the way the program works?

Other AT-based evaluation methods exist which uses different categories and criteria, such as the Activity Checklist (Kaptelinin *et al.*, 1999), but upon examination, these methods are more theoretical than practical (see previous work on a comparative survey of AT-based methods for IS development (Quek and Shah, 2004)).

#### THE CASE STUDY

The ATIEM method was developed, applied, and refined within the context of Tracker, an ongoing collaborative research project between Staffordshire University and the University of Lancaster in the United Kingdom, and funded by a grant from EPSRC, the national research council for engineering and physical sciences. The objective of Tracker is to develop a prototype software tool for tracking the decisions and actions made in meetings in order to reduce unnecessary rework. There were 5 members in the team, including the project leader. The author’s role in the project was effectively as an analyst, to use AT to structure user contact, interface design, and evaluation. The ATIEM method was developed and used over a period of 6 months, throughout the second half of the Tracker project.

In the first phase of the evaluation, components 1 to 3 of the method were used to guide interviews with users. Actor diagrams, use cases, and use scenarios were produced, together with a description of the problem situation(s) within which the need for the tool arose. These were used to construct the evaluation criteria. The results also helped the developers and the other members of the project team to get to grips with what the tool was supposed to do.

In the second phase, the criteria formed from Components 3 to 6 were used to evaluate the tool. The output of this first iteration of the ATIEM method, in the form of the evaluation templates<sup>1</sup> containing the evaluation criteria and results, as well as the bug lists, were given to the developers, who then made the necessary changes to the tool. The developers then provided subsequent new versions of the tool to the researcher, for re-evaluation. For each new version, templates 5 to 8 were updated with the progress of the tool. This cycle was carried out three times, and the evolution of the tool was tracked. Throughout this process, we were able to observe how the tool evolved over time. Working in tandem with the developers enabled us to feed the results of the evaluation, and any requirements that emerged, into the next coding cycle. The developers sometimes needed clarification on the bugs and evaluation result tables, so face to face discussion took place. The developers also found that the bug lists valuable as the easiest presentation that clearly showed the required course of action.

#### ABOUT THE ATIEM METHOD

ATIEM is based on activity theory (AT), which is centred on the human and their activity. ATIEM is a sound, theoretically based guide to evaluating the degree to which the system supports users who are carrying out purposeful activity. This results in a shift in focus – from the technical aspects of development to the human aspects - does the system actually support the work that the users need to do?

ATIEM is suitable for the evaluation of generic multi-user information systems within an organisation, such as office automation systems, or intranet applications. We are looking at developing extensions, or add-on components for the method, to provide support for specialised domains such as web-based systems, and e-commerce.

ATIEM is for use by IS developers or systems analysts, with the cooperation of users. ATIEM is ideal for the in-house development of support systems that aim to automate and improve current work, or the re-designing of existing software systems that are problematic. This method can also be used in organisations that are concerned with producing usable systems, and who wish to take steps towards participatory design. User involvement is essential, as the success of the method depends upon obtaining data from the real users. We cannot stress enough the importance of high and continuous involvement with the end users.

There are two types of outcome from an application of ATIEM- abstract and concrete.

---

<sup>1</sup> The complete ATIEM templates can be found at <http://soc.staffs.ac.uk/aq2>.

#### Abstract – Concept-based

- Prioritising the work to be done.
- To see to what extent the ‘essential’ functionality is supported by the system. ‘Essential’ would be support for Components 1 – 3 (see Figure 1), unless it is a specific type of system, eg CSCW, or CSCL.
- To see other aspects of human activity that should / could be supported by the system. Whether these aspects are implemented or not of course will depend on the timescales of the project.
- As a capability maturity model (or a ‘human activity’ maturity model), to show, at any one point in time, what is the next step in development to take, from the point of view of supporting human activity.
- Awareness of human and social aspects that contribute to achieving a design that is more suited to the particular needs and work practice of the real users within the context of their organisation.
- A different perspective – with AT, the object under scrutiny is the human and their activity. This provides a shift in focus – from the technical aspects of development (eg programming) to the human aspects (eg does the system actually support the work that the users are doing/want to do).
- As a tool for supporting collaborative work – between developers as well as between developers and users.

#### Concrete – Paper-based

- Functional requirements are produced.
- A running report of the current state of the system, which tracks the changes made as a result of the evaluation over time. The report contains tables denoting functions that are supported and not supported.
- Usability bug report.
- Functionality bug report.

ATIEM supports mainly the evaluation phase, but is also strongly recommended for use with requirements elicitation, analysis, and design.

To apply ATIEM, practitioners need only a minimal knowledge of AT, although some working knowledge will be appropriate to help the developers to select the most suitable components of the framework for use in their particular situation.

A major assumption of ATIEM is that the user of the method already has the skills and training needed to communicate with and elicit information from the users. Qualitative techniques borrowed from IS can be used, such as ethnography, or Contextual Inquiry (Holtzblatt and Jones, 1993) which provides practical advice for practitioners to determine how to structure customer visits and interviews. For a full review of such techniques, see Muller *et al.* (1997).

The practitioners of ATIEM also need to possess the skills to identify and decompose use cases correctly. In order to carry out this evaluation, a thorough knowledge of the real user and the user’s context is necessary. This can be achieved through the use of a variety of techniques, such as interviews and observation, from qualitative fields such as ethnography, contextual design, or participatory design.

#### The ATIEM method

##### *Component 1: Identify the Main Activity*

Purpose: To analyse and understand the current real world context of the system.

To identify the main activity, ask your users:

- What work is the system automating/improving?
- What is the purpose of the system?

Complete Template 1 by providing answers to the guiding questions (you can find the complete ATIEM templates on the Web, at <http://www.soc.staffs.ac.uk/aq2>.) Practitioners can also refer to the 8-step model in Mwanza (2001), or sections I and II in Korpela *et al.* (2000) for other guidelines that have been produced on how to fill in each field of Template 1.

A graphical representation of the main activity can be drawn to aid communication with users. Fill in Template 2 with the results that you obtained above. Modify the labels of the arrows to fit the particular situation under analysis.

##### *Component 2: Solving the Right Problem*

Purpose: To evaluate awareness of and support for the problems the system was meant to solve originally.

*2.1: Identify problems in the current activity, which lead to the purpose of developing the system.*

Ask your users the following questions:

- What problems are the users facing in carrying out their work or tasks?
- What is stopping them from doing their job smoothly?
- Is there a feature of the (existing) system that they would like to see changed?
- Is there anything that is constraining the user’s work unnecessarily?
- Are there any problems/issues occurring between how the activity is carried out currently, and an envisaged ‘better’ way of doing it?
- Are there any work practices or parts of the system that conflict with each other?
- Is there ever any confusion when carrying out everyday work processes? What are the confusing situations about?
- Are there any problems between the main activity as a whole, and other activities that are taking place within the organisation?

If you are developing an IS to improve or replace an existing IS, pay close attention to problems related to the existing system.

What are problems?

Problems can also be seen as dilemmas, conflicting objectives, tensions, or difficulties faced by the users (or potential users) in their current working situation.

### *Component 3: Constructing the Evaluation Criteria*

Purpose: Based on the knowledge learned from the previous components, construct the evaluation criteria.

In order to do this, first we need to identify the immediate and extended actors. (the term ‘actors’ refers to the different user types and user roles that would use the system).

#### *3.1: Identify the immediate and extended actors / stakeholders*

Purpose: Evaluate support for the different actors who use the system, both immediate and extended. Supports (if necessary) the other activities in the organisation’s value chain

##### **To identify immediate actors:**

Immediate actors refer to all possible actors who will use the IS directly. These include the end users, and the members of their community.

##### **To identify extended actors / stakeholders:**

Extended actors are those who belong to the activities surrounding the main activity, along the organisation’s value chain. Complete Template 3, filling in all the possible extended actors / stakeholders of the system.

#### *3.2: For each actor, identify and model their actions.*

Purpose: To evaluate coverage of and support for each actor’s actions.

Actions refer to the specific tasks that would be carried out by each actor through using the system. By collaborating with both the main and extended users, identify the different types of work that the users currently do. Include also the improved ways of working that the users envisage with the new system, and the ways in which the new system is supposed to solve the problems faced by the users. Close collaboration with users is essential to determine the full range of actions that they perform. Use an approach such as Contextual Design (Beyer and Holtzblatt, 1998) to understand the users’ current working practice.

Actions can be thought of as ‘use cases’, as previously suggested by Korpela *et al.*(2000). Use the guidelines provided by UML (Fowler and Scott, 1999) to identify and model actions in the form of use cases. Using Template 4, draw a separate use case model for each actor.

#### *3.3: Decompose these actions into operations*

Purpose: To evaluate support for the **actions** and **operations** that are taken by the user in order to achieve their **goals**.

This can be carried out with the help of ATIEM templates 5 and 6 for each actor. Complete Template 5, listing the actions (use cases) that were identified. If necessary, instantiate high-level actions into several typical scenarios of use.

Following that, complete Template 6 by breaking down each of the use scenarios into operations. The completed templates containing actions and their corresponding operations are the **evaluation criteria**.

This is when the real evaluation begins – all that we have done so far is to develop the right criteria. Work through the actions and operations listed on the templates. If the action or operation is supported by the system, record a ‘Yes’ or ‘Partial’ in the second column (Supported (Y/N/Partial)). If the action or operation is partially supported or not supported, a priority is assigned, and entered in the table together with the date that the evaluation took place. Once an action has been implemented fully, the date completed and the version number of the software is recorded in the final column.

The typical use scenarios, and the exact sequence or series of operations, should be elicited through contact with users, or observation of the users carrying out the task. In addition to supporting the users’ main activity, Components 4 – 6 focus on specific areas for evaluation, namely, support for change and development, support for collaborative work, and support for learning. These components are described below.

### *Component 4: Support for Future Change and Development*

Purpose: To identify how the activity might change in future and to evaluate support for change.

Questions that can be used to elicit scenarios for future change and development:

- What functionality might the users wish to change in the system?
- How might the users be able to change the way the system works?
- What functionality might the users wish to change in future?
- How might the users be able to report the need for changes in the system?
- Does the system need to adapt over time?

In Components 4 – 6, the envisaged work scenarios elicited by these sample questions can be treated as actions, and decomposed as per Component 3. Templates 5 and 6 can be used in the same way as in Component 3.

Some example actions for this section might be:

- a) Changing settings in the system.
- b) Using user-defined 'styles' in work.
- c) Contacting developers to notify problems / errors.
- d) Adding/deleting users of the system.
- e) Changing the way that the system works – adding new functionality.
- f) Upgrading the system / installing updates, patches, new version.

*Component 5: Support for Learning*

Purpose: To identify how the system could best support the users in learning to use the system. To evaluate support not just for initial learning, but ongoing learning – the system needs to support and encourage the smooth transition between stages of proficiency - phases of novice and expert user

Questions that can be used to elicit learning scenarios:

- How might the users wish to learn to use the system?
- How can the system help to speed up the learning process?
- How might the users wish to shorten the steps that need to be taken to reach a goal?

Some example actions for this section might be:

- Obtain help on a particular problem/aspect of using the system.
- Obtain help on learning to use the system from scratch.
- Access other help resources – eg people/ documentation/online help.

*Component 6: Support for Collective Activity*

Purpose: To evaluate support for collaborative work between users, horizontally and vertically as well as with other people with seemingly no connection to the activity.

Questions to elicit cooperative scenarios:

- How might users wish to share information through the system?
- What information do the users share currently (without the system) and how do they do so?
- How might the users want to work together through the system?

Some example actions for this section might be:

- Share information/work with other users through the system.
- Enable users who are using the system at the same time to remain aware of what each other is doing.
- Enable users to communicate with each other through the system: eg instant messaging, bulletin boards, chat, email.

*Recording Usability and Functionality Bugs*

Templates 7 and 8 provide a way in which to record bugs that are found during the evaluation. You can refer to software development literature on the best way to record bugs. It is up to you whether you wish to track usability bugs in the same database as functionality bugs. There are different viewpoints in software about whether these bugs should be tracked together or separately (see Wilson and Coyne (2001) for a discussion on this, as well as ways in which to ensure that usability bugs get an equal priority to functionality bugs).

**Iteration of the method**

Following any changes made to the system, you can repeat the steps from Component 3 onwards, by modifying the status of actions/operations, and updating the table to show when they were supported and in what version. In this way, you can observe how the system evolves as a result of the evaluation.

Component 2 can also be iterated to determine the new conflicts or problems that emerge through the use of the system over time.

**ILLUSTRATIVE EXAMPLE**

One of the outcomes of Tracker was to develop a prototype software tool for tracking the decisions made in meetings in order to reduce unnecessary rework. This example documents a summarised result of the first iteration of ATIEM applied to the Tracker project. The results detailed here are from the very first iteration of evaluation on the project.

*Component 1: Identify the Main Activity*

For the Tracker project, the main activity is **recording decisions in meetings**. Table 1 shows the results of analysing the main activity, and Figure 1 shows these results in a visual model.

Name of Element	Main Activity
Main Users	Meeting Participant/Secretary
Objective	Recording actions explicitly, recording decisions implicitly usually in minutes
Tools	Paper and writing tool, memory
Rules	The minute taking practices enforced by the organisation, meeting minute format, how to determine the people to be invited, or the agenda to be discussed
Community	Manager, other participants, other team members
Division of Work	Secretary keeps written record, other participants rely on own writing/memory

Table 1: **Elements of the Main Activity**

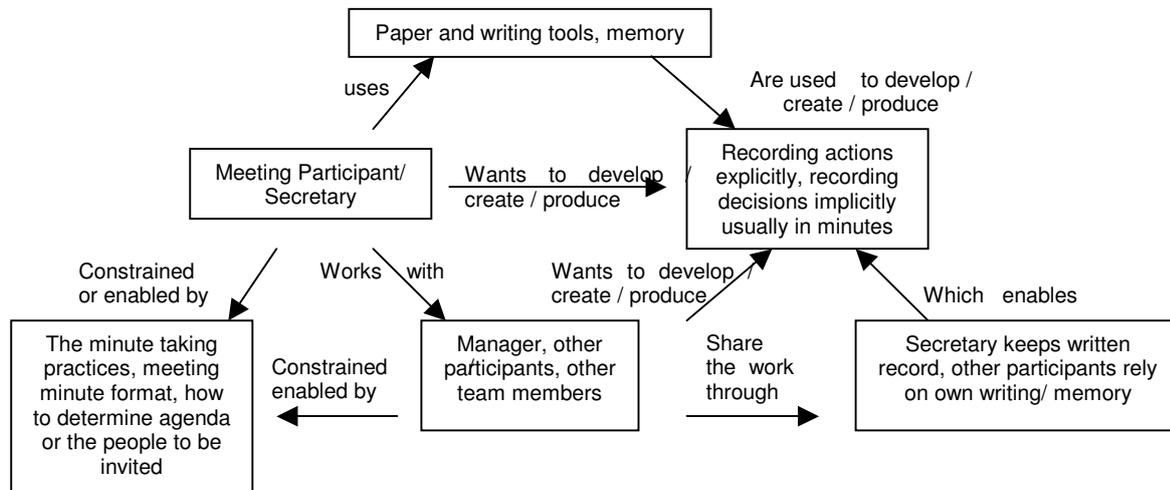


Figure 1: **Modelling the Main Activity**

*Component 2: Solving the Right Problem*

Following analysis of the users' work and situation, the current activity faces the main problem of decisions being reworked in meetings. Rework is caused by various other problems such as:

- a) Forgetting what decisions have been made and why.
- b) Decisions not formally documented.
- c) Absence of key personnel at meetings, which slows things down.
- d) Late distribution of meeting minutes.
- e) Not realising that a decision made today impacts upon a related one made in the past.
- f) Changes in requirements, needs, circumstances, that force a rehash of old decisions.

Problems a) to d) are related to the need for a memory aid that enables quick recovery of decisions and meeting information.

Problem e) relates to the need to view links between decisions as they are made, and

we note that in order to address problems a) to d), the current activity itself will need to be improved. Therefore, the system not only should support current work (which is minute or note-taking), but also needs to support:

- Search and retrieval of 'decisions' during and outside of meetings.
- Ability to create 'links' between 'decisions'.

We decided that the best way to address these problems is to create an application with a 'decision database', which could be queried. This would result in a change in working practice, by having one of the meeting participants carrying out the role of 'Search Assistant' during meetings.

*Component 3: Constructing the Evaluation Criteria*

*3.1 Identify the immediate and extended actors / stakeholders*

Immediate actors: Secretary, Search Assistant, Meeting Participant

General name of the extended activities	Extended Actors
User-training activity	New team member, Developer
Tool-making activity	Transcriber, Administrator
Management activity	Manager, Accounts and finance, Consultant, Director of the Organisation
Use Activity	Department or Organisation as a whole, external organisations that benefit from what the main activity produces

Table 2: **Identifying extended actors / stakeholders**

*3.2 For each actor, identify and model their actions.*

Due to time constraints in Tracker, and also that the system is just a prototype, the following actors were selected as essential:

- Search Assistant
- Secretary
- Meeting Manager

The full use case diagrams for each actor are available at <http://soc.staffs.ac.uk/aq2>.

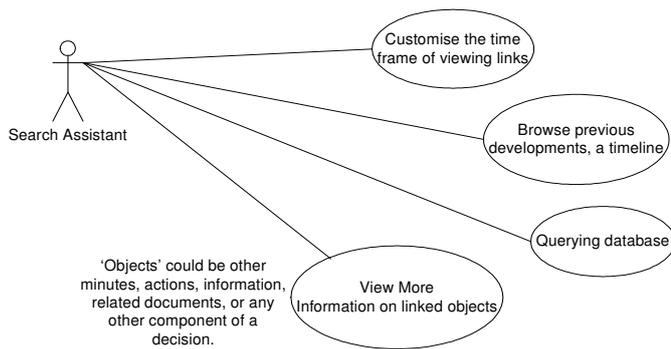


Figure 2: Search Assistant's Use Case Model

### 3.3 Decompose these actions into operations

Following consultation with the users, the Search Assistant's main use case, 'querying the database', was instantiated into several common examples of querying scenarios. Tables 3 and 4 show a selection of results from this step, due to space constraints we are unable to show the entire set of results.

The selected results for Steps 4 till 6 are shown in the same table format as Step 3.3, see tables 5 till 9. Additionally, for examples of the bug lists that were produced, see tables 10 and 11.

Date evaluated	Actions	Supported (Y/N/Partial)	Priority (H/M/L)	Completed in version / on date
	Querying the Database Common Query Scenarios: a) What did A say last Tuesday about <search string>? b) What actions was B given and has he followed up on them?	Partial – single parameter search only Partial – able to search for B's name, but the status of the actions are unknown	H H	

Table 3: Evaluating and Tracking Support for Use Cases, Actions – selected results shown

Date evaluated	Operations	Supported (Y/N/Partial)	Priority (H/M/L)	Completed in version / on date
	Query Scenario 1: What did A say last Tuesday about <search string>? a) Start the search b) Look for place to enter A's name c) Obtain results, conduct a subsearch for <search string> d) Sort the results by date e) Look for Tuesday's date f) Display a particular record	Y Y N N N N – all records are displayed directly in full	H H M M	1.0; 13/12/03 1.0; 13/12/03

Table 4: Evaluating and Tracking Support for Use Cases, Operations – selected results shown

#### Component 4: Support for Future Change and Development

Date evaluated	Example Actions	Supported (Y/N/Partial)	Priority (H/M/L)	Completed in version / on date
	a) Changing notification timespans for actions b) Changing 'keywords' used in the system c) Contacting developers to notify of a bug d) Adding/deleting users of the system e) Changing the way that the system works – adding new functionality	N N/Partial – users are unable to modify keywords, but can contact developers to do so N Y N	M L M M L	1.0; 13/12/03

Table 5: Support for Future Change and Development, Actions – selected results shown

*Component 5: Support for Learning*

Date evaluated	Example Actions	Supported (Y/N/Partial)	Priority (H/M/L)	Completed in version / on date
	a) Locate specific help regarding the use of the system b) Learn to use the system from scratch c) User should be able to access contextual help	No– learning support is not provided	H  M  L	

Table 6: Support for Learning, Actions – selected results shown

Date evaluated	Operations	Supported (Y/N/Partial)	Priority (H/M/L)	Completed in version / on date
	Locate specific help regarding the use of the system a) Open help b) Type in question or keywords c) View result d) Follow the instructions if given e) Close help f) The user may wish to view previous help records again with a shortcut.	No– help is not provided	H  H H M H L	

Table 7: Support for Learning, Operations – selected results shown

*Component 6: Support for Collaborative Work*

Date evaluated	Example Actions	Supported (Y/N/Partial)	Priority (H/M/L)	Completed in version / on date
	a) Send a specific record obtained from the system to another user b) Circulate minutes to other users c) Enable users to communicate with each other through the system: eg instant messaging	No–collaborative work is not supported.	M  H  L	

Table 8: Support for Collaborative Work, Actions – selected results shown

Date evaluated	Operations	Supported (Y/N/Partial)	Priority (H/M/L)	Completed in version / on date
	Send a specific record obtained from the system to another user a) Indicate/select/display recordset/record that is to be shared b) Go to ‘share information’ c) Input name/contact details of user to share with, or input email address. d) Input a supplementary note about the record(s) e) Send (via one or more channels, email, print, link, save as file, drag and drop, copy and paste etc) to another user.	Not supported	M  M M H  H  H	

Table 9: Support for Collaborative Work, Operations – selected results shown

## Examples of the bug lists produced

### Functionality Bugs

Date	Part of application	Description of bug	Priority	Action taken
	Search results	Errors in subsearching, subsearches produce results from transcripts and not from within initial search.	High	
	Search results	Search text /keywords have case-sensitive problems.	Medium	

Table 10: **Functionality Bugs** – selected results shown

### Usability Bugs

Date	Part of application	Description of bug	Priority	Action taken
	Help menu option on main window	Shortcut key F1 is not shown, without using this there is no way to access contextual help.	High	
	Browse	Unclear that double clicking on record number is a function.	High	

Table 11: **Usability Bugs** – selected results shown

## REFLECTIONS ON APPLYING ATIEM

It is appropriate to frame our reflections on the method as a series of tensions, or contradictions.

### Method vs. Framework

Although ATIEM is called an evaluation method, in reality, it is a framework, which is made up of components that can be carried out in relative isolation. Although there are certain ‘steps’ to the method, it is not compulsory to carry them out in strict order. The IS developer or analyst is free to choose the components that they feel most appropriate to apply at a specific point in time.

### AT-based vs. IS-practicable

ATIEM evolved from a version that used strictly AT-based terms and concepts, to one that eliminated as much jargon as possible. There was an inherent tension that emerged in developing a method that is sufficiently abstract, and grounded in theory, yet practicable and applicable. In these early stages of making AT practical, a clear link to the theory needs to be shown to strengthen the method academically. However, IS developers might not use a method that was overly theoretical, because it would result in a steep learning curve to learn the theory. Therefore, as far as possible, we have constructed ATIEM using contemporary, IS development language.

### Iterative vs. Waterfall

When the tool was demonstrated to the users after 3 cycles of implementation and evaluation, the users came up with several potential uses of the tool that had not been thought of before, which in turn opened up various possibilities for how to take the tool forward. This makes a strong case for an iterative evaluation method such as ATIEM, because “a tool always implies more possible uses than the original operations that gave birth to it” (Engeström, 1990). Therefore, an iterative or evolutionary style is more capable of handling changes than a waterfall-type development cycle. Software development has a parallel to this in the

prototyping and evolutionary development life cycle, which alternates between (re-)design and (re-)evaluation.

### Analysis vs. Evaluation

Although ATIEM is designed specifically for evaluation, the first half of the method (Components 1, 2, and part of Component 3) is dedicated to analysis, to understand the user’s work, context, needs, and problems. These factors must be elicited and understood because they are the needs and constraints that the tool must address in order for it to be successful and accepted. It is based on the results of this analysis, that the evaluation criteria are created. Bannon (1994) describes the process of use, design, and evaluation as melding into each other in practice. ATIEM supports this idea, and thus includes analysis as an essential part of evaluation.

### Implicit vs. Explicit Criteria Construction

Early AT-based evaluation methods such as the Activity Checklist (Kaptelinin *et al.*, 1999) consisted of abstract, general pointers, which practitioners were encouraged to instantiate to fit their situation. It was not described how the actual evaluation criteria were to be constructed or used. The idea that the construction of evaluation criteria should in itself be a part of the evaluation was looked at briefly by Earle and Stevenson (2000). We support this suggestion, and develop the idea further in ATIEM by dedicating the entirety of Component 3 to explicitly **constructing** the evaluation criteria. We argue that the explicit creation of evaluation criteria is an essential part of evaluation, because the activities and circumstances that need to be supported will be different for each system and in each context of use.

### Criteria vs. Means of Developing Criteria

Closely related to the above point, ATIEM also advances this work further by recognising the distinction between **criteria** and **the means of developing criteria**. Because the creation of criteria is an essential step in evaluation, ATIEM does not in itself provide criteria, as IS evaluation

methods tend to do; instead it provides **tools** and **guidelines** with which to **construct** criteria. It is crucial to understand this distinction, so as to avoid any confusion between the two.

#### **ADDITIONAL ISSUES AND RECOMMENDATIONS**

This section details additional issues that emerged from the process of developing ATIEM, describes several strengths and weaknesses of the method, and discusses the realistic applicability of the method.

#### **AT – A ‘User Interface’ Theory?**

In the field of IS development, AT is sometimes described as a ‘user interface’ theory. However, in the course of carrying out ATIEM, it was found that a large proportion of the insights gained had to do with the functionality (or lack thereof) of the tool. The focus of an AT based evaluation, and the focus of ATIEM, is to find out the extent to which the tool **supports human activity**. We found that an AT-based evaluation is equally strong at bringing out flaws/weaknesses in the **functionality** of the tool. AT is therefore not just a ‘user interface’ theory.

#### **Collaboration and Communication Issues**

It is important to maintain close collaboration with users. The effectiveness of ATIEM (or any other participatory approach) depends upon obtaining real use cases and use scenarios that are closest to the actual work that the system is automating/improving. It is also important to work together with the programmers from the start, as they may be reluctant to make changes in the system at a late stage. Ongoing support from management is crucial; they should be convinced that the work is worthwhile, and they need to consent to early contact with users. Some literature that may help IS practitioners to convince key people (managers, developers) of the importance of involving users and understanding work practice, etc, are Wynn (1991), Bloomer and Croft (1997), and Karasti (2001).

#### **Cost-justifying AT in commercial IS development**

AT research may need to take a leaf out of other fields, by doing work on how to cost-justify AT analyses, and how AT-based methods can quantitatively benefit an organisation eg Blythin *et al.*(1997) on ethnography, and Kujala (2003) on the benefits of user involvement in design. In order to further support strong and enthusiastic industrial collaboration, we need to work towards documenting categorical and empirical evidence that AT analyses bring definite benefit to IS development in industry.

#### **User-centred Design Awareness**

It was found in this project that at times the developers had similar opinions about usability and user-centred design as were described more than a decade ago by Gould and Lewis (1985). Interface issues were not considered to be as important as technical functionality of the system. In this project, developers often did not consider revisions to the interface to be significant. (Previous versions were retained

if there were significant changes in functionality, but older versions of revisions to the interface were overwritten). This is possibly due to the pressure on the developers to produce a working proof-of-concept prototype, and the developers’ lack of exposure to user-centred research and literature.

#### **Strengths and Weaknesses**

ATIEM is in no way a ‘final product’. Much work is still needed to develop AT-based methods to the standard demanded by industry. One limitation of ATIEM is that it caters towards the generic information system. If the system is for a specific purpose, e.g. learning, or collaborative work, there are particular factors that need to be taken into account, and therefore ATIEM currently would not be focused enough to use on these systems. We are looking at future work to develop ‘extensions’ or ‘add-ons’ to ATIEM, to enable it to cater for other types of IS, such as web-based systems and groupware.

#### **Realistic Applicability**

The realistic applicability of the ATIEM method has not been fully explored, as at the time of submission, it has only been used within one case study.

The success of the method relies heavily on the presence of a real user. Minimal user contact will result in a lack of data that is needed to construct the evaluation criteria.

AT itself states that the real usefulness of a system only emerges **in use, over time**. Within the constraints of some projects, it is not possible to observe the use of the system in place over time. However, ATIEM can be used to conduct an iterative evaluation, while maintaining close collaboration with one or more representatives of the real users, ideally in a ‘real’ situation.

Future work will be to improve the method, including testing it in a real organisation. We are also looking at developing extensions to the method to cater for web based systems evaluation.

#### **CONCLUSIONS**

In conclusion, we find that ATIEM can be a practical, useful, and theoretically based alternative way of evaluating information systems, which also supports iterative development. However, several issues remain to be addressed by AT research, such as the extent to which knowledge of AT is needed, and the lack of quantitative evidence to show the effectiveness of AT in IS development. Our future work aims to extend the method to suit the different needs of specific types of IS, such as web-based systems.

#### **ACKNOWLEDGMENTS**

Grateful thanks to Professor Bernadette Sharp for her valuable comments and suggestions that greatly helped to improve this work.

This research is funded by the Tracker project at Staffordshire University (EPSRC Grant No. GR/R12176/01).

## REFERENCES

1. Bannon, L. J. (1994) Use, Design and Evaluation - Steps towards an Integration. In *The Design of Computer-Supported Cooperative Work and Groupware Systems* (Eds, Shapiro, D., Tauber, M. and Traummuller, R.) Elsevier Science, Amsterdam, pp. 423-443.
2. Beyer, H. and Holtzblatt, K. (1998) *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufmann Publishers, Inc, San Francisco, California.
3. Bloomer, S. and Croft, R. (1997) Pitching Usability to Your Organisation. *interactions*, **November - December**.
4. Blythin, S., Rouncefield, M. and Hughes, J. A. (1997) Ethnography in the Commercial World. *Interactions*, **May-June**.
5. Earle, A. and Stevenson, I. (2000) Evaluation Framework: The CANDLE Project- Collaborative And Network Distributed Learning Environment. [http://www.candle.eu.org/pdfs/deliverable\\_6-1.pdf](http://www.candle.eu.org/pdfs/deliverable_6-1.pdf). Accessed on 27/04/04.
6. Engeström, Y. (1990) When Is A Tool? Multiple Meanings of Artifacts in Human Activity. In *Learning Working and Imagining* (Ed, Engeström, Y.) Orienta-Konsultit Oy, Helsinki.
7. Fowler, M. and Scott, K. (1999) *UML Distilled: A Brief Guide to the Standard Object Modelling Language*, Addison Wesley.
8. Gould, J. D. and Lewis, C. (1985) Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM*, **28**.
9. Holtzblatt, K. and Jones, S. (1993) Contextual Inquiry: A Participatory Technique for System Design. In *Participatory Design: Principles and Practices* (Eds, Schuler, D. and Namioka, A.) Lawrence Erlbaum Associates, Hillsdale, New Jersey.
10. Kaptelinin, V., Nardi, B. A. and Macaulay, C. (1999) The Activity Checklist: A Tool For Representing the "Space" of Context. *Interactions*.
11. Karasti, H. (2001) Increasing Sensitivity Towards Everyday Work Practice in System Design. PhD Thesis. Department of Information Processing Science, University of Oulu.
12. Korpela, M., Soriyan, H. A. and Olufokunbi, K. C. (2000) Activity Analysis as a Method for Information Systems Development. *Scandinavian Journal of Information Systems*, **12**, 191.
13. Kujala, S. (2003) User involvement: a Review of the Benefits and Challenges. *Behaviour and Information Technology*, **22**, 1-16.
14. Muller, M. J., Haslwanter, J. H. and Dayton, T. (1997) Participatory Practices in the Software Lifecycle. In *Handbook of Human-Computer Interaction* (Eds, Helander, M., Landauer, T. K. and Prabhu, P.) Elsevier Science B V.
15. Mwanza, D. (2001) Where Theory Meets Practice: A Case for an Activity Theory based Methodology to guide Computer System Design. In *Proceedings of INTERACT 2001 - 8th IFIP TC 13 Conference on Human-Computer Interaction*, Tokyo, Japan.
16. Quek, A. and Shah, H. (2004) A Comparative Survey of Activity-based Methods for Information Systems Development. In *Proceedings of ICEIS - 6th International Conference on Enterprise Information Systems*, Porto, Portugal.
17. Wilson, C. E. and Coyne, K. P. (2001) Tracking Usability Issues: To Bug or Not to Bug? *interactions*, **May + June**.
18. Wynn, E. (1991) Taking Practice Seriously. In *Design at work : Cooperative Design of Computer Systems* (Eds, Greenbaum, J. and Kyng, M.) Lawrence Erlbaum Associates, pp. 45 - 64.

# Designing Weak and Strong Interactions: An Approach based on Cultural-Historical Psychology

**Antonio Rizzo**

University of Siena  
Department of  
Communication Science

53100 Siena - Italy

+39 0577 814369

rizzo@unisi.it

**Simone Pozzi**

University of Siena  
Department of  
Communication Science

53100 Siena – Italy

+39 0577 286833

pozzi@media.unisi.it

**Luca Save**

University of Siena  
Department of  
Communication Science

53100 Siena - Italy

+39 0577 286833

save@media.unisi.it

**Mark Sujan**

University of Siena  
Department of  
Communication Science

53100 Siena - Italy

+39 0577 286833

MarkSujan@aol.com

## ABSTRACT

This paper presents a practical description of an analysis and design methodology for complex socio-technical systems. The basis of the approach is a re-elaboration of the unit of analysis originally proposed by Vygotsky. The method focuses on man-artefact interactions in order to inform the design of new artefacts and patterns of interactions. Depending on the required level of design intervention and on the level of structure of the domain the focus is directed either towards the analysis and re-design of weak interactions, or towards the analysis of strong interactions in order to support the design of innovative artefacts and patterns of interaction. Descriptions of the individual steps of the approach, including heuristics for the practical conduct of the analysis and design, are given and illustrated with examples from a range of projects from different domains (railway, clinical, and educational domain).

## Keywords

Unit of analysis, cultural-historical psychology, activity theory, socio-technical systems, interaction design.

## INTRODUCTION

Socio-technical approaches to system analysis and design acknowledge the fact that social, technical and organizational aspects have to be understood in their mutual interaction. The particular approach followed here identifies the mediated activity as the basic unit of analysis, which links technology and people, and people and their social, organizational and cultural context. An important distinction to other approaches is the fact that the context is not regarded as a container within which human activity unfolds, but rather it is regarded as something which is actively constructed and reconstructed through human activity.

As a consequence, the analysis and design method focuses predominantly on modes of interaction and mediation.

Characteristics of the approach are:

- *Unit of analysis*: individual mediated activity
- *Applicability*: Complex socio-technical systems (from safety-critical to educational and entertainment systems)
- *Goals of the method*: studying patterns of interaction inside human activities in order to inform the design of innovative technological artefacts and patterns of interaction
- *Analysis technique*: interviews, focus groups, document analysis, ethnographic observations (based on audio and video recordings or participant observations)
- *Tools to represent activities*: workflow matrix, timeline, hierarchical task analysis, textual and visual scenarios, etc.
- *Formal outputs*: written reports, design concepts and interactive prototypes.

In the following the aims of the approach are discussed in more detail (*Aim of the Method*). Afterwards, a brief theoretical discussion motivates the basic unit of analysis upon which the approach is based (*Unit of Analysis*). The main part of the paper is dedicated to a practical step-by-step description of the approach, including descriptions of real-world examples (*Description of the Method*). Practical considerations about the applicability of the approach and about lessons learned from the different domains conclude the paper (*Conclusion*).

## AIM OF THE METHOD

Every design activity has its own history and is unique. It is almost impossible to follow the same design path twice. Design techniques and methods have to be tailored to the specific design activity a design team is involved in. What we propose in this paper is a heuristic schema, which helps to navigate between design phases and methods, inspired by the cultural-historical activity theory, and centred on an

elaborated version of the original mediation triangle proposed by Lev Vygotsky (1978). As we will discuss, the schema is the result of the combination of these theoretical concepts with a variety of design experiences we conducted in different domains. The experiences ranged from the redesign of operational systems (railway transportation, air traffic control, hospitals, etc.) to the improvement and innovation of educational systems for primary school children.

The starting points of the schema are the level of modification we would like to produce in a human activity - referred to as *levels of design* - and the *kind of interaction* between people and the mediating artefacts.

Level of design refers to the different types of change in the nature of the activity people would expect from the design outcomes. The concept of levels of design is derived from the general genetic law of cultural development (Vygotsky 1978, Wertsch 1985, Cole 1996) and the associated idea that we can only understand activities in their evolution. In order to simplify the definition of the level of design, three levels along a continuum, which could be defined as “from goals to curiosity”, have been representatively selected (Marti and Rizzo 2003). The three levels are defined as *Reactive*, *Proactive* and *Emergent*, and are expressed according to: i) what we could assume would be the perception of the potential users when starting a design process concerning their activity, ii) the mission of the designers, and iii) the main drivers for starting the design activity.

- Reactive Design:
  - i) Users: I know what I want, and I can also specify most of the conditions of satisfaction of my actions and of the results I would like to get.
  - ii) Designers: This is the case where designers are called to solve problems for a well-established human activity/task already mediated by existing and fully operational systems/tools.
  - iii) Drivers: Problems in performing the activity, competition with more effective activity mediators.
- Proactive Design:
  - i) Users: I’m interested in doing it, I cannot tell you precisely what I do except in term of actions or results, but as things evolve I will try to tell you.
  - ii) Designers: Designers are called to develop a new system/service for a well-defined human activity supporting a clear category of users.
  - iii) Drivers: exploiting new enabling technology or envisioning new business for evolutions in human activities.
- Emergent Design:
  - i) Users: I’m curious about it, but I do not know what this implies in terms of my actions, neither what I can expect as results.

- ii) Designers: Designers strive to “envision” new human activities that are designed together with the enabling technologies
- iii) Drivers: Academic and industrial research which produces visionary scenarios for human activities.

A complementary concept to the design levels is *the kind of interaction* between people and the mediating artefacts. This concept comes from our re-elaboration of the mediation triangle (see the following section). According to our vision the interaction can be characterised by a critical configuration of the mediational resources – referred to as *weak interactions* – or by a successful configuration of them – referred to as *strong interactions*.

Weak interactions are conditions of human activity, where goals can be achieved only through a restricted path of resource manipulation, i.e., situations where there are few ways of establishing a particular state of the world, and where there are few if any alternatives to that state of the world. Weak interactions can be exacerbated by unreliable or approximate resources, i.e., ways to manipulate energy that are not consistent with the aim of an intention. They are defined *weak*, because a minimum disturbance in the established flow of activity can cause relevant breakdowns and mismatches.

Strong interactions are conditions of human activity where goals can be achieved through multiple and diverse resource manipulations, i.e., situations where there are redundant and diverse ways of establishing a particular state of the world, and where there are several available alternatives to that state of the world. Strong interactions are amplified by modular and inter-operational resources, i.e., ways of acting in the world which allow an interchange of resources. They are called *strong*, as they can tolerate considerable variations in the expected flow of the activity, without causing breakdowns and mismatches and because they can support a continuous evolution of the activity itself.

At the beginning of each design process, the designer can decide whether to focus more on weak interactions or on strong interactions. The decision is related both to the design objectives s/he wants to address (what we called the *design levels*) and on the *level of structure* of the activity s/he is going to analyse.

For what concerns the design levels, the analysis of weak interactions is more suitable for the reactive level, in which the main purpose of the design activity is the reliability and efficiency of the system. On the other hand, the analysis of strong interactions is the best option for “envisioning” new creative solutions, leaving the designer free from the current technological and organisational constraints of the system under analysis (emergent level).

A general heuristic can be provided also referring to the level of structure, distinguishing between *structured* and *unstructured* activities. For highly structured systems, like railway transportation and air traffic control, which are

characterized by a well defined set of tools and procedures, the analysis of weak interactions is generally preferred. For less structured systems, such as learning activities within a school, where patterns of interaction and rules are neither fixed nor clearly established, the focus is normally on strong and successful types of interaction. Assessment of disturbances and breakdowns can be useful even in these cases, but it should not be the main concern of the design work. Otherwise the designer may concentrate on finding only *ad hoc* solutions for specific problems in a given context, which are unlikely to be observed again given the high variability of unstructured domains. Therefore, in the case of unstructured domains, the great challenge is to look for innovative solutions, which can prove valid for a large variety of conditions, sometimes drastically modifying the current situation.

Different indications can come from the combinations of the two criteria. For example, in cases where the intervention of the designer is at the emergent level, even highly structured systems can be studied relying on the analysis of strong interactions, rather than looking for disturbances and breakdowns in the activity. The emergent objective often requires overlooking current situational characteristics, as they may constrain the design space and direct towards ready-made solutions. As a consequence structured systems like railway transportation or air traffic control systems can be studied and analysed as if they were unstructured ones.

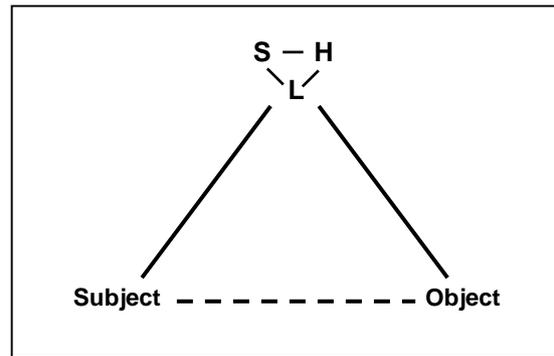
#### UNIT OF ANALYSIS

The analysis and design method we propose is grounded in Cultural Historical Activity Theory, with an explicit link to the original studies and assumptions developed by Vygotsky (1978). In accordance with the Law of Semiotic Mediation and with the General Law of Cultural Development (Vygotsky 1978, Wertsch, 1985, Cole 1996), the individual activity consisting of Subject, Object and Artefact is regarded as the basic unit of analysis. This implies a slight departure from the idea of an expanded unit of analysis, as suggested by Engestrom (1987) and other authors in Activity Theory. Instead, the original unit of analysis is adapted and modified by splitting the artefact into three different categories of mediating resources: *Hardware* (H), *Software* (S), and *Liveware* (L) (see Fig.1).<sup>1</sup>

The new elements are defined as follows:

*Hardware* is a resource that may mediate the direct transfer of energy from the subject to the external world. A hammer, for example, is considered a *hardware* artefact when it allows the user to transform the kinetic energy of the arm into the kinetic energy which drives a nail into the wall.

*Software* is a resource providing the subject with the knowledge required to use a particular artefact for her/his



objectives. It allows the subject to appropriately direct her/his energy on a material artefact and it “tells” the user the correct interaction modalities. For example, the process of learning that a hammer should be held by the terminal part of the handle, rather than by its metal part, implies the acquisition of the appropriate *software* for driving nails into the wall. Note, that humans can share or transfer *software* in time and over physical space in many ways, either in written or non-written forms<sup>2</sup>.

*Liveware* is a human resource mediating the interaction of a subject with the external world. Even though it would be unreasonable to consider people at the same level as tools, it should be acknowledged that most human activities are mediated by other humans. For instance, most of the activities accomplished by air traffic controllers are mediated by other people (i.e., *liveware* resources). Even if the controller seems to interact only with her/his radar display and to consider only the aircraft plots on it, every action is conducted in continuous coordination with pilots and other controllers.

We would like to emphasise that no sharp *a priori* distinction should be traced between the subject and the *liveware*. From a theoretical point of view any person participating in an activity can be referred to both as a *liveware* and as a subject. The only difference is the vantage point on the activity that is adopted for the analysis, and usually there is no reason for adopting a unique vantage point. We will see below (Section 4a), how vantage points can also be shifted during the same analysis to better understand issues of interpersonal coordination.

<sup>2</sup> In this sense the concept of *Software* can be considered similar to Norman’s definition of *mental model* (Norman 1988), a sort of mental image that guides the user in interacting with a specified instrument. What we would like to stress is that a software can be transferred and incorporated in different kinds of material substratum (namely hardware and liveware), mediating human activity in very different ways.

<sup>1</sup> The names of the three resources are derived from the SHEL model proposed by Edwards (1988).

This taxonomy addresses different mediational characteristics that may also appear in combination in a single artefact. Most artefacts consist of *hardware* and *software* resources, which are exploited together in the activity. In one of his numerous examples Norman explains how the typical design of scissors aids the user in understanding their correct use (Norman 1998). The different diameters of the two openings of the handle indicate to the user where to put the index finger and where to put the thumb. In this case we can say that part of the required *software* is directly integrated in the *hardware*. On the other hand, there are many cases in which *hardware* and *software* can follow different and independent evolutions. For example, a *hardware* resource, which was designed for a particular use, can be used in a completely different way, when transferred to another cultural context. In such a case the *hardware* will come in contact with a different *software* resource, perhaps completely changing the original scope of its invention and use. The reason for distinguishing between resources on an ontological basis is that one of the resources may be more apparent in the subject-artefact interaction and thus attract the designer's attention, while its role in the activity may be a limited one. For instance in Umberto Eco's famous novel "The name of the rose", there is a design case where *hardware* and *software* interaction is exploited to enforce a rule in a monastery. The librarian monk believes that the knowledge (*software*) contained in the second volume of Aristotle's *Poetica* is harmful and should be kept secret. However it is not the knowledge that kills monks that do not obey the rule. It is instead the less-apparent *hardware* part of the book, that is the poison spread on the book pages. The more monks read, the more poison they get. In this sense, the librarian perfectly designed the book for his purposes, as a naïve observer will be lead to think that the knowledge contained in Aristotele's book is deadly. The design should consider all the mediational aspects of an artefact (or a set of artefacts), and in particular the relationship of *hardware* and *software* resources, and the special nature of the human resource. The proposed taxonomy is intended to ensure that every mediational aspect can be considered for its own peculiar characteristics in the design process, since the activity is shaped by all of them.

Furthermore, *software* is often interrelated with the mediating function offered by a person (i.e., *liveware*), but in quite a different way from that of a *hardware* resource. A *hardware* resource is an artefact deliberately designed for a certain goal, which also represents the reason for its existence. The existence of a person, however, is obviously independent of and not related to a precise goal. This implies that even though people can mediate the intentions of another person and provide essential *software* for the activity (i.e., acting as "artefacts"), these people will always incorporate their own intentions and objectives, which can often be different from or even inconsistent with those of the person whose intentions they are mediating and even with the *software* they are meant to provide. This particular

nature of the *liveware* resource has to be addressed explicitly during the analysis and design process, since *liveware* mediation may easily go unnoticed because of its non-formal aspect.

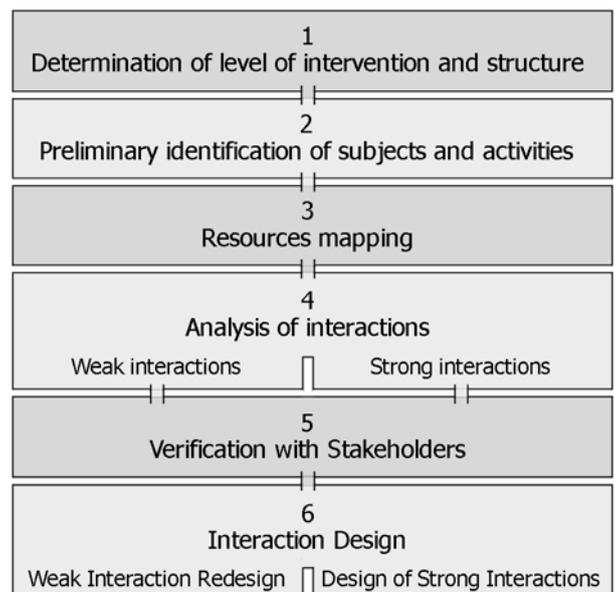
As a conclusion, in cases where the objective of the intervention is the design of new tools for a particular socio-technical system, consideration needs to be given to the nature of resources mediating particular activities (*hardware/software* or *liveware*), since their mediational properties should be preserved (or at least considered) in any new configuration.

## DESCRIPTION OF THE METHOD

*Depending on the specific type and objectives of the project, the proposed analysis and design method can be exploited in most of the phases of the design process (preferably in close coordination with other complementary approaches), i.e. from activity analysis and requirements definition, to prototype development, to the evaluation of existing systems and their redesign, and so forth.*

The phases below should ideally form a unity, in the sense that the phases of analysis are conducted with possible design recommendations in mind, and that the design is carried out based on findings of thoroughly conducted prior analysis and evaluation. Furthermore, the process is highly iterative, since new findings may prompt the designer to go back to some of the previous steps and deepen the analysis. For the sake of clarity, a linear representation is chosen here for the description of the individual steps (Fig. 2). A tabular summary of the proposed method and of individual steps is also included at the end of the paper (Tab. 1).

**1. Determination of level of design intervention and level of structure:** *determine the level of design intervention (reactive, pro-active, emergent) and the level of structure of the domain (ranging from fully structured to*



*unstructured*). This will determine the time spent on subsequent phases, the analysis and design focus (weak/strong interactions), and the tools, techniques and representations used.

Before commencing the actual analysis and design activities, the first step is to determine the required level of design intervention and the level of structure of the domain under consideration. As was discussed above (see *Aim of the Method*) the level of design intervention can be categorized as reactive, pro-active, or emergent. The level of structure of the domain can range on a continuum from fully structured to unstructured. The purpose of this classification is to determine roughly the time to be spent on individual steps of the analysis and design methodology, as well as the analysis and design focus (i.e., weak or strong interactions). In turn, this will affect the tools, techniques and representations best used. For example, a reactive intervention in a fully structured domain (e.g., re-design and usability evaluation of a train information form) usually requires more time to be spent on the evaluation of weak interactions, and less time on the analysis of activities since these are stable and well documented. On the other hand, a design intervention on the emergent level in an unstructured domain (e.g., the design of innovative technologies to support the narrative capabilities of primary school children) usually requires more time to be spent on the analysis of strong patterns of interaction, and on the design of innovative concepts. Also, the tools used for representation will be more flexible and descriptive than in the previous case (e.g., the use of descriptive activity models rather than a hierarchical task analysis representation).

The output of this phase is a classification of the design intervention in terms of level of design intervention (reactive, pro-active, emergent), level of structure of the domain (ranging from fully structured to unstructured), and a resulting subsequent focus on weak or strong interactions.

**2. Preliminary identification of subjects and activities:** *identify relevant activities of the observed system, focusing on subjects and objects.*

In this step relevant activities are selected for further analysis. The objective is to obtain a list of activities that will be analysed in order to conduct the design intervention.

The basic unit of analysis requires consideration of artefact-mediated activities, i.e., consideration of subject, artefact, and object. It is often the case that the initial specification of the design intervention only refers to macro areas of activity, hence at this stage it is necessary to deepen the analysis and identify relevant activities at a level coherent with the basic unit of analysis. This can be achieved by articulating the macro-area objectives in smaller constituent objects, which individual activities correspond to. In the same way, subjects involved in specific activity areas can be identified. Strictly speaking, the analysis requires consideration of a specific activity at a certain point in time, with a unique subject and object. However, in

particular in more structured domains it is possible to abstract and simplify the analysis. For instance, the analysis of subjects can consider roles instead of individuals. A role is constituted by a group of individuals who share a set of recurrent activities. Examples of roles in the railway domain are a *train driver*, a *signaller*, a *train manager*, etc. Examples in a hospital setting include a *nurse*, a *pharmacist*, or a *phlebotomist*. Depending on the needs of the analysis, these role definitions can be further refined to include, for example, *staff nurses*, *senior staff nurses*, and *ward sisters*, which in turn can be even further refined to include, for example, *medical staff nurses* and *haematology staff nurses*. The preliminary role definitions are not necessarily equivalent to the role definitions used later on during the analysis, but rather serve as a coarse starting point. In the same way some recurrent objects may be assumed as invariant over time and subjects.

In less structured domains such abstractions usually are not possible. While it may be possible to determine general objectives, such as the development of narrative competencies in an educational domain, the behaviour of each individual will be determined to a large extent by the formation of various differing objectives and motivations, often specific to the particular context. The analysis in such a domain needs to be based on specific activities, and on subjects as individuals.

In this phase tools can be stakeholder interviews (management, domain experts, etc) in order to gain an understanding of the “official” view of the organization, to understand the evolution of the organization, to understand global goals and their evolution, to refine the understanding gained during the initial analysis, and to establish and secure contacts and in-depth analysis possibilities for the following analysis stages. For example, a doctor and a nurse can be interviewed to give a basic understanding of the ways a hospital works, and of the different people involved. Afterwards, the management of the specific hospital under consideration can be contacted for specific information about the structure of the particular hospital, and to establish contact points for the subsequent in-depth analysis steps. The success of the subsequent analysis often depends to a large extent on the initial support by the management.

Analysis of official documents, (e.g., organigrammes) should be used to complement and verify the above information.

The output of this phase is a list of activities to be further analysed, in terms of subjects and objects. For example, consider an intervention in a hospital concerned with the analysis and improvement of patient identification processes. After having talked to, for example, a physician, a pharmacist, and a nurse, a general understanding of the basic ways a hospital is organized (e.g., departmental structure such as Accident & Emergency, Medical, Surgery, Maternity etc.; team roles in each specialty such as consultant, senior registrar etc.), and of the basic “patient

journeys” (different ways to enter a hospital, e.g., A&E self referral, GP referral etc.; transition between departments, e.g., from A&E to a ward, from a ward to X-ray etc.) and the corresponding data flow (information pertaining to the patient) can be constructed. In addition, initial scenarios for relevant roles can be constructed. These will vary in detail depending on the expertise of the interviewee. However, for the purpose of this phase it is useful to abstract these scenarios to a consistently high level, yielding, for example, lists of activities such as

- “A ward nurse is responsible for the general well-being of patients (e.g., washing, comfort, etc.)”
- “S/he administers drugs to patients twice a day on her/his drug rounds.”
- “S/he takes samples from the patient and has them delivered to the respective department.”
- “On each ward and each shift there is a designated nurse who handles the reception and the transfer of patients.”

As a result the analyst may decide to observe drug rounds, sample collection and distribution, and patient transfer in the next analysis phase. During this subsequent analysis the different activities constituting these areas are identified (i.e., during step 3).

**3. Resources mapping:** *map the relevant means/resources (tools, procedures, practices, people, roles, etc.) which mediate the activities.*

The aim of this step is the construction of an analytical representation of the activities carried out. This includes the generation of a pool of scenarios which document the various activities and subjects, tools and objects involved therein. The analysis is concerned with mediated activities, and in this step particular emphasis is given to the mapping of mediating artefacts and of the evolution of the activities.

This step typically involves a series of on-site interviews and observations, where possible including audio and video recordings for offline analysis and consideration.

The exact conduct of this step and the time required depends on a number of factors, such as the structure of the domain, the required level of intervention, previous experience of the analyst etc. The analysis can span from a couple of weeks to several months. There is no specific and static analysis schedule. Typically, the analyst tries to capture the basic mediating artefacts through observations of activities selected in the previous step, before refining the understanding in more detail to include all the available mediating artefacts (even those seldom used or less apparent), such as standards, rules, procedures, technology, tools, documents, annotations, checklists etc. The communication and interaction taking place with other people is a central part of documenting the mediating artefacts. Finally, in accordance with Vygotsky’s *genetic method* (Vygotsky 1978), the historical development of the activities should be traced where possible in order to allow

a better understanding of the activities. One approach to achieve something close to this, is the study of the same processes performed in different ways and with different levels of technological support, which simulates in a way the natural evolution of the activities.

This analysis phase is iterative in the sense that phases of observations are followed by offline structuring and modelling activities during which uncertainties will arise, which in turn have to be resolved through further analysis.

In our experience we exploited typical tools for analytic modelling of activities, such as Flow Charts, Hierarchical Task Analyses, Workflow Matrices, and Critical Scenarios, depending on the specific characteristics of the system under investigation. The proposed method does not select a privileged modality, but aims at adapting representations already used in other methods to include the particular requirements of the basic unit of analysis. For instance, it is possible to employ the hierarchical (and also graphical) representation of Hierarchical Task Analysis and annotate it in such a way as to include descriptions of the mediating artefacts. The same can be achieved more explicitly employing an extended workflow matrix with separate columns for *hardware*, *software*, and *liveware* resources.

In unstructured domains and for reactive and emergent levels of interventions rich descriptive approaches are preferable. Here, emphasis is placed on the intentions and motivations of the subjects, and on the knowledge they involve in their activities, rather than on an exact mapping of the technological resources employed. Such an approach avoids a premature constraining of the solution space, and instead keeps the attention on the higher-level objectives.

Examples of such representations are scenarios (Carroll 1995) and other *thick descriptions* as those used in ethnography (Geertz 1973), that display information about the subjects involved in the activity by including in the description basic elements as well as those contextual elements usually not present in more formal techniques. Their goal is to provide a rich and meaningful framework, where any relevant activity element may be integrated if needed for the objectives of design, rather than to specify in detail and abstract terms the activity in terms of resources, constraints, relations, etc.

Examples are given below in Fig. 3 and Fig. 4, and are discussed in the next section in the context of analysing weak and strong interactions.

**4. Analysis of interactions:** *analyze interactions between subjects and artefacts, in relation to the respective objects. Depending on the context (structured, unstructured) and on the design level (from reactive to emergent), focus either on strong or weak interactions.*

Once a pool of activity representations has been generated, specific interactions should be selected for further analysis. The aim of this step is the analysis of weak and strong interactions (depending on the type of domain and on the

level of intervention) in order to inform and to focus the subsequent design activities.

#### 4a. Analysis of Weak Interactions

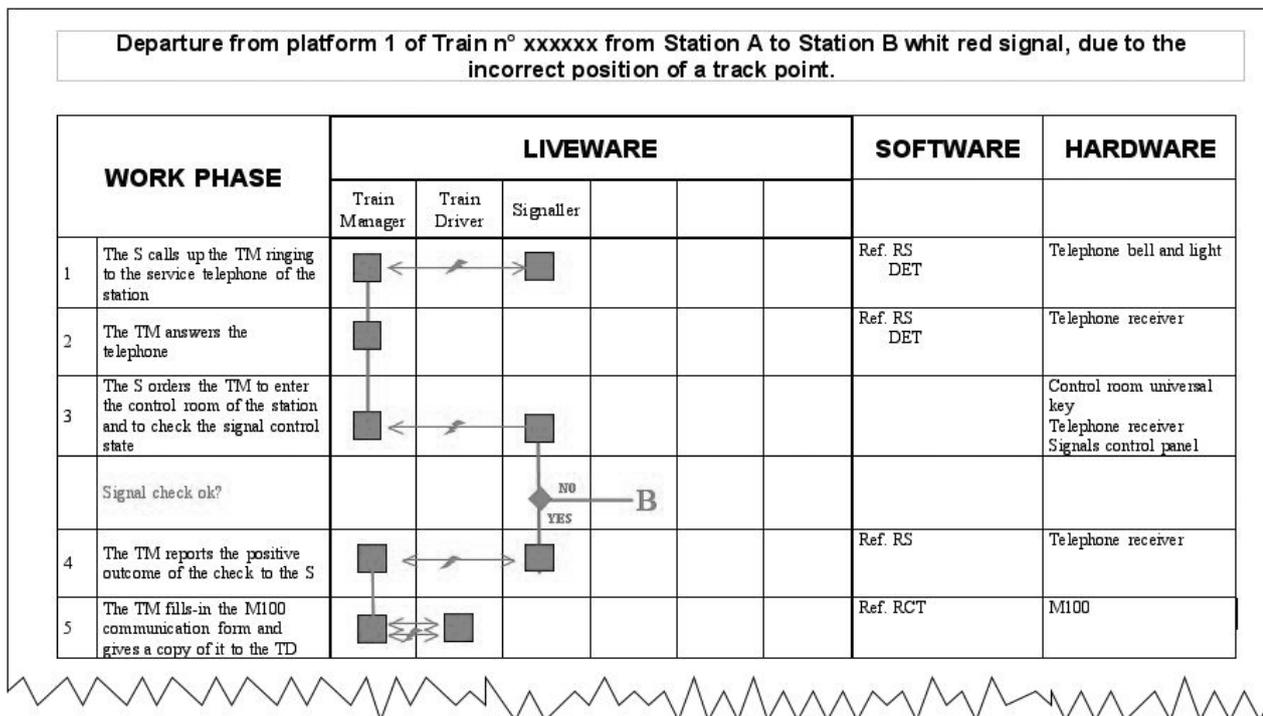
In structured domains and for reactive or proactive interventions, the analysis of weak interactions is the usual choice. Such an analysis helps to identify problems in the interaction within fairly stable activities, which result in disturbances and breakdowns. This analysis can then inform the design, which focuses on remedying the problematic interactions.

Weak interactions can be identified in a number of ways. Two basic heuristics are discussed below:

- *Mismatch between actual and expected outcomes:* a first strategy in order to identify weak interactions is the observation of differences between actual and expected outcomes of the activities. The starting point is the observation of the activities from the point of view of their *objects*. Consider this incidental scenario from a hospital setting: a nurse administers a wrong set of drugs to a particular patient, because two patients with the same surname had been placed in the same room, and the nurse cross-checked only the surname on the patient's wristband and on the drug chart. The incident is the result of a severe failure in the patient identification process and points to a range of critical issues, such as the bed shortage of the hospital (usually patients with the same surname are not placed in the same room), communication problems during shift hand-over, etc. The incorrect outcome helps to identify a set of weak interactions inside the activity: the

interaction with the patient identifiers (name & date of birth on the patient's wristband and on the drug chart), the interaction with procedures (ask the patient to state name and date of birth), and the interaction with the patient (verify with the patient the accuracy of the drugs). In safety-critical systems a possible source of information for identifying this kind of mismatches are incident reporting documents. In organizations promoting near-miss reporting schemes, in the form of anonymous or confidential documents, these could be an important starting point of critical interaction analysis. For example, incident reports in blood transfusion may evidence issues, such as confusion over patients with similar names, or checking remote from the patient's bedside.

- *Subjective assessment by participants in the activity:* a second strategy comes from observations and interviews in which participants point to problematic interactions observed from the point of view of the *subjects*. System operators, for example, can remark when there is a mismatch between the official procedure and the adaptive practices they are obliged to put in place, even if the system is apparently performing well. For example, in a critical scenario studied with the Italian railways a *train driver* complained about the need of pursuing an unofficial practice in case of a breakdown in the signalling system. The scenario concerned a train entering a small station, without any personnel employed in the management of the signals. In the past the station used to be managed by a *local signalman*, whereas now the system is automated and controlled remotely by a *signaller*, with a remote controlling system. Nevertheless, in case of a breakdown of



the remote system, the *train manager* is supposed to get off the train and access the old control room of the station, where they can fix the problem through interaction with the manual controls and by speaking on the telephone with the *signaller* in the remote control room. According to the official procedure the role in charge of this task is the *train manager*, whereas the *train driver* is supposed to remain inside the train, to prevent uncontrolled movements, failures of the braking system or other kinds of danger. However, the procedure does not take into account the evolution of the *train manager* role in the Italian railway system. For a variety of reasons, during the last years this role tended to become increasingly a commercial role, rather than a technical one as it was in the past. The recently trained *train managers* are competent in managing the relationship with the railway customers, in checking and selling tickets, in ensuring a smooth and comfortable accommodation or answering to their complaints, while they are not trained thoroughly in technical issues of the train (i.e., the locomotive settings, the characteristics of the rail track, the management of points and signalling systems). In addition, *train drivers* perceive them as unreliable or poorly competent practitioners in managing the train from a technical point of view. As a consequence, when this kind of critical scenario actually occurs *train managers* cannot accomplish their duties as rapidly as the *train driver* urges them to. Typically, the *train driver* violates the procedure by leaving the locomotive in order to manage the problem with the signalling system directly, and to reduce the train delay to a minimum. The *train driver* calls the *signaller* by phone and enters the control room of the station to interact with the manual controls. In these circumstances the *train manager* can just fill in a mandatory form with notes about the breakdown of the system, but s/he cannot replace the *train driver* in the locomotive, as s/he was not trained to use the locomotive controls. In addition, s/he cannot really help the *train driver* in speeding up the procedure, as s/he is excluded from the phone conversation with the *signaller* and relegated to a marginal role. This scenario, revealed by a *train driver*, points to a number of weak interactions in the activity of the *train manager*: the interaction with the *hardware* needed to fix the signalling problem and to control the locomotive, the interaction with the knowledge and skills (*software*) needed to manage the situation, the interaction with the mediating role of the *signaller* (*liveware*), etc.

An essential contribution of the previous and of this analysis phase is the construction of a model of the activity, suitable for identifying and discussing with the domain experts the weak interactions among the subjects and the different S-H-L resources.

The matrix example provided in figure 3 is an excerpt from a detailed analysis of the work process from which the mentioned railway scenario emerged. The representation includes a cluster of activities performed within a more general work process: “*Departure from platform 1 of train n°xxxxx from Station A to [...]*”. Even if the term *subject*

is not used, the *liveware* column includes all the human operators involved in the scenario and each work phase can be seen as an activity observed from the point of view of one or more subjects, sharing the same object. In 4 of the 5 steps depicted in the example subjects are using other roles as *liveware* resources to accomplish their task. Furthermore, corresponding to every step, there is also a selection of relevant *software* and *hardware* resources used by the subjects. For instance the *software* column contains some references from railway rulebooks, such as the RS, describing the procedures for the management of signals, the DET relating to the remote controlling system and the RCT, a general-purpose manual for all of the railway activities. The *hardware* column includes the telephone, the universal key to enter the control room, the paper form called M100, etc. The representation of the whole process, built and refined together with domain experts, was the tool to better identify the weak interactions and to start proposing possible remedial interventions in the following design phase (for further details see Bagnara *et al.* 1998, Rizzo and Save 2001).

#### 4b. Analysis of Strong interactions

In less structured domains and in particular for design interventions at the emergent level the analysis of strong interactions is preferable. Here, the aim is the creation of new forms of interaction, and consequently the focus is on what is already considered successful, since it represents best the intentions of the people, their motivations, and the core competencies characterising an activity. While in the case of weak interactions the analysis is concerned with problems and system breakdowns, in the case of strong interactions the analysis is concerned with positive characteristics and desirable outcomes in order to enrich and support successful activities or to exploit them to create radically new forms of interaction.

Two heuristics can be used to identify strong interactions:

- *Best outcomes*: a first strategy is to look for activities that are perceived as particularly efficacious by the stakeholders. In analogy with the first strategy for weak interactions the activities are observed from the point of view of their objects to identify those that are deemed as providing best outcomes. A major difference with structured domains is that in unstructured domains often the activity outcomes and their assessment criteria cannot be precisely defined a priori. Moreover, the success criteria are better related to process quality than to outcome characteristics. For instance, one of our projects addressed the design of an educational environment (called Pogo World), where innovative technological solutions could support the development of narrative capabilities of primary school children (Rizzo *et al.* 2002). In that case it would be a gross oversimplification to equate the story produced by a child to the main outcome of the activity, and even worse to measure the success in terms of a story “quality”. In fact, teachers (the most relevant stakeholders) rely more on a large variety of subtle indicators, such as the

active collaboration between children, their capacity to recall and reflect on past experiences, how easily they remained focused on the assigned task, etc. Subjective assessment by stakeholders is then recommended as the main criterion to identify strong interactions, in order to avoid drawbacks due to non-formally defined outcomes and the high relevance of process-related aspects.

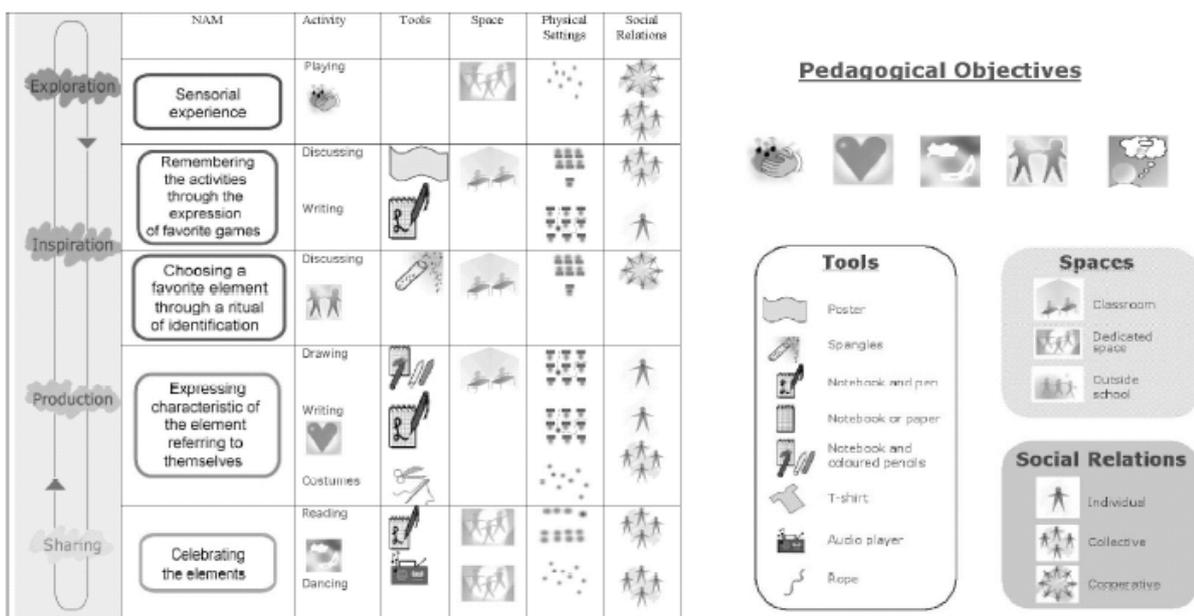
- *Personal satisfaction perceived by participants in the activity*: this second heuristic addresses the satisfaction, which subjects obtain by engaging in an activity. Thus, this heuristic is intended to represent the subject point of view. For instance, in the Pogo World case interactions were selected to reflect those most engaging for children. In domains where adults act as subjects, more diverse satisfaction criteria are likely to be identified, other than straightforward engagement. Those criteria are frequently neglected in the official organisational accounts. Nonetheless, the design intervention should consider them, as they often reflect the subjects' most important motivations and needs.

In the Pogo World design we exploited Narrative Activity Models as analysis and design representations. These models consist of two parts: a textual description of observed events (a scenario) and a tabular representation. The primary elements of the tabular representation include the main structural phases of the activity, their temporal sequence, the artefacts used, the type of activity (individual, collective, collaborative), the space where the activity took place (e.g., classroom, library, theatre etc.), the physical setting with a description of how the children and the teacher interact within that space, and the supported pedagogical objective. Figure 4 exemplifies the activity called "the element festival", where children first are

involved in games addressing the characteristics of the four natural elements, then are asked to remember and describe the games in order to choose a favourite element, and eventually to act as the chosen element in a collective game. All these activities are performed on a long time scale and in different places, nonetheless they share a common object (*telling a story where each child acts as the favourite element*) and possess their meaning only in relation to it.

All the elements of the unit of analysis adopted in this method are represented in the NAM, but they are distributed in categories that closely relate to the specific domain (e.g., the spatial disposition in relation to the teacher, or the physical space where the activity takes place), rather than strictly adhering to the *hardware-software-liveware* taxonomy. An overly analytical representation would lose clarity for the main stakeholders and domain experts (i.e., the teachers) and require them to think and reflect with categories unfamiliar for their domain. Instead, the NAM represents and groups the relevant elements of the activity in categories closely related to domain experts' experience. In this way the activity can be analysed easily in more detail whenever required by the design work, by adding details in the general framework of the NAM. Each row of the NAM represents a configuration (usually a strong one, since activities were selected from the most successful ones), and the relative array of resources. Teachers are then required to imagine what is likely to happen if any of the resources were eliminated ("sabotage method"), so that interactions contributing most to the strong configuration can be identified.

The output of this phase is an analytical representation of activities and of strong interactions therein. Furthermore, a



thorough understanding of strong interactions should be acquired, in order to preserve them in the design phase and use them as a lever for the introduction of enabling technologies.

In conclusion, we would like to highlight that the same approach can be applied to highly structured activities, like those accomplished in safety critical systems, in cases where the designer aims at an intervention at the emergent level. Even in very structured contexts, the focus on strong interactions is the best strategy to transcend current activity constraints, while at the same time preserving most of the “winning features” of the system (Marti and Moderini 2002).

**5. Verification with Stakeholders:** *Verification of models with stakeholders, further analysis.*

The models produced during the previous analysis stages have to be continuously verified with the stakeholders. This can be done with the stakeholders, who were observed, until uncertainties have been removed. This usually evidences inaccuracies, prompts further questions, and requires further analysis sessions to clarify issues and to remove problems.

The models can then be presented to and discussed with stakeholders who did not take part in the field study. This can be done both within the same organisation, and within other organisations. The aim of this step is the assessment of the adaptability and scalability of the models produced, in particular in structured domains. For example, models derived from observations conducted within the pharmacy of a particular hospital, can be discussed and assessed in focus group meetings, where pharmacists from a number of different hospitals participate. The models can serve as shared representations, and allow elicitation of further information from a larger number of stakeholders. This may also help with understanding the historical development of objects and motivations. Focus group meetings often are half-day meetings with about 10 – 15 participants.

**6. Interaction Design:** *Weak interactions are re-designed by increasing the redundancy or diversity of the mediating resources. Strong interactions are envisioned in brainstorming sessions with stakeholders utilizing and extending the descriptive activity models produced in the preceding stages.*

Drawing on the resources map and on the interaction analysis produced in the previous phases, the next step is the development of reactive, proactive or emergent design solutions. Again, the strategies and the objectives are quite different depending on whether the process is based on weak or strong interactions. However, a common element is the involvement of stakeholders in a variety of forms. This can range from the simple evaluation of prototypes to the participation in focus group and concept design sessions. As anticipated before, this phase should alternate and iterate with the analysis phase, considering that some

design proposals will probably stimulate the analysis of interactions previously not considered.

#### *6a. Weak Interaction Redesign*

Design based on weak interactions is normally a *redesign* process: most of the resources are already in place, but the designer needs to integrate, adapt and configure them in order to reach a stable and dependable system. Both in reactive and in proactive design actions the aim is to increase the system’s tolerance of the degradation of components, and of failures related to variable and unexpected operational conditions.

Based on the SHL taxonomy, we can define at least two basic strategies for an effective configuration of the mediational resources:

- *Increase redundancy of the system:* mediate activities with more resources of the same kind (S-S-S, L-L-L, H-H-H, etc).
- *Increase diversity of the system:* mediate activities with different kind of resources (S-H-L, S-H-L, S-H-L, etc.).

Taking into account the available design space (economic constraints, cultural context, stakeholder and decision maker attitudes) both strategies should aim at preventing the system from overloading single resources, and should adequately take advantage of their different mediational properties. For example, a work process relying on a considerable *liveware* redundancy (L-L-L) could be more dependable than a process critically dependent on only one *liveware*.<sup>3</sup> In case of unexpected conditions, provoking the exclusion of an operator from the process, another person could replace him/her in some of his/her tasks. However, the system can efficiently tolerate the disturbance only if the operators have correct expectations about their reciprocal behaviour, i.e., if they have a shared representation of at least a part of the process. In this case an adequate diversity (L-S-H) in the system should ensure the presence of the required *software* and *hardware* (e.g., common training sessions, shared physical environments for mutual awareness, good communication channels, written emergency procedures, etc).

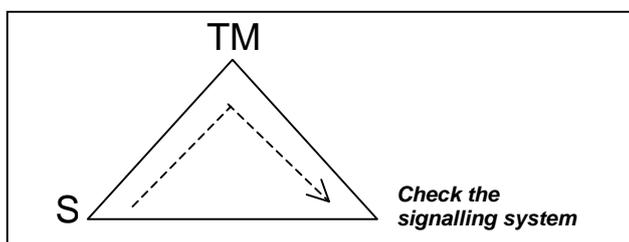
In a similar way a strong *hardware* redundancy (H-H-H) can protect the system from the effects of mechanical or human failures, providing automatic backups when they are needed. Yet, if all the backups act automatically and no informative feedback is provided to people, the humans could entirely lose the control of the process and be unprepared to manage new situations. Also in this case, an adequate diversity (H-L-S) should ensure that humans are never completely excluded from a general overview of the system, an essential requirement to manage any system (Reason 1990, Leveson 1995). (Examples of resources

<sup>3</sup> However, in the special case of humans (*liveware*), also the opposite effect is possible if this heuristic is not applied cautiously. A well-known (counter-) example is the issue of extensive double-checking (for example in maintenance tasks).

could be computer displays to monitor the process, usable control panels, quick reference manuals of the system, etc.).

The two strategies - redundancy and diversity - serve as a guide to the designer during focus group sessions with the stakeholders. Here, redesign solutions are proposed and negotiated using the activity models produced in the previous phases as a frame of reference to focus the discussion. At least one representative for each relevant role involved in the process under discussion should participate in the sessions, in order to adequately represent the different points of view (for example a *train driver*, a *train manager*, a *signaller*, etc.). Therefore, every design proposal should take into account its impact on the overall configuration of SHL resources. If a resource is changed or removed, the proposal should specify how its functions and its interactions with other resources will be replaced in the new configuration.

For instance, the focus group held on the above mentioned railway scenario (see Section 4a) resulted in a number of design proposals aiming at reinforcing the interaction between the *train manager* and the *signaller* speaking by telephone. In terms of the individual unit of analysis we can say that, in case of failure of the remote control system, the *train manager* functions as a *liveware* resource for the *signaller*, because s/he mediates her/his interactions with the local control room of the station (figure 5).<sup>4</sup> The interaction is critical due to the lack of the adequate *software* for guiding the behaviour of the *train manager*. Actually, the procedure combines a double weakness: firstly, the insufficient training of the *train manager* and secondly the impossibility for the *signaller* to see the console of the station, while the *train manager* is interacting with it. This forces the train manager to ask the *signaller* many questions to correctly guide her/his performance, and it obliges the *train manager* to spend most of her/his precious time in interpreting the *signaller*'s orders. However, the emergent redundant role of the *train driver* replacing the *train manager* does not seem an appropriate solution, because it removes the safe custody of the locomotive.



<sup>4</sup> As stated in the analysis phase also the *signaller* is obviously a *liveware* resource for the *train manager*'s objects. In this context we point to the *signallers*'s perspective because s/he is the role leading the checking signal process.

One of the proposals negotiated by the stakeholders was to build a quick reference manual (QRM) for the *signaller*, containing a schematic description of the consoles present in every small station of her/his control area (*introduction of a new software*). Another proposal consisted in a partial modification of the signals' console, including simple affordances and constraints for the *train manager*, such as blocking the controls s/he is not allowed to use and colouring the most important buttons and levers (*manipulation of hardware*).<sup>5</sup> Finally, the stakeholders proposed the introduction of a loudspeaker modality for the train personnel's mobile phones. The proposal was intended to allow the *train driver* to follow the conversation between the *train manager* and the *signaller*, during the signal checking procedure (*introduction of a new hardware*).

Output of this phase is a report containing the redesign proposals specified in the form of new activity models, complemented, when needed, by mock-up scenarios.

#### 6b. Design of Strong Interactions

The most important aspects of strong interactions redesign are the actual involvement of stakeholders and an effective integration of competencies coming from different domains. This phase is conducted through group discussions between designers and stakeholders to review and consider successful configurations, in order to enrich them through the introduction of new enabling technology and to ensure that their successful characteristics will not be disrupted. In other words, the design objective is twofold. On the one hand to ensure that the successful characteristics of the original activities are preserved and/or reinforced by the design intervention. On the other hand to stimulate the stakeholders in envisioning new solutions and scenarios, transcending the current constraints of the activities. Rather than correcting what is not working properly, the design team (designers and stakeholders) starts from what is more satisfactory and aims at discovering how this can be further improved by adding elements (taken from their own background) that ease or enlarge the possibilities of participants. Outputs of this phase are scenarios of use that represent and detail the design concepts by bringing together current successful activities and enabling technologies.

This goal is pursued by converging and by integrating the technical expertise of the designers and the professional knowledge of all the stakeholders. It is thus ensured that the activity evolves by benefiting from the already existing strong configurations (i.e., of what is currently deemed as successful), and that the introduction of enabling technologies is effectively woven into this pre-existing configurations. This also implies that knowledge and

<sup>5</sup> Note that in this case the "diversity" consists of replacing some of the *software* brought by the *train driver* in the unofficial practice, with a *software* embedded in the *hardware* of the signals' console.

competencies that currently characterise the activity are not spoilt, given that the activity evolves in accordance with the professional competencies of those involved (thus satisfying one important condition to acquire users' support for the introduction of technology).

Many techniques may be adopted in this phase, however for ease of analysis we will select a focus group session as example. We will try to highlight those characteristics that should be preserved, regardless of the chosen technique. The focus group session should bring together stakeholders, designers, and experts from other relevant disciplines (e.g. architects, musicians, engineers, etc). A focus group session can be deemed as successful if everyone can contribute effectively, so that the identified design solutions reflect competencies of most participants.

A key factor is the activity representation used to support the discussion, since the chosen representation should ensure that every participant can directly manipulate it to represent the aspects that s/he deems as more relevant. In the previously mentioned case of Pogo World design (see Section 4b) we used the NAMs obtained in the previous phases to define Enhanced Narrative Activity Models (ENAM). The ENAM represents the integration of proposed concepts and enabling technologies into the corresponding NAM, such that a narrative scenario of use is obtained. ENAMs are obtained by modifying the NAMs' textual scenario and tabular representation to show the exact role of the design proposals in the envisioned situation of use. In practical terms, focus group participants are requested to identify design concepts, and then to discuss them in the focus group session to imagine how proposed design concepts might be integrated in the corresponding NAM. The group discussion should both highlight some potential problems (for instance, if any other successful activity is likely to be negatively affected by the design solution) and, most of all, add participants' specific knowledge to refine the design. For instance, a designer may propose to support the activity with a specific technology or with a practice exploited in a different domain, while other participants (in the Pogo World case mostly the teachers) are requested to detail the representation (adding elements and specifying their interactions, or simply commenting) in order to anticipate if the proposed solution is likely to work effectively and which interactions will occur.

Outputs of this phase are representations (e.g., scenarios of use) of the envisioned activities, where the integration between concepts/enabling technologies and the activity is apparent. Design proposals should also be characterised in terms of the underlying strong interactions, i.e.: they should present situations where subjects can achieve their goals in multiple interchangeable ways.

## CONCLUSIONS

The method presented in this paper is intended to provide indications for the design of systems in the form of written reports, complemented with design concepts and interactive

prototypes that could have been developed during the design process. However, as mentioned at the beginning of this paper, the main aim of the method is providing an additional resource for the design process, that is a map for the choice and application of the most suitable methods and tools among all the traditional ones. As such, even the final outcomes are deeply related to the specific characteristics of the whole design process, to its history and objectives, to the level of intervention and the level of structure of the context. Nevertheless the revised mediation triangle and the selected type of interactions (either weak or strong) remain as common elements in all the design processes, as they help in identifying the appropriate spaces for design in all the different circumstances. As such they also support the cross-fertilisation among different design experiences, encouraging a rich interchange of concepts and methodologies from a domain to the other.

A less tangible but equally important outcome of the method is the acquisition of design competence and knowledge by the involved stakeholders. This is probably the area where considerable work has still to be done. It is often the case that during the design process stakeholders tend to reason in analogy with their current situations, and cannot properly judge what the introduction of technology might imply for their work. Most of our current efforts are directed to maximise stakeholders' active participation, rather than simply having them as subjects of study, or external evaluators (Lanzi *et al.* 2004). In the same way, the management is often biased towards the introduction of state-of-the-art technologies, and further efforts should be invested to ensure that their attention is also directed to the subjects and objects of the activity. An active participation of stakeholders (both management and front-line people) in the design process can contribute to raising their awareness of design methods and tools, thus mitigating some of the most common organisational biases (e.g., fixation on ready technological solutions or reluctance to adopt innovations).

## REFERENCES

- BAGNARA, S., DI NUCCI, P., PASQUINI, A. and RIZZO, A., 1998. Progressive strategies for introducing human factors in safety- critical system. *Manufacturing agility and hybrid automation*.
- CARROL, J.M. 1995, ed., Scenario-Based Design. Envisioning Work and Technology in System Development, John Wiley & Sons, New York.
- COLE, M., 1996. Cultural Psychology. A once and future discipline, The Belknap Press of Harvard University Press, Cambridge (MA).
- EDWARDS, E., 1988. Introductory Overview, Wiener, E.L. & Nagel, D.C. (eds.), *Human Factors in Aviation*, Academic Press, New York.
- ENGESTROM, Y., 1987. Learning by Expanding: An Activity Theoretical Approach to Developmental Research, Orienta, Helsinki.

GEERTZ, C., 1973. *The Interpretation of Cultures*, Basic Books, New York.

LANZI, P., MARTI, P., POZZI, S. and SCRIVANI, P., 2004. Users as cultural mediators in interactive system design. *In Proceedings of ECCE12*, York, UK, September 2004.

LEVESON, N.G., 1995. *Safeware. System Safety and Computers*, Addison Wesley Publishing Company, Reading (MA).

MARTI, P. and MODERINI, C., 2002. Creative design in safety critical systems. *In Proceedings of ECCE11*, 159-166, Catania, Italy, September 2002.

MARTI, P. and RIZZO, A., 2003. Levels of Design: from usability to experience. *In Proceedings of HCI International 2003*, Crete, July 2003.

NORMAN, D. A., 1990. *The Psychology of Everyday Things*, Basic Books, New York.

REASON, J. T., 1990. *The Human Error*, Cambridge University Press, Cambridge.

RIZZO, A. and SAVE, L., 2001. SHELFs: A Proactive Method for Managing Critical Issues. *In RTO Meeting Proceedings 32 - NATO Research and Technology Organization*, January 2001.

RIZZO, A., MARTI, P., DECORTIS, F., MODERINI, C. and RUTGERS, J., 2002. The design of the POGO story world. In E. Hollnagel ed., *Cognitive Task Design*, Erlbaum, Mahwah (NJ).

VYGOTSKIJ, L.S., 1978. *Mind in Society. The Development of Higher Psychological Processes*, Harvard University Press, Cambridge (MA).

WERTSCH, J. V., 1985. *Vygotsky and the social formation of mind*, Harvard University Press, Cambridge (MA).

Tab. 1 - Tabular summary of the individual steps of the methodology.

<b>Step</b>	<b>Description</b>	<b>Tools</b>	<b>Outputs</b>
1. Determination of level of design intervention and level of structure	Determine the level of design intervention (reactive, pro-active, emergent) and the level of structure of the domain (ranging from fully structured to unstructured). This will determine the time spent on subsequent phases, the analysis and design focus (weak/strong interactions), and the tools, techniques and representations used.		Classification of level of design intervention and level of structure of the domain; rough plan of tools and representations to be used, and of time to be spent on the next steps.
2. Preliminary identification of subjects and activities	Identify basic structure of the organisation under consideration, and relevant activities of the observed system, focusing on subjects and objects (from individuals to roles).	Stakeholder interviews, document review	Basic structure of organisation under consideration, set of basic activities to be considered, set of basic roles, set of contacts for analysis
3. Resource mapping	Map the relevant means/resources (tools, procedures, practices, people, roles, etc.) which mediate the activities.	Interviews, observations, audio & video recordings, document review	Set of activity models (HTA, workflow matrix, scenarios, Narrative Activity Models etc.)
4. Analysis of interactions	Analyze interactions between subjects and artefacts, in relation to the respective objects. Depending on the context (structured, unstructured) and on the design level (from reactive to emergent), focus either on strong or weak interactions.	Interviews, observations, audio & video recordings, document review	Set of weak / strong interactions. Analysis of influencing factors.
5. Verification with stakeholders	Verification of models with stakeholders, further analysis.	Interviews, Focus Group meetings	Revised models
6. Interaction design	Weak interactions are re-designed by increasing the redundancy or diversity of the mediating resources. Strong interactions are envisioned in brainstorming sessions with stakeholders utilizing and extending the descriptive activity models produced in the preceding stages.	Focus Group meetings, brainstorming, mock-ups, simulations	Reports describing re-design scenarios, mock-ups, extended activity models

# IS Design as Domain Construction

Lars Taxén

Campus Norrköping, Linköping University

Department of Science and Technology

SE - 601 74 Norrköping, Sweden

{[lars.taxen@telia.com](mailto:lars.taxen@telia.com)}

## ABSTRACT

In this paper we describe a method for IS design which focus on constructing the social reality in which the IS is used. This reality is structured as a particular form of work practice – the activity domain – which is the main construct in the Activity Domain Theory. The gist of this theory is to integrate coordinating elements of a practice into a coherent whole in which the IS is one of these elements. The theory originated in the Ericsson telecommunication company where it has been gradually refined over more than a decade by the author. It has profoundly influenced the coordination of the development of the 3<sup>rd</sup> generation of mobile systems.

## Keywords

IS design, coordination, praxis, construction of social reality, shared meaning.

## MOTIVATION (initial reflection)

Product developing organizations are facing a turbulent reality today due to increased product complexity, diversification of organizational functions and an ever increasing rate of change. An example of this from Ericsson is the “anatomy” shown in Figure 1. The anatomy shows the coordination of development tasks (square white boxes) in one of the nodes in the 3<sup>rd</sup> generation of the mobile system network. Each task, which is called a ‘work package’, develops a specific functionality. The thin lines mark dependencies between the packages, indicating which packages must be ready in order for other packages to function properly. The development is carried out in the same order as the actual system ‘comes alive’, hence the term anatomy. Thick arrows show the datum for a particular integration and verification of the packages. Small dots indicate the status of a package such as ‘in design’, ‘in test’, ‘delayed’, ‘ready’, etc. The ovals signify basic

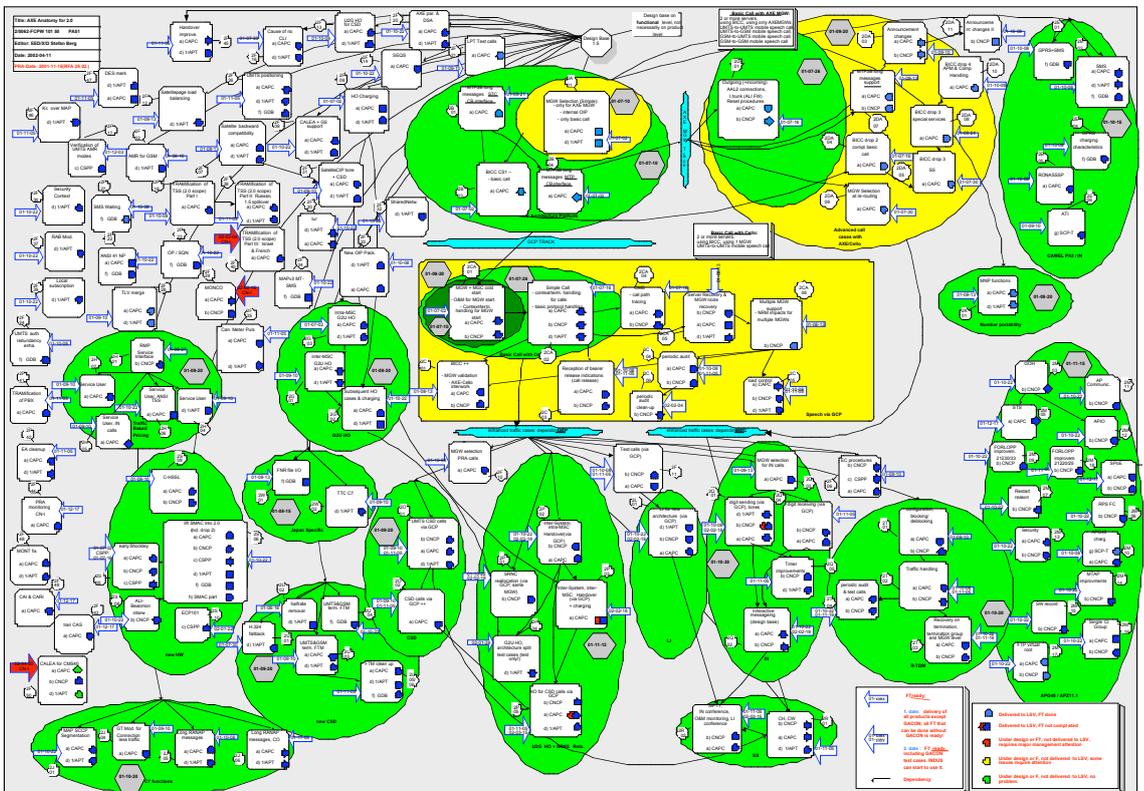


Figure 1 The anatomy of a node in the 3<sup>rd</sup> generation of mobile systems

services like registering the location of the mobile, calling to the mobile, answering a mobile call, etc. The work packages are developed by teams distributed all over the world. In most cases the functionality is provided by software, where the total number of source code lines may be in the order of millions.

The coordination of a development task like this requires information system (IS) support. A product in a telecom system may consist of several hundreds of sub-products, each one described by a number of product related documents. In addition, other types of items such as requirements, engineering change orders, baselines, milestones, etc. must be considered. All in one, between 5,000 to 10,000 items must be tracked with respect to their revisions, states, dependencies, etc. Moreover, the development task is constantly revised due to changed customer requirements, new insights, errors discovered, available resources, etc. Thus, the technical challenges of developing IS support for this kind of application are indeed considerable.

However, the most arduous task in these circumstances is to establish a workable consensus among the actors concerning the nature of the coordination (Taxén, 2003). First, there must be a sufficient level of shared meaning about what should be coordinated and how. This concerns the identification of which items are crucial for coordination, how to characterize them and how to relate them to each other. Often, new abstract concepts are introduced, something which is particularly difficult to acquire a shared meaning about (March & Simon, 1958). Second, the actors may be geographically dispersed, have different roles, come from different traditions, speak different languages, etc. Third, the contents and structure of coordination will change according to new insights, new demands from the market, new tools and methods supporting coordination, etc. Finally, cues in models and diagrams such as those in Figure 1 must make sense to the actors.

In this contribution we describe an IS design method which addresses both the technical and social issues as described above. The gist of the method is to construct *the entire work practice* in which the IS is used. This means that the IS is but one element being constructed. The most important result is the construction of a social reality in which shared meaning is one of the outcomes. Thus, rather than focusing on the core of the IT artifact as Benbasat & Zmud (2003) suggest, we move in the opposite direction. Our focus is not the core of the IT artifact, nor the IS in its context, but rather the context in which the IS is immersed.

In order to achieve this, the work practice is structured as an *activity domain* (Taxén, 2004). An activity domain may be regarded as particular perspective of a work practice where its coordinating elements are emphasized. The activity domain is the central construct in a new theory for coordinating human activity – the Activity Domain Theory (ADT). The ADT has many features in common with the Activity Theory (AT) (e.g. Engeström, 1987; Bedny et al., 2000). However, ADT also differs in essen-

tial aspects from AT. The experiences show that the proposed method derived from ADT enables the design of ISs which can support the coordination of very complex system development tasks while taking individual, social and technical aspects into consideration.

The paper is outlined as follows. In the next sections we describe the method and its outcome in detail. This is followed by some practical experiences. Next we describe the ADT and compare it with AT. We finish up with some reflections about the transferability of the method to other settings beside Ericsson.

## **METHOD (practical)**

In this section we describe the main features of the method, which is called “domain construction strategy”. The reason why we call it ‘strategy’ rather than ‘method’ is that the fine graded steps have to be defined for each individual application.

## **Result**

The result of the method is a constructed social reality – the activity domain – in which the IS supporting the coordination of tasks is one of its elements.

## **Form of result**

The intangible form of the result is a shared meaning among the actors about what constitutes coordination and how it should be carried out.

The tangible forms of the result in terms of tools and artifacts are as follows.

### *The context model*

This model signifies the structure and extension of the activity domain. It shows what types of phenomenon are considered relevant in the domain, their relations and their characterization in terms of attributes, state sets, revision rules, etc.

In order to facilitate the signification process it is important that the model notation is easily understandable. One such notation is based on the Object Modeling Technique (OMT) (e.g. Rumbaugh et al., 1991). The Universal Modeling Language (UML) notation is less useful since it has a rich repertoire of constructs which usually are familiar only to specialists. Standard drawing tools like PowerPoint may be used to describe the model (see the example in Figure 3).

### *The coordination model*

The coordination model signifies the dependencies between the tasks in the domain. This model has the same purpose as ordinary process models. The notation used is called Information Flow Diagrams (IFD) (see the example in Figure 8). Again, the signifying properties of the model are the prime concern.

### *The transition model*

The transition model signifies how different activity domains interact. This model is an elaboration of the Specification Based Data Model suggested by Gandhi & Robertsson (1992). The main influence from this model

is to direct the attention to the transition between domain borders. For example, the status of a work package in Figure 1 is assigned according to the states of work package internal items such as documents, etc.

**The domain core**

The purpose of this result is to be a place-holder for various items which provide stability to the domain. Examples of such items are identification rules, notations, cues, etc.

**The running application: the IS supporting coordination**

Typical features implemented are support for requirement management, configuration management, test management, project planning and control, etc.

**Phase of the design process**

The basic mode of design in the method is an ongoing interaction between reflection and action. Tentative models are implemented in the IS and tried out in the development practice which is to be coordinated. Thus, the method, which may be characterized as an experiential learning based method (Kolb, 1984), does not follow the traditional phases of requirement analysis, design, implementation, testing and deployment. As described below, the method rather seek to achieve a sufficient level of shared meaning about the work practice in which the IS is used. This means that the IS is gradually being shaped by the actors in the practice alongside the emergence of shared meaning. Therefore, requirements are not detailed in advance. Rather, they are stated on a high level such as "There shall be support for requirement management and engineering change order management".

**Type of systems**

In the applications so far the IS platform has been Matrix (Matrix-One, 2004). This system is targeted as a backbone for managing product related data in large, globally distributed organizations. It can be characterized as a high performance, complex system of its own.

In Matrix the domain models are implemented without programming in the type definition module (called the Business Modeler). The instances (objects) are managed in another module of Matrix. Besides these two modules there is a module for administration of the system and a module for communicating directly with the database.

**Type of design process**

Since the method aims at the construction of an activity domain, it might be characterized as inherently in-house. Due to situational circumstances such as historical influences, actor's knowledge, available resources, norms, values, etc., activity domains will be constructed differently regardless of the whether the domains provide the same type of result. Moreover, the issue of re-design is not relevant since a continuous modification of the IS is a deliberate feature of the method. Also, the method presupposes that the major stakeholders are participating in the design. Thus, the method can be characterized as a participatory design type of method.

**Who is performing the method?**

Since the method aims at the coordination of development task such as projects, the main stakeholders in projects are participating. Typically, these are project managers, requirement managers, configuration managers, product managers, etc., who can be characterized as users. In addition, IS platform specialists are participating. Finally, actors with an expertise in domain modeling are involved to provide a bridge between the users and the IS platform specialists. This means that users, IS specialists and domain expertise are all participating in the system design. The borders between these competences are blurred. Rather, the actors bring their expert knowledge into a common playground where they together construct the activity domain.

**Competences needed**

No particular competence is needed besides understanding and accepting the ideas and concepts in the ADT. However, this is not trivial. For example, an organization used to develop systems in a linear fashion might not accept the constant modifications strategy inherent in the method.

**Procedure**

The construction of the activity domain requires certain prerequisites. Besides the usual ones of personal and financial resources, management approval, etc., the most important prerequisite is the availability of the IS platform. The capacity of the platform and the communication network must be secured. This is especially important if the IS is to be used globally. Also, strategies for replication and synchronizing data exchange must be defined and tried out.

The construction of the domain is carried out in three phases: *exploration*, *trust boosting* and *expansion* (see Figure 2). In the first two phases the focus is on establishing the activity domain as a 'bridgehead' in one project before expanding it to other projects in the third phase. This means that the gist of the strategy is to quickly establish a relatively stable core of shared meaning in a small group of actors which is then propagated to other actors in an ongoing domain construction process.

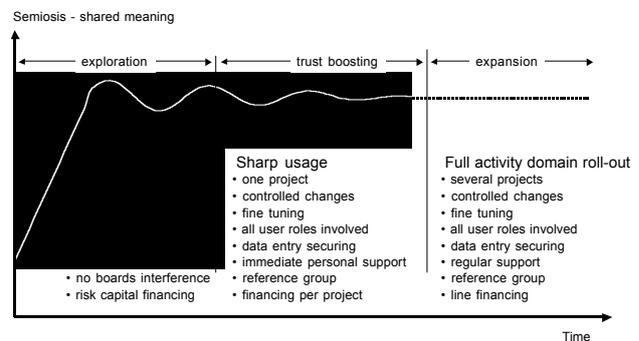


Figure 2. The domain construction method

### Exploration

In this phase the initial construction of the domain is carried out. The main purpose is to rapidly achieve a tentative consensus about the content and structure of the domain. The work is carried out in a ‘daily build’ manner in close interaction among the actors. The work is financed on a risk capital basis. Detailed return on investment analysis is not required since the reliability of such analysis will be low.

The following tasks are carried out in this phase:

1. State the coordination requirements on an overall level, for example, “There shall be support for engineering change order management, requirement management (RM) and work package based software development”. These different areas are called *coordination areas* and might be considered as domains on their own.
2. Define a ‘task force’ for each coordination area. For example, for RM this force may consist of the project manager, the requirement manager, the domain modeler and the IS specialist.
3. The established methods for, say RM, may be used as a point of departure. From these a first version of the context model is proposed by identifying the relevant phenomena and how these are related to each other. Define attributes, cardinalities, revision stepping rules, state sets, etc. In Figure 3 an example from Ericsson of a context model for RM is shown.
4. Implement the context model in the IS. In the IS used at Ericsson, Matrix, no programming is needed to do this. The boxes are implemented as types, the links as relations, state sets as ‘policies’, etc.
5. Instantiate object of the types, for example, a number of requirements and requirement issuers, and relate them to each other. Create reports. Evaluate the information: What is missing? Is this correct? Etc.
6. Make changes to the context model and implement these anew. Continue in this manner until the actors agree that the constructed domain is useful.

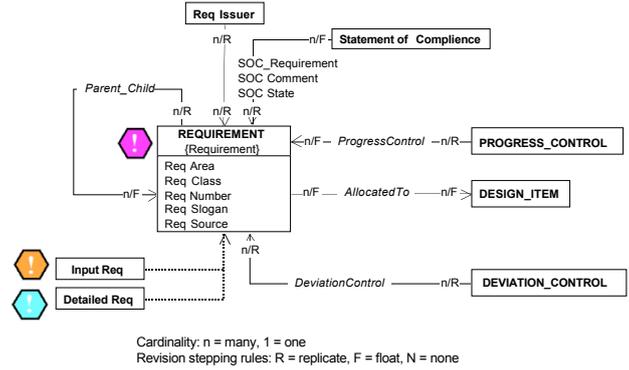


Figure 3. An example of a context model for RM

### Trust boosting

The purpose of this phase is to boost the trust about the feasibility of the domain as constructed in the exploration phase. Key issues are getting all actors in the project to trust the data in the IS and to make sure that the performance of the IS is acceptable at all units world-wide. This is done in one sharp project, that is, a project which develops a product for some client. The task force is still driving the construction. All user roles around the project are involved and immediate, personalized support is provided. The development of the domain progresses by controlled changes and consists of fine tuning steps. No major reconstruction of the domain is allowed. Reference groups and steering boards are consulted and the financing is done on a project basis.

The following tasks are carried out in this phase:

1. Transfer data from previous sources into the IS. For example, requirements previously kept as text in requirement specification documents are translated into requirement objects which can be individually managed and related to other items according to the context model.
2. Set a date when the project shall start using the data in the IS as their primary source for planning and monitoring the project. The reason for this is that the data otherwise may be inconsistent.
3. Take measures to make the actors use the IS, i.e. enter data into and retrieve data from the IS. This can, for example, be done by requiring that the only source for progress reports is the data in the IS.
4. Keep a list of issues that need to be attended to. Any such issue needs to be agreed upon by the task force before it is implemented in the IS.

### Expansion

In this phase several projects are included in the domain. As in the trust boosting phase, the construction is done by controlled changes, however now in a formalized way. The financing is done by the line organization rather than the project organization to keep the domain intact between projects.

The following tasks are carried out in this phase:

1. Changes in the models must be agreed upon by a change control board before it is implemented in the IS. This means that an incoming issue is sent out to an analysis group where its impact is estimated in terms of cost and implementation effort. A formal decision to go ahead is taken and the issue is followed up until it is fully implemented.
2. If the domain has to cooperate with other domains a work must be initialized to coordinate these domains. When doing so, a balance must be struck between what is absolutely necessary to coordinate and what can be left to each individual domain to decide independently.

**EXAMPLE (practical)**

The example is taken from Ericsson. During the late 1990s Ericsson was in a process of replacing the so far dominant ‘waterfall’ software development method. It had become painfully clear that this method was unable to cope with issues such as increased turbulence of the market, more complex systems and organizational upheaval.

The aim of the replacement was to come up with some kind of incremental development method which enabled the system to be implemented and tested in small steps in contrast to the ‘big bang’ approach in the waterfall method. Before 1996 some projects had attempted to use various incrementally flavored methods, but there was no shared meaning on the essence of this approach.

A project was initiated with the purpose of defining a methods package for the incremental development of large software projects. This project, in which this author participated, had severe problems in agreeing on what constituted incremental development. Only when the strategy suggested in this paper was applied, shared meaning began to emerge. In Figure 4 an example from 1997 of the context model is shown. It can be seen that the focus on incremental development brings about several new categories in the domain (the grayish boxes).

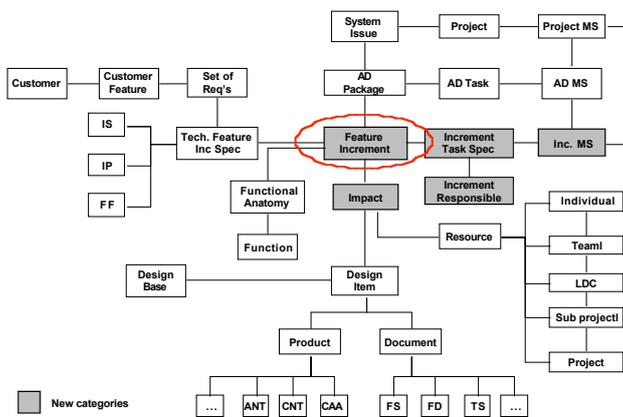


Figure 4. A context model

In Figure 5 the associated coordination model is shown:

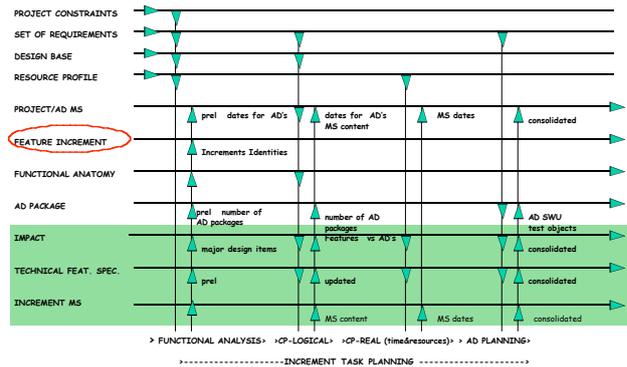


Figure 5. A coordination model

The implementation of the context and coordination models is shown in Figure 6.

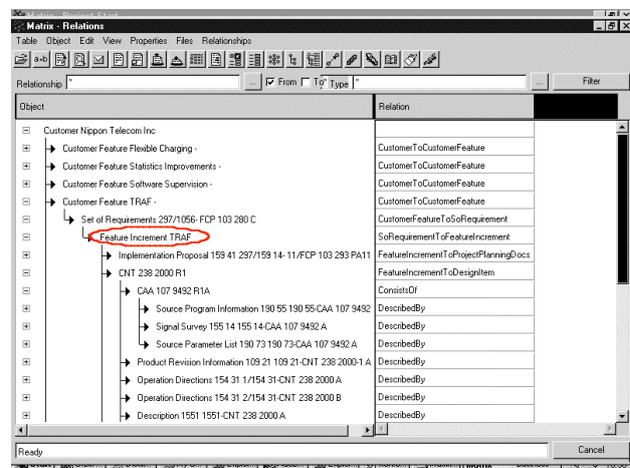


Figure 6. An implementation of the context and coordination models in the IS.

As can be seen, the same type of entities appears in the context and coordination model. In the IS, instances of these types are shown. The entity type in focus, the ‘Feature Increment’, is indicated by an oval.

By a continuous interplay between these three elements the domain for coordinating the incremental development of software was gradually constructed. This process was going on as long as the domain existed. As an example, in 1998 the context model was modified several hundred times. More details about this can be found in Taxén (2003).

**Experiences**

The domain construction strategy began to influence the Ericsson practice around 1996 with the introduction of the method package for incremental development of large software systems. The first sharp project to use Matrix was carried out in 1998. Between May 1999 and mid 2002 the number of projects using the strategy rose to around 140 distributed over more than 20 development sites worldwide. During this period four coordination domains were constructed. As indicated in the following

statement the impacts on the Ericsson practice were profound:

“Especially for the execution part I think we would not have been able to run this project without the tool. I think if you simply look at the number of work packages, the number of products that we have delivered, the number of deliveries that we have had, if we would have had to maintain that manually, that would have been a sheer disaster. [...] we had some, only in my part of the project, some 200 work packages or work packages groups or whatever you want to call them, deliveries, on the average 2-5 subprojects within them 5-10 blocks being delivered, just keeping track of that [...] would have been a hell of a job.” (Project manager, 3G development)

Other identified effects are reported in Taxén (2003).

### THE ACTIVITY DOMAIN THEORY (reflection)

The ADT was developed by the author in his professional work at the Ericsson telecommunication company over a period of more than 10 years (for details, see Taxén, 2003). Usually a certain element in the theory was triggered by a need in the Ericsson practice.

For example, in the early 1990s Information Flow Diagrams appeared as an alternative way of modeling processes. One example of such a diagram from 1994 is given in Figure 8. The process model was printed on a large sheet of paper and put on the wall in the project room. This meant that all actors involved had the same picture of the task and could easily orient themselves by this picture.

A striking insight was that a very complicated design process could be coordinated by a comprehensive picture. No sophisticated tools were needed. What mattered was that the actors involved had some shared meaning of the picture. Thus, its signifying and coordinating qualities were of prime importance. These observations gradually matured over the years. They were eventually incorporated in the ADT as the temporalization modality and a focus on signs and their mediating roles in the coordination of human activity.

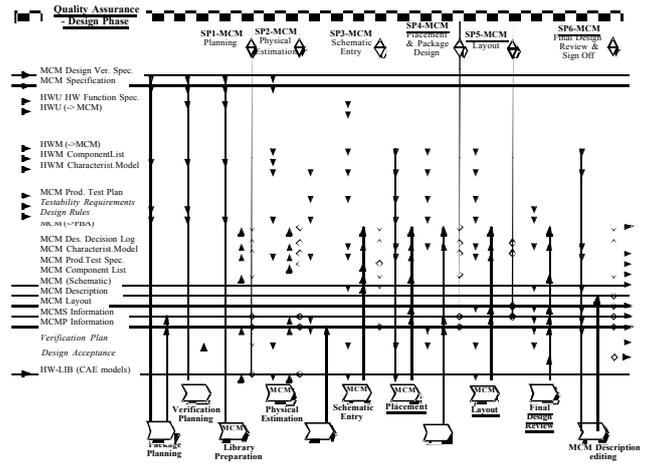


Figure 8. Information Flow Diagram for Multi-Chip Module design

This pattern was repeated for other elements of the ADT. Between 1990 and 1998 these elements were gradually shaped by practical experiences. Between 1998 and 2003 the ADT was theoretically grounded in the author’s Ph.D. studies alongside with further empirical grounding in the Ericsson practice. Thus, the ADT was developed in close interaction between theory and practice.

At present, the theory has been applied in the Ericsson setting only. However, the aim of the ADT is bold: to provide an integrating framework for coordination that can be utilized for analytical and constructive purposes, including IS design. It is also our ambition that the theory will open up new lines of research in organizational studies.

When reflecting on human activity, usually an individual or a systemic perspective is taken as the Unit of Analysis (UoA). However, as a long discourse has shown, neither of these approaches is entirely satisfactory (e.g. Volo\_inov, 1929/1986). The individual perspective tends to ignore trans-individual phenomena such as social institutions and the structural properties of language. On the other hand, the systemic perspective easily downplays individual phenomenon such as cognition, meaning and everyday utterances.

To overcome this dilemma the *practice* has been suggested as a proper UoA where the individual and systemic may be reconciled. In this approach, the practice is regarded as the primary generical social thing (Schatzki, 2001:1). This reflects an ontology where the “... social is a field of embodied, materially interwoven practices centrally organized around shared practical understandings.” (ibid.3). Practices are considered to be a materially mediated nexus of activity where the “... forms of individual activity depend on the practices in which people participate.” (ibid.11). Thus, both the individual human mind and social order are to a significant extent considered to be constituted within practices.

In order to make the notion of practice operational we will start from a Marxist perspective. Here, socially organized

labor is seen as the basic form of human activity. "By thus acting on the external world and changing it, he at the same time changes his own nature" (Marx, 1867/1967:177). In particular, the ADT uses the theoretical perspectives of *praxis* as developed by the praxis philosophers in the former Yugoslavia. In this school, praxis permeates the whole of man and determines him in his totality:

"In its essence and generality, praxis is the exposure of the mystery of man as an onto-formative being, as a being that *forms* the (socio – human) reality and *therefore* also grasps and interprets it (i.e. reality both human and extra-human, reality in its totality). Man's praxis is not practical activity as opposed to theorizing; it is the determination of human being as the process of *forming* reality." (Kosik, 1976:137)

Epistemologically ADT accepts "... that we can have no objective, observer-independent, access to reality but ... there is an independent external world constituted by structures or entities with causal powers..." (Mingers, 2001:118). In interaction with this reality man constructs a social reality (e.g. Searle, 1995). The meaning of any relevant phenomena in a practice is the result of social interaction processes among the actors in the practice. As in pragmatism, *usefulness* is the most important criterion by which a certain action is considered valid or not (Wicks & Freeman, 1998). For example, it may be suggested to use a certain concept like 'increment' in a software development practice. If this leads to successful results, 'increment' will be recognized as useful in that practice.

It is possible to conceive of practices as the concrete, everyday manifestations of praxis. The praxis perspective emphasizes certain qualities of human activity such as historicity, dialectical interaction, contradictions as the drivers of change, etc. By introducing the construct of *activity domains* in the ADT we strive to maintain these qualities while simultaneously giving practice a structure which is suitable for analytical and constructive purposes. This means that both the practice and its constituting elements may be taken as units of analysis.

### The constitution of activity domains

When characterizing human activity at least the following aspects should be considered:

- The activity is a systemic entity which is in constant development. Its elements as well as the activity as a whole are in constant motion.
- The elements and the whole are dialectically related to each other. The whole is given its properties from the parts and, equally important, the parts are given their properties from the whole.
- The characterization of the activity should be grounded in the individual cognitive system as well as in the social practice where the individual acts.

Usually, the characterization of systemic entities proceeds from a structural perspective, that is, from an analysis of entity elements and how they relate to each other. For example, Bedny & Karwowski propose a structure consisting of subject, task, tools, methods, object and result (Bedny & Karwowski, 2004b:140).

However, since the basic feature of activity is development and motion, we propose to proceed from this perspective. To this end we will characterize the activity in terms of *activity modalities*, where 'modality' is apprehended as "a modal relation or quality; a mode or point of view under which an object presents itself to the mind." (Webster's 1913 Dictionary). The modalities can be apprehended as types of dynamic, inner processes within the activity which are dialectically interrelated.

During the socio-historical development of the activity domain two forms of objectivizing emerge (Kosik, 1976): *objectification* and *objectivation*. Objectification ("Vergegenständlichung") concerns the transformation of the world into objects such as tools, institutions, organizations, etc. Objectivation ("Objektivierung") refers to the integration of man in a trans-individual whole as one of its components. This incorporation transforms the subject: "The subject abstracts from his subjectivity and becomes an object and an element of the system." (ibid. 50).

This means that each modality will be manifested in two ways: as objectified, "external" objects and objectivated, "internal" modes of cognition. From this we conjecture that activity modalities are suitable candidates for the grounding of activity in both individual cognition and social practice. Thus, we assume that human activity systems are constructed in resonance with the cognitive apparatus of humans. For example, with the emergence of symbolic thinking it became possible to conceive of a temporal dimension besides the immediate here and now. This is reflected in ordinary practices by plans, processes, calendars, etc. but also in the neural system of humans as in the following example:

"It was demonstrated that some *neurons* in the cerebral cortex could react to several stimuli from different modalities at the same time, processing light, sound and sense by touch, pain, etc. These neurons react not only to stimuli from different modalities, but are also able to *select stimuli important to the temporal needs of the organism* [our emphasis] from many external and internal influences." (Bedny & Karwowski, 2004:263).

These considerations are reflected in the construct of activity domains, which are characterized as follows. In the activity domain, *actors* come together in order to produce a *result*. As actors they participate in socially organized labor where individual-psychological goals, motives, ambitions, etc. are aligned and transformed into trans-individual goals and motives. In this sense, the activity domain has a *motive*, which is the reason why the activity domain exists. Likewise, the domain has a *goal* which

adheres to the motive of the domain. Starting from certain *prerequisites* the actors modifies an *object* according to the goal of the domain. The result is the actual *outcome* of the activity. This result may in turn become a prerequisite for other activity domains.

These elements are influenced by the Activity Theory (Engeström, 1999; Bedny & Harris, 2004). Next, we propose that the activity domain can be characterized by the following activity modalities:

- *Stabilization*: Over time, actors in the domain develop a common ideology, by which we understand any wide-ranging systems of beliefs or ways of thought. The ideology stabilizes the activity domain and is manifested as norms, values, routines, rules, etc. The coherence to domain ideologies may destabilize the cooperation between domains if shared ideological elements are not developed.
- *Contextualization*: In the activity domain, the actions are focused and situated. This means, for example, that a particular phenomenon will be apprehended and characterized differently depending on the context in which it is considered relevant.
- *Transition*: Activity domains interact with each other. The outcome of one domain is the prerequisite of other domains. Since the stabilization brings about partly different domain ideologies, the result may be characterized differently. If so, there is a need for a translation and interpretation of the result in the transition between the domains.
- *Spatialization*: In the domain actors orient themselves spatially. This orientation concerns which phenomena actors perceive as relevant and how these are related.
- *Temporalization*: In the activity domain actors orient themselves temporarily. This orientation concerns the dependencies between the tasks in the domain.
- *Mediation*: In the activity domain the actions are mediated by instruments which can be essentially material or symbolic in character, like a hammer and a law.
- *Communication*: Communicative acts are performed by actors in order to reinforce their coordination (e.g. Goldkuhl & Röstlinger, 2002).
- *Interaction*: Interaction is fundamental for meaning creation (semiosis). During interaction, human mental processes evolve. The specificity of this interaction is determined by the socio-cultural development of the activity (e.g. Bedny & Karwowski, 2004b:138).

The reason why precisely these modalities have been included in the ADT is that they have been identified as strongly influential in the practice of complex systems development (Taxén, 2003).

### Objectification

The objectified forms of the modalities are collected in a Framework as follows (ibid. 2003):

- A *context model* which emanates from the contextualization and spatialization modalities.
- A *coordination model* which emanates from the temporalization modality.
- A *transition model* which emanates from the transition modality.
- A *stabilizing core* which emanates from the stabilization modality.
- *Information systems* which emanates from the mediation modality.
- *Communicative acts* which emanates from the communication modality. Such acts may be assignments, agreements, commitments, requests, etc.
- A *domain construction strategy* which emanates from the interaction modality.

Most of these elements have been discussed in detail in previous sections. Together these elements constitute the activity domain (see Figure 7).

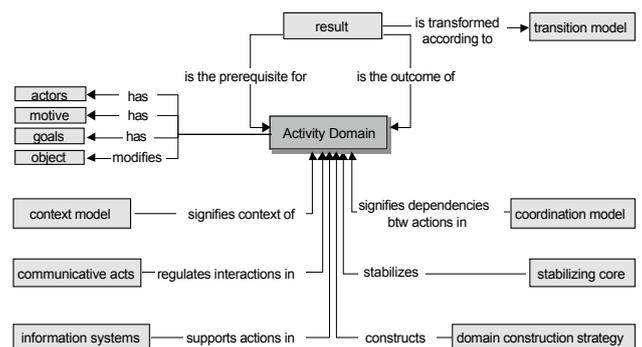


Figure 7. The constitution of the activity domain

It can be noted that the activity domain is a recursive construct. The transition model makes it possible to regard the activity domain as embedded in a larger context where other activity domains provide prerequisites for and uses the outcome of the activity domain.

### Objectivation

Objectivation implies that the individual actor has acquired an understanding of the meaning of the elements in the activity domain. Otherwise she cannot perform meaningful actions in concert with other actors.

According to Bedny & Karwowski (2004c) it is important to distinguish between meaning and sense. Meaning has an objective character and is referred to as “objective meaning” while sense has an individual, subjective character and is referred to as “subjective sense”. When acting in a particular situation the “... system of subjective representations of the situations unfolds in the form of dynamic models...” (ibid. 136). These dynamic models allow the actor “... to quickly orient in the current situa-

tion and adequately regulate his/her actions.” (ibid. 137). The dynamic model is continuously transformed and adjusted by reflection and self-regulation. What makes the dynamic reflection of the situation in the mental model possible is the “transformation of objective meanings into subjective senses and their integration into [a] holistic framework.” (ibid. 137). This in turn enables further action which may impact the objective meaning. In accordance with the previous discussion, the meaning and sense making processes may be called objectivation and ‘subjectivation’ respectively.

We argue that objective meaning is a *shared* meaning concerning the objectified elements at a particular moment in the socio-cultural development of the domain. The shared meaning has an external manifestation outside the heads of the individual actors. For example, the context model in Figure 3 may be regarded as the objective meaning about the spatial structure and extension of the activity domain.

The purpose of the domain construction strategy is thus to construct shared, or objective, meaning which simultaneously is transformed into subjective senses in such a way that coordinated action is possible. In the first phase, exploration, a ‘seed’ of objective meaning is constructed in a small group of actors. In the next phase, trust boosting, this seed is diffused and transformed to the ensemble of actors. The viability of the meaning is probed in one particular elaboration of the activity, such as a single project. In the final phase, expansion, the objective meaning is established through repeated elaborations of the activity, for example, in several projects.

#### **A COMPARISON BETWEEN THE ADT AND AT (reflection)**

In this section we will discuss the relation between the ADT and the cultural-historical variant of AT, also known as CHAT (e.g. Engeström, 1987; Nardi, 1996; Engeström, 1999). As pointed out by Harris (this publication), this variant differs in essential aspects from the original AT, which was developed by psychologists in the former Soviet Union from the beginning of the 1930s. Other variants rooted in the original AT, such as the systemic-structural theory of activity (e.g. Bedny et al., 2000) may be more apt for informing practical applications. A comparison with this variant has not been performed so far.

#### **Stabilization**

The stabilization modality is present in both theories. In AT the actions can be transformed into operations which may be seen as an objectivation process. Also, rules, norms, conventions, etc., which mediate between the subject and the community in AT, would be derived from this modality.

#### **Contextualization**

Contextuality is salient in both theories. In AT cognitive processes are not independent and unchanging ‘abilities’ - “they are processes occurring in concrete, practical activity

and are formed within the limits of this activity” (Kuutti, 1996:33).

#### **Transition**

The transition modality in ADT does not appear to be emphasized in AT. The element “division of labor” could possibly be elaborated to include this aspect if the mapping and interpretation between the activities are taken into account.

A transition element is traceable in the discussion of “boundary objects” (Bertelsen, 1999). These are objects that can be interpreted differently by different groups (say users and designers) but still maintain some commonly understood feature which tie different praxes together.

#### **Spatialization and temporalization**

In AT actions are composed of operations and may participate in activities with different motives and objects. Thus, AT has a strong temporal orientation. However, the objectified outcome of spatialization in terms of structures, relations between phenomenon, characterization of phenomenon, etc. is not stressed.

Moreover, there is no clear indication of interdependency between spatialization and temporalization in the AT. Engeström (1999) touches on this when he classifies the mediating artifacts into what, how, why and where types. In ADT, this interdependence is emphasized. In the Framework the context model (spatialization) and the coordination model (temporalization) are strongly interdependent.

#### **Mediation**

In AT the concept of mediation plays a key role. Human activity is directed towards an object and mediated by signs (semiotic activity) and tools (instrumental activity) (Engeström, 1999:23 ff.). However, the distinction between semiotic and instrumental activity is problematic (Bødker & Bøgh Andersen, 2004). Even if these two types of activities differ with respect to their material and social effects they should not be regarded as belonging to different realms of reality (ibid. 6). Bødker & Bøgh Andersen propose a model in which the semiotic triangle is combined with the AT triangle into a combined model where “... instrumental and semiotic activities are variants of the same pattern but with different kinds of emphasis. This predicts a smooth transition between the two.” (ibid.10).

This is also the approach taken in ADT. Signs are considered as fundamental mediating elements which comprises both semiotic and instrumental mediation: ““Signs (...) are particular, material things; and (...) any item of nature, technology or consumption can become a sign, acquiring in the process a meaning that goes beyond its given particularity. A sign does not simply exist as part of a reality - it reflects and refracts another reality (...).” (Volo\_inov, 1929/1986:10). Leiman has also pointed out the need for an articulation of the sign concept in AT (Leiman, 1999).

## Interaction

The position taken in ADT is that representations are formed in dialectical interaction (Bickhard & Terveen, 1995). Representations are seen as *interaction potentials* (ibid.), regardless of whether that potential is semiotic or instrumental in character. These potentials may be regarded as *affordances*: “The *affordances* of the environment are what it *offers* the animal, what it *provides* or *furnishes*, either for good or ill.” (Gibson, 1986:127). However, the affordances offered to humans are nested in the cultural-historical activity: “Direct perception of the affordances (...) is based on the perceiving observer’s *inclusion* in adequate *societal forms of praxis*.” (Bærentsen & Trettvik, 2002:8). This means that intersubjectivity is considered a prerequisite for individual understanding. Moreover, the interaction perspective may be further articulated by the experiential learning model (Kolb, 1984).

In AT, the object, which can be material or intangible, is shared for manipulation into the outcome. Engeström describes an ‘expansive learning’ cycle in AT consisting of seven steps which has an experiential learning touch (Engeström, 1999b). However, interactivity and experiential learning do not appear to have a central position in AT.

## Communication

Communication is emphasized in the ADT. This is mainly due to its focus on coordination. Language is not only used for describing and expressing the world but also for acting in it (Austin, 1962; Searle, 1969). Obviously, communicative acts such as directives and commitments are powerful coordinating mechanisms. In AT, communication does not seem to play a significant role.

## Practical impacts

The ADT has clearly demonstrated its constructive capabilities in designing ISs for demanding practical tasks such as coordinating development tasks in the telecommunication area. The main impact of AT seems to be analytical.

However, Korpela et al. (2004) has used a modified version of cultural-historical AT called Activity Analysis and Development (ActAD) for constructive purposes. Like ADT, ActAD “zooms out” from the IT-artifact to include work activities in organizational, economical, social, cultural and political contexts. From this, methodological guidelines for IS design are derived. So far, these are in a “prototyping phase” and have not been tried in practice.

A major difference between ADT and ActAD seems to be the type of design process. While ADT suggests an ongoing iterative process ActAD is based on a linear process. The first phase is to define a work activity model which is then translated into process diagram which is further elaborated using UML. There is no indication of a feedback from the later phases to the work activity model. Thus, the interaction modality in the sense of ADT is not salient in ActAD. This modality has been absolutely decisive for the practical impacts achieved in applying the ADT.

## General reflection

It is evident that ADT and AT have many features in common. Both acknowledge the dialectics between the individual cognition and the social praxis where the individual acts. It appears that the modalities of spatialization, transition and interactivity are more emphasized in ADT than in AT. On the other hand temporalization is more accentuated in AT. From an ADT perspective this creates an unbalance between the modalities which may conceal vital interdependencies between the modalities.

AT has strong roots in the Soviet cultural-historical psychology founded by Vygotsky, Leont’ev and Luria. The key problem for Vygotsky was “the establishment of a cultural-historical science about humans.” (Hedegaard *et al.*, 1999:13). It appears that the overall trajectory of CHAT has been from a focus on the individual towards praxis. For example, the notions of operations, conscious and unconscious, zone of proximal development, etc. indicate a focus on the individual. Engeström added the elements of rules, community, division of labor, etc., (Engeström, 1987). Recently there has been a surging interest in the interrelationships between activity systems (e.g. Virkkunen, 2004). However, it seems that the main influence of AT in the IS community has been to re-focus the Unit of Analysis from an individual, cognitive oriented one to a social oriented activity (Virkkunen & Kuutti, 2000; Kuutti, 1996).

The trajectory of the ADT is rather the opposite. It began in industrial settings as an attempt to coordinate various practices, such as software and hardware design. Thus, the focus was on the praxis rather than the individual. This brought into focus problems like how to structure a practice, how to manage the transitions between practices, etc. Individual cognition became more in focus with the insight that semiotic problems, such as achieving shared meaning, are of major concern.

It can also be noted that the concept of contradiction, which is fundamental to AT, is not emphasized in ADT in its present appearance. Contradictions have not been operationalized in the Framework so far. In the future dialectical interaction between theory and practice, it might be fruitful to exploit the contradictions between the activity modalities. This will however be depending on the practical impacts of including contradictions in the theory.

Thus, CHAT and ADT have moved towards the dialectical centre of activity from two different directions, much like a thesis and anti-thesis. Whether there is a viable synthesis between the theories remains to be investigated.

## TRANSFERABILITY OF THE RESULTS (reflection)

So far, the method has been applied in one target area only, that of the development practice at Ericsson. This means that the transferability of the method to other target areas has not been demonstrated. However, the method is not specific for the Ericsson organization. On the contrary, the ADT is a general theory which should be appli-

cable to any constellation of cooperating practices whether these are internal inside an organization or external between organizations. The operationalization of the theory may take different forms. For example, a practice which does not use ISs obviously will not contain that element.

### CONCLUSION (reflection)

In this paper, we have described the Activity Domain Theory and its relation to the cultural-historical Activity Theory. These theories share many features and are grounded in the same philosophical perspective. However, there are significant differences. The ADT emphasizes interactivity, signs and domain transition which are more peripheral in AT. Also, the ADT has proven capable of influencing the design of ISs supporting the coordination of exceptionally complex system development tasks. To the best of our knowledge, there is no similar achievement reported for AT.

### REFERENCES

- Austin JL (1962) *How to do things with words*, Oxford, University Press.
- Bærentsen K, Trettvik J (2002). An Activity Theory Approach to Affordance, the *Second Nordic Conference on Human-Computer Interaction*. NORDICHI 2002. Tradition and Transcendence. Aarhus, Denmark, October 19-23.
- Bedny G, Seglin M, Meister D (2000) Activity Theory: history, research and application, *Theoretical Issues in Ergonomic Sciences*, Vol. 1, No. 2, pp. 168 – 206.
- Bedny G, Harris S (2004) The Systemic-Structural Theory of Activity: Applications to the Study of Human Work, forthcoming in *Mind, Culture and Activity*.
- Bedny G, Karwowski W (2004) A functional model of the human orienting activity, *Theoretical Issues in Ergonomic Sciences*, Vol. 5, No. 4, pp. 255 - 274.
- Bedny G, Karwowski W (2004b) Activity theory as a basis for the study of work, *Ergonomics*, Vol. 47, No. 5, pp. 134 - 153.
- Bedny G, Karwowski W (2004c) Meaning and sense in activity theory and their role in study of human performance, *Ergonomia IJE&HF*, Vol. 26, No. 2, pp. 121 – 140.
- Benbasat I, Zmud, R (2003) The Identity Crisis within the IS Discipline: Defining and Communicating the Discipline's Core Properties, *MIS Quarterly* (27:2), pp. 183 - 193.
- Bertelsen O (1999) Mediation and Heterogeneity in Design. In *Social Thinking - Software Practice* (Dittrich Y, Floyd C, Jayaratna N, Kensing F, Klischewski R, Eds.), Dagstuhl-Seminar-Report; 250, 05.09.1999 - 10.09.1999 (99361). Wadern: IBFI gem. GmbH. pp.16-20.
- Bickhard MH, Terveen L (1995) *Foundational Issues in Artificial Intelligence and Cognitive Science: Impasse and Solution*. Elsevier Scientific, Amsterdam.
- Bødker S, Bøgh Andersen P (2004) Complex Mediation (draft).
- Engeström, Y (1987) *Learning by expanding: An activity-theoretical approach to developmental research*. Helsinki: Orienta-Konsultit.
- Engeström Y (1999) Activity theory and individual and social transformation. In *Perspectives on Activity Theory* (Engeström Y, Miettinen R, Punamäki RL, Eds.), Cambridge University Press, Cambridge UK. pp. 19 - 38.
- Engeström Y (1999b) Innovative learning in work teams: Analyzing cycles of knowledge creations in practice. In *Perspectives on Activity Theory* (Engeström Y, Miettinen R, Punamäki RL, Eds.), Cambridge University Press, Cambridge UK, pp. 377 - 404.
- Gandhi M, Robertson E L (1992) A Specification-based Data Model. *Indiana University Computer Science Department Technical Report TR344*, Indiana University, Indiana. Available at <http://www.cs.indiana.edu/ftp/techreports/index.html> (accessed Oct 2002).
- Gibson JJ (1986) *The Ecological Approach to Visual Perception*. Lawrence Erlbaum, Hillsdale.
- Goldkuhl G, Röstlinger A (2002) Towards an integral understanding of organisations and information systems: Convergence of three theories, accepted to the 5<sup>th</sup> *International Workshop on Organisational Semiotics*, Delft.
- Harris S (2004) Systemic-Structural Activity Analysis of Video Data, this publication.
- Hedegaard M, Chaiklin S, Juul Jensen U (1999). Activity Theory and Social Practice: An Introduction. In *Activity Theory and Social Practice: Cultural-Historical Approaches* (Seth Chaiklin, Mariane Hedegaard and Uffe Juul Jensen, Eds.), Aarhus University Press, Aarhus.
- Kolb D A (1984) *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall, Englewood Cliffs, New Jersey.
- Korpela M (2004) ActAD in Information Systems Analysis and Design, this publication.
- Kosik K (1976) *Dialectics of the concrete. A Study on Problems of Man and World*, Reidel, Dordrecht.
- Kuutti K (1996) Activity theory as a potential framework for human-computer interaction research, in Nardi B A (Ed, 1996) *Context and consciousness*. Activity theory and human-computer interaction, MIT Press, Cambridge.

- Leiman M (1999) The concept of sign in the work of Vygotsky, Winnicott, and Bakhtin: Further integration of object relations theory and activity theory. In *Perspectives on Activity Theory* (Engeström Y, Miettinen R, Punamäki RL, Eds.), Cambridge University Press, Cambridge UK, pp. 419 - 434.
- Mingers J (2001) Embodying information systems: the contribution of phenomenology, *Information and Organization*, Vol. 11, pp. 103-128.
- March J G, Simon H A (1958) *Organizations*. Second edition, Blackwell Publishers, Cambridge, Massachusetts, USA.
- Marx, K (1867/1967) *Capital, I*. New York: International Publishers. Available at (Chapter 7): <http://www.ecn.bris.ac.uk/het/marx/cap1/index.htm> (accessed Sept 2004)
- Matrix-One (2004) <http://www.matrixone.com/> (accessed Oct. 2004).
- Nardi B (1996) *Context and Consciousness: Activity Theory and Human-Computer Interaction*, (Bonnie A. Nardi, Ed.), Cambridge, MA: MIT Press.
- Rumbaugh J, Blaha M, Premerlani W, Eddy F, Lorensen W (1991) *Object-Oriented Modeling and Design*, Prentice-Hall International, Inc., New Jersey.
- Schatzki TR (2001) Introduction: Practice theory. In *The practice turn in contemporary theory* (Schatzki TR, Knorr Cetina K, von Savigny E, Eds.), Routledge, London.
- Searle J R (1969) *Speech acts. An essay in the philosophy of language*, Cambridge University Press, London
- Searle J R (1995) *The construction of social reality*. London: Allen Lane.
- Taxén L (2003) *A Framework for the Coordination of Complex Systems' Development*. Dissertation No. 800. Linköping University, Dep. of Computer & Information Science, 2003. Available at [http://www.ep.liu.se/diss/science\\_technology/08/00/index.html](http://www.ep.liu.se/diss/science_technology/08/00/index.html) (accessed Sept 2004).
- Taxén L (2004) Articulating Coordination of Human Activity - the Activity Domain Theory. In *Proceedings of the 2<sup>nd</sup> International workshop on Action in Language, Organisations and Information Systems (ALOIS-2004)*, Linköping University. Available at <http://www.vits.org/konferenser/alois2004/proceedings.asp> (accessed March 2004).
- Webster's 1913 Dictionary. Available at [http://humanities.uchicago.edu/orgs/ARTEFL/forms\\_unrest/webster\\_form.html](http://humanities.uchicago.edu/orgs/ARTEFL/forms_unrest/webster_form.html) (accessed Sept 2004).
- Wicks AC, Freeman RE (1998) Organization Studies and the New Pragmatism: Positivism, Anti-positivism, and the Search for Ethics. *Organization Science*, Vol. 9. No. 2. March-April 1998, pp. 123-140.
- Virkkunen J, Kuutti K (2000) Understanding organizational learning by focusing on "activity systems", *Acting., Mgmt. & Info. Tech.* 10 (2000), pp. 291-319.
- Virkkunen J (2004) Hybrid agency in co-configuration work, *3<sup>rd</sup> Nordic conference on Cultural and Activity Research*, 3-5 September, Copenhagen, Denmark.
- Volo\_inov, V N (1929/1986) *Marxism and the Language of Philosophy*. Harvard University Press, London. (originally published in 1929).

# Narrative Transformation: Designing Work Means by Telling Stories

Bettina Törpel

IT University of Copenhagen

Rued Langgaards Vej 7

DK-2300 Copenhagen S, Denmark

+45-72185164

beetee@itu.dk

## ABSTRACT

Narrative Transformation is a method for clarifying the purposes and functionality of IT work means in groups of designer-users with self-selected membership. This includes that interests are clarified and pursued. Narrative Transformation has been developed specifically for groups in which the members themselves are largely responsible for setting up their own work situation, organization and work means (e. g. associations of freelancers, communities of researchers). As part of the interest guided development process the participants identify things they have taken for granted; they explore if it is justified to take them as »matters of course«; if not, they develop more appropriate notions.

The procedure of Narrative Transformation comprises diverse steps. Every participant individually writes *episodes* describing events that occurred to them, and that they think highlight important aspects of the work setting that is to be changed by IT means. Then the participants collectively »search« their episodes for »*matters of course*«. The »*search criteria*« are developed as part of this process. The participants explore how appropriate the assumptions in the episodes are and develop new, more appropriate notions, if necessary. This corresponds with forming and discussing hypotheses regarding: (1) characteristics of the participants' settings, conditions, situations and constellations; (2) specific relations between possibilities and restrictions for fulfillment that »are contained« in them; and (3) changes - especially in terms of work means and practices - that are beneficial. The participants examine their hypotheses in their every-day life/work settings, and their according experiences are fed back into the Narrative Transformation group process, where they can serve as impulses for further processes of reflection and change, possibly using Narrative Transformation.

Narrative Transformation is based on notions of Activity Theory, and here especially the research direction of Critical Psychology, such as the specific notions of agency, (inter-) subjectivity, purpose, meaning, objectification and appropriation. The procedure of Narrative Transformation itself is an elaboration of the procedure of Memory Work.

## Keywords

Narrative Transformation, IT design, »inside« standpoint of the involved subjects, (inter-) subjectivity, designer-users, fragmented work environments, work means, circles, purpose, meaning, functionality, objectification, appropriation, Critical Psychology, Memory Work.

## INTRODUCTION AND OVERVIEW

In this contribution the method of Narrative Transformation is described, explained and exemplified. After this introduction a *practical description* is provided in part I and an *academic reflection* is presented in part II. The practical description contains a *step-by-step guide* and a *practical example* of the use of Narrative Transformation, along with practical information regarding the *purpose*, *outcomes* and *products* (intermediary and final) of Narrative Transformation, specifics of the *systems* to be developed, *development processes*, *phases* and *target groups/settings* for which Narrative Transformation is recommended, and the *actors* who perform Narrative Transformation, with their *competencies* and *propensities*. The academic reflection comprises sections addressing *practical experiences*, *utility* and *evaluation* of Narrative Transformation, those *concepts* from Activity Theory - especially as further developed by Critical Psychology - that are relevant for Narrative Transformation, and some of the *historical background* of Narrative Transformation.

The method of Narrative Transformation described here was developed by the author and some of her co-workers (cf. e. g. Törpel 2003, Törpel & Poschen 2002). In developing the method they aimed specifically at providing support for clarifying, articulating and pursuing *interests* as part of developing functionality supporting individual and cooperative work from an »inside perspective« of the members of *fragmented work environments* (see below). This involved the participants working on their own notions and assumptions; and working on what they took *for granted* but needed to explore and revise if IT work means useful and beneficial to their own interests were to be developed.

Narrative Transformation was inspired by the method of Memory Work (cf. Haubenreisser & Stöckmann 1993, Haug 1999). Memory Work is one of the attempts at translating the abstract action research programme of Critical Psychology (cf. Holzkamp 1983, chapter 9) into

concrete measures. Narrative Transformation is a procedure based on *episodes* written by the participants. In their episodes the individual participants describe encounters relevant with respect to their work setting, their individual and cooperative work, and to their work means. The participants analyze their episodes in a collective process and, if necessary, draw consequences, e. g. in terms of their notions, their practices and their work means.

The term "Narrative Transformation" suggests two kinds of association. On the one hand, the participants using the procedure write, discuss, re-conceptualize and *re-write their own episodes and notions* as expressed in the episodes: they re-shape (transform) their episodes and notions (narrations). On the other hand, their joint work with the episodes is geared toward *re-shaping relevant phenomena in the participants' working lives* to which the episodes refer, such as technology/functionality, work and organization. Work/life phenomena are changed (transformed) by working on the episodes (narratively).

The kind of method Narrative Transformation exemplifies involves aspects of both *reflection* and active *change/improvement*; it supports an interplay between exchange, explication, discussion, use of fantasy and planning on the one hand - and practical experimentation and modification on the other.

The developers of Narrative Transformation have, more specifically, pursued a question that can be framed as follows: What kind of method do people need when they are in a work setting (also: situation, constellation etc.) where work can only be accomplished when the members themselves clarify what kind of IT work means they should use in which ways for their work? Over the past few years, the developers of Narrative Transformation have gained practical experience with procedures in such self-determined work settings. As a part of this action research work the original research/practice question has been differentiated, e. g. regarding target groups of participants, work settings and ways of successfully acting in such settings. Currently, the target group is mostly designer-users that work in fragmented work environments (see below).

## **PART I: PRACTICAL DESCRIPTION OF NARRATIVE TRANSFORMATION**

### **Purpose of the method of Narrative Transformation**

The motivation for such methods as Narrative Transformation is to develop computer applications plus related practices of use and further development »from below« or »from the inside« of those who will later work with their own products and/or alterations. The aim is for the outcomes to be appropriate and beneficial rather than detrimental from the perspectives of the people involved. This includes the aim that participants should be supported in an interplay of forming, identifying and using »the right« *circles* for achieving what they aim for - and (further) developing, finding and making use of work means that help achieve what they aim for. They *clarify* their *purposes, interests* and *matters of course* and *change/improve* them, by practical steps, in a way that the

work/life of the participants improves from their own perspectives.

### **Applying the method of Narrative Transformation - a step-by-step guide**

In order to get started with the method, a Narrative Transformation group requires a *common objective* - some phenomenon that the group members experience as limiting their possibilities, and that they wish to overcome. Getting started with, or getting the idea of using Narrative Transformation, also requires that the people who eventually form the group learn something about Narrative Transformation and decide to use this procedure as a support for their change/improvement processes. The members must perceive their interests as converging to the extent that they want to perform the Narrative Transformation process together. Narrative Transformation hence allows for and requires that *circles* for developing one's own work means and for practicing Narrative Transformation are formed on a voluntary basis between persons who trust each other. This is an expression of the necessity to tackle challenges related to designing the functionality of IT work means by using the »inside« subjective perspectives of the people who will be affected by their own results as an essential resource in *fragmented work environments* (see below).

The practical steps of Narrative Transformation are:

- A Narrative Transformation group is formed.
- In this group, a tentative topic of the joint Narrative Transformation process is formulated. During the actual process this topic may have to be re-formulated according to the preliminary results of the process.
- As an ongoing process in Narrative Transformation, the participants do research, in the sense that they acknowledge results from relevant fields/efforts, form hypotheses and reshape them by confronting them with the reality as they experience it.
- In order to track down specific symptomatic areas and issues for reflection and improvement, all participants write episodes describing events related to the agreed-upon topic they find relevant and promising for figuring out more. Each episode is about one page in length, is written in the third person, has a plot with a beginning and an end and describes something that actually happened to its author.
- The collective analyses of the episodes comprise the analysis of the individual episodes as well as comparisons between episodes according to dimensions generated by the group itself, such as: plot of the episode, construction of the actors, contradictions, clichés, feelings, implicit theories and use/development/modification of work means, creation and use of further resources, aspects of the situation in need of improvement and »good practices«.
- Sentence by sentence, the participants »scan« and discuss the paragraphs of each episode according to the analysis criteria as »search guides« in a search for notions that the participants suppose should be

discussed, clarified and possibly replaced. The group examines which content of an episode is relevant for each generated dimension. The steps and results are written down and archived together with the episodes and analysis dimensions.

- Most likely, the dimensions of the analysis will change dynamically as part of the analysis, and the changes and reasons for them are then written down. Additions, such as new comments, episodes, dimensions and results are ongoingly elaborated, and they are also archived.
- When the participants agree that their understanding of phenomena and situations worth improving is sufficient, they generate ideas, projects and plans for shaping them and constructively acting in the face of them. This can involve a wide variety of approaches, e. g. designing artifacts, inventing negotiation strategies, establishing and technically supporting new circles (see also the practical example below).
- All the newly created or harnessed possibilities are, as part of the method, soon put to the practical test, usually outside the Narrative Transformation group meetings.
- This yields specific results, depending on the specifics of the group, topic, process, situation, constellation, etc. which are discussed and evaluated in subsequent meetings.

The Narrative Transformation process can comprise »cycles« of a set of the steps that have been described so far, plus maybe steps specific for the group, system, process etc. Results cannot be directly/immediately »read out« of or »derived« from the episodes. They evolve in a dynamically intertwined process of the specifics of the tackled issues, individual recollection, joint (re-) construction and group process phenomena.

Narrative Transformation is not necessarily a self-contained procedure. Approaches and procedures from diverse fields can - if expected to be beneficial - be integrated in the Narrative Transformation process. Other methods, such as scenario-based design (e. g. Carroll 2000) or other methods described in the current proceedings can, for example, be incorporated.

#### **Outcomes, documents, tools and other artifacts used and produced as part of applying the method of Narrative Transformation**

The main outcome of successfully applying Narrative Transformation is *clarification* - as to which projected/actual IT work means and other relevant conditions can, in which ways, be beneficial for the people who are involved. The "other relevant conditions" include such phenomena as practices related to work means and the integration of the work means into social and technical infrastructures.

Beyond clarification, Narrative Transformation can also yield practical and tangible results: technologies, practices, circles, circumstances, assumptions, aims, etc. By using the method, these outcomes are devised in such a way that

they are judged suitable for the participants according to criteria that they themselves have explicated and elaborated. Depending on the kind of work means that are to be developed - and also depending on the different settings, groups, projects, constellations, situations, participants' purposes, agendas, problems, reasons (individual and collective) etc. involved - the concrete outcomes of applying Narrative Transformation can be diverse.

Narrative Transformation can be regarded as a meta-method that in the course of being applied can yield diverse products and outcomes (intermediate and final): artifacts, practices or »next steps« specific for the constellation/situation in question - and these are related to the (further) development, introduction and use of IT work means. Examples of outcomes are diverse running computer applications, practices, agreements, papers, the feeling of individual and/or collective clarification/improvement, and many more.

Outcomes that are intermediately produced and further used - and hence serve as tools - in the process of Narrative Transformation are

- a topic and focus for the Narrative Transformation process,
- episodes, written or printed on paper, in which the participants describe events they experienced and that are related to the setting and topic of the process,
- wall charts capturing important points of the discussions in the meetings,
- lists with descriptions and explanations of analysis dimensions/criteria for the collective analysis of the individual episodes (e. g. as wall charts, on paper or electronically),
- minutes of the meetings (e. g. on paper or electronically),
- artifacts and phenomena that participants assumed or experienced to help capture or improve work, circumstances, functionality, practices, problem descriptions, objectives, assumptions, matters of course and so forth.

The documents, tools and other artifacts used in concrete processes of Narrative Transformation may be very diverse. The episodes written by the participants are usually the initial documents, and they have the form of handwritten sheets of paper, computer prints or electronic computer files, maybe represented by (maybe cooperative) computer tools, e. g. if Narrative Transformation should be practiced in electronically supported forms. It also makes sense to systematize, archive and make accessible the episodes, discussion notes and wall charts, any kinds of results and minutes from the Narrative Transformation sessions, for future use in general and for the use in long-term Narrative Transformation processes.

Currently, if and how Narrative Transformation processes could be electronically supported so that people from different locations might participate is an open question. One idea for structuring such an electronically supported

process is to use episodes and topics as »focal points« that participants can annotate, comment and discuss in the form of threads, as in newsgroups. Electronic tools that might be suitable are e. g. Shared Workspace Systems such as the BSCW system (Bentley et al. 1997, BSCW 2004), combined with chat tools and email; or document editing, linking and visualizing tools (exemplified by the cooperation infrastructure KOIN, cf. Kleinen & Herczeg 2003). Some initial attempts have been made to perform Narrative Transformation over distances, using both of the tools mentioned.

### **The kinds of development processes, phases and systems for which Narrative Transformation is suitable**

The application of the method of Narrative Transformation is not bound to any particular kind of *development process* (e. g. in-house development, design of a new product, re-design, participatory design) or development setting. Nor is it restricted to any particular *phase of a design process*. Narrative transformation may be found useful e. g. in early clarification re. the work situation's improvement possibilities, early requirements analysis, planning, realization, interface evaluation, introduction, integration of old and new work means and practices, and re-arrangements in use. Furthermore, the application of the method of Narrative Transformation is not bound to the development of any particular *kind of system* (such as office automation systems, CAD/CAM systems, mobile systems, groupware systems).

Narrative Transformation can be useful when reflection, re-thinking, re-design and re-arrangement seems necessary to the potential participants. Narrative Transformation can be meaningfully applied and help steer the process in a beneficial direction when the constellation, situation and projected system leave room on the one hand for reflection/discussion and, on the other for practical experimentation, modification and improvement. The criterion for applying Narrative Transformation in terms of the kind of process, phases and systems is that the participants are open towards, and see a reason to engage in the involved interplay of reflection and change and engage in the according steps.

### **The primary target settings and groups for Narrative Transformation**

The method of Narrative Transformation is an attempt to put to practice a far-reaching form of *participation* in the form of *self-inclusion*. According to the experiences of the author this self-inclusive form of participation is especially relevant for a selection of such »new« forms of work that are characterized by:

- short-term contracts and project tasks,
- geographical distribution, and hence commuting, traveling and/or communicating and collaborating mediated by tools that help bridge spatial and temporal distances,
- flexibility and extreme short-term modulations, e. g. in terms of business objectives, groups of collaborators and income,

- high levels of self-determination combined with high levels of dependency,
- unclear and delicate employment status (for example, oscillating between aspects of being employed, self-employed, employer)
- a status beyond representation by unions or traditional entrepreneur associations.

This selection of new forms of work, for the purposes of the present text, can be most accurately referred to as "fragmented work environments" (cf. e. g. Törpel et al. 2002, Törpel 2004; one such setting has been extensively described in Törpel et al. 2003). Other labels that largely address the same forms of work but highlight other characteristics and implications are "virtual organizations", "virtual corporations", "network organizations" etc. (cf. e. g. Powell 1989, Dawidow & Malone 1992, Mowshowitz 1997) or "self-employed labor" (cf. Voß & Pongratz 1998, Törpel 2000).

The kind of method that is here exemplified by Narrative Transformation further more specifically aims at people

- who need *new work means*,
- whose *work/life will change* when the projected work means are introduced,
- whose *work means and related practices are essential for their success* and fulfillment in work,
- who are in a situation where *taking active part in shaping their own work is an essential success factor*,
- whose *professional success relies on actively forming their own professional circles* - project groups, professional networks, circles for developing work means etc. - beyond organizational boundaries (this differs from work settings where the members get their central circles provided as part of the work organization they once joined, and where specialized organizational units are responsible for making choices of work means and for suggesting practices related to the work means in use),
- who need and want to *explore, understand and improve the relationship and interplay of their work means, work circumstances, wishes, assumptions and matters of course* as part of getting a realistic picture of viable options,
- who do this together, *for and by themselves* and do not have someone else do it for them,
- who do this following their own *subjective perspectives, reasons and feelings as starting points and success indicators*,
- who want to have *work means and practices that are beneficial* for them, and who want to explore the issue of beneficiality in their own work means and use practices of them in their specific settings,
- who *explicate and practically experiment* with their practices, means, assumptions and circumstances with the *aim of improving* them,

- who have decided to engage in this kind of process, aware of the fact that it has *political implications* (e. g. Suchman 1994, 1995) with possible direct personal consequences.

The target group comprises therefore neither developers nor recipients of IT work means but *partial agents* who rely on largely by themselves setting up their own work places, professional structures and work infrastructures in a very self-determined way, beyond pre-formed workplaces, already existing organizations and planfully set up and/or grown infrastructures. In this way Narrative Transformation differs from most approaches in Participatory Design (e. g. Bjercknes et al. 1987, Greenbaum & Kyng 1991, Schuler & Namioka, 1993, Kuhn & Muller 1993). In much Participatory Design, it is usually assumed that the distinguishable groups of professional designers and recipients of technology are working in organizations with management and immediate work, where there is a clear distinction of »capital« and »labor«.

It is also possible that Narrative Transformation can be suitable for use in other kinds of settings, such as:

- in product development as part of the business strategy of companies who develop computer applications for a segment of the market that first needs to be explored;
- in emergent development teams, possibly together with potential or actual customers;
- in distributed development teams as part of team establishment or consolidation strategies in parallel to productive work;
- in collaborations between users and developers in in-house development; and
- in groups, circles, alliances and exchange relationships of advanced users who have become tailors of applications they use themselves, or of customers' or colleagues' applications.

#### **The actors who perform the method of Narrative Transformation and their competencies and propensities (regarding Activity Theory and otherwise)**

The actors who perform Narrative Transformation are the people who contribute to (mostly their own) IT work means and the practices and circumstances corresponding with the work means (e. g. users, consultants, designers, usability specialists). The method can best be made use of in groups, and, more specifically, Narrative Transformation matches with these groups' needs best when their members (i. e. participants)

- experience high levels of mutual trust,
- perceive the purposes, concerns and objectives for their use of Narrative Transformation as highly relevant and shared,
- perceive their interests as highly converging and
- have themselves experienced those problems and limits of their work situation that all participants take seriously enough to consider it worth putting effort and time into working on them by means of

developing new work means aided by Narrative Transformation.

For practicing Narrative Transformation no extensive prior knowledge in Critical Psychology or Activity Theory is required. Reversely, a practical, and not necessarily academic, knowledge of Critical Psychology (especially Subjektwissenschaft / subject research, cf. Holzkamp 1983, chapter 9, see also below) can be gained by engaging in Narrative Transformation. It may also be suggested that those people whose lifestyles are in accordance with the kind of theory Critical Psychology is (see below) may find it especially worthwhile to engage with Narrative Transformation.

In the example of the geographically distributed research team (see below) only one person had a background in Critical Psychology and Activity Theory; the other participating colleagues had not had direct prior exposure to these research traditions, some of them had indirect experiences by discussing colleagues' research (concepts, methods, paper drafts etc.).

#### **An illustrative practical example of the use of the method of Narrative Transformation**

The work setting in the following example is a multidisciplinary research group at a research institute. When the group became the target for its own research and development efforts in Computer Supported Cooperative Work it had been in existence for only a few months, and comprised six members: two senior researchers, one junior researcher, two research assistants and an associated PhD candidate. A particular feature of the working group was its geographical distribution: the sites of the cooperation were five different towns in a middle European country and a town in another European country; except for one member the research institute was located outside members' places of residence. At the time when the group was established, a geographically distributed research group was unusual within the research institute.

The research group set itself the objective of developing viable cooperative research practices and, correspondingly, of finding suitable IT means for cooperation support. This objective suggested a number of practical problems to be addressed, for instance:

- how information would be exchanged as part of the day-to-day work,
- how work would be divided between members,
- how the group would deal with the different research approaches of its members so that a multidisciplinary group could become an interdisciplinary research cooperation,
- which research activities would be performed and which research topics, questions, concepts and methods would be used,
- how to balance funded research projects and individual and group research topics,
- how modes of working involving transparency and accountability would be established,

- which results would be produced, using which methods,
- how the requirements of the geographic distribution of the working group could be met, and
- which work means, including IT means, would jointly be used and in which ways.

The Narrative Transformation process began in spring 2001 and lasted for 18 months. Two working group members only participated in the initial meeting for this effort and it turned out that they did not have time for further participation. Therefore the process to be described had mostly four participants. The author was one of the participants.

The initial episodes of the four participants addressed problems such as:

- specific difficulties in maintaining an overview and setting the right priorities when living and working in different places that are far apart from each other,
- the technological and organizational possibilities and limits for close, active, productive and enjoyable collaboration in research project work, for writing research papers and for individual qualification,
- the difficulty to be reliable and accountable as a person and as a research group within a larger network of research groups and
- the possibilities and limits for exchange and collaborative research within the research group and the establishment of circles for this within and beyond the institutionalized research group

(cf. the full episodes in the appendix).

Over the whole period of using Narrative Transformation, the participants in the process developed and used such analysis dimensions for the episodes as: plot of the episode, construction of the actors, contradictions, clichés, feelings, implicit theories, development, modification and use of work means, creation and use of further resources, aspects of the situation in need of improvement and »good practices«.

The course of the development and clarification process of jointly developing cooperative practices and IT cooperation support, using Narrative Transformation can be described in three phases: 1. initial expectations, 2. unexpected discoveries and 3. active steps shaping the situation. This trajectory will be described in the remainder of this section.

Using their initial episodes, the participants of the process found out that they had specific shared initial expectations relevant for setting up cooperative research practices and corresponding IT cooperation support, e. g.: they had strong inclinations toward

- far-reaching participation of all research group members,
- mutual exchange, explication and negotiation and feedback,
- mutual accountability,

- carefully taking account of their geographical distribution and
- concern, curiosity, and enthusiasm for collaborative research work.

In the course of the every-day work of the group, and through thorough reflection, experimental design and especially by using the method of Narrative Transformation, it was discovered that the original objective of establishing cooperative practices in the working group and at the same time as searching for appropriate IT work means and use practices, did not apply. Various initially unexpected discoveries could eventually be expressed in terms of a polarity between two perspectives on the cooperation within the research group. This polarity can be characterized by a set of oppositions:

- participatory vs. hierarchical approach,
- exchange, explication, negotiation and feedback vs. avoiding communication about the joint research and cooperation,
- mutual accountability vs. directives from above,
- carefully taking into account distributed work vs. ignoring distributed work and putting up with possible exclusions of currently physically absent persons and
- concern, curiosity and enthusiasm in the collaborative research work vs. minimalist fulfillment of duties for project deliverables.

In order to cope with the newly-identified situation of conflicting expectations the participants took active steps to shape the situation. The main approach/result was the establishment of new research circles both within and beyond the research group (and the research institution). The circles that were established were for instance: groups for discussing and elaborating texts, theses, publications, concepts, methods, research questions, personal research projects, work techniques and a colloquium with national and international guests. These new constellations beyond the institutionalized research group met many of the participants' expectations regarding exchange and cooperation that had originally been geared toward the research group. For preparing, coordinating and doing the every-day project work for the institutionalized research group, additional meetings of subgroups of the institutionalized research group took place outside the research group. The participants felt that otherwise it would not have been possible to get enough research work done within the boundaries of the institutionalized research group, either in terms of basic project chores or genuine insight-providing research. The additional research project-task related meetings involved clarification, agreements, measures for setting up mutually accessible information and for warranting mutual accountability, etc. The participants perceived the cooperation here as enjoyable, interesting and dedicated. The official institutionalized research group lost significance while the four participants increasingly got plenty of opportunity to engage in interdisciplinary distributed cooperative research work in the new circles they had established.

The members of the institutionalized research group gained, one important result from the research: the discovery that objectives and interests that could not be matched within the group could be put into effect through a far-reaching differentiation, expansion and separation of spheres. Clarifying and pursuing the participants' then present objectives and interests resulted in new structures and practices that could not be integrated in »the« one research group in the research institute.

The work means that were in use during, or as a result of, this clarification and restructuring process included: face-to-face meetings, stationary telephone, mobile telephone, email, email lists, a shared web workspace system, paper lists, calendars (paper, palm, psion, web calendar tool). If the research group had immediately looked for »the« IT support then the »wrong« work setting may have been the target setting. Scrutinizing their every-day life experiences, assumptions and interests facilitated the rearrangement of the participants' every-day work lives according to their wishes. This rearrangement involved establishing new circles with diverse needs for IT support. The adaptation and use of the shared web workspace system that became a work means in most of the new circles can serve to illustrate this: the workspace of the official research group served as a document repository; the workspace for the new project-work related meetings outside the research group was used to store, comment and modify texts, agendas, schedules, checklists (ToDo, research criteria etc.) and minutes; in most of the other circles members' texts were uploaded to the workspace and later supplemented with written feedback notes; the reading group did not require a workspace at all.

## **PART II: ACADEMIC REFLECTION ON NARRATIVE TRANSFORMATION**

### **Reflection on the practical experience with and evaluation of Narrative Transformation**

#### *Practical experiences*

So far, the only setting where an extensive Narrative Transformation process has taken place has been the distributed research cooperation setting described above. However, short (and hence necessarily rudimentary) processes of Narrative Transformation have taken place in a variety of settings, mostly workshops, lasting from a half day (e. g. workshop as part of the annual meeting of German-speaking women in science and engineering) to several days (e. g. four days as part of a project course on the development of computer applications, Software Engineering, participation and gender at the Computer Science department at the Humboldt University of Berlin, ten days in a teaching course at the Technical University of Berlin).

An intense Narrative Transformation process may take place over a period of months (18 months in the example of research cooperations above) and be quite time-consuming: in the example, the meetings took a few hours every about 3-4 weeks, and they required extensive preparation and post-processing.

As an initiator and participant the author has found the above-described Narrative Transformation process productive and enjoyable: new ideas emerged in a playful yet serious and committed way, practices and artifacts were tried, and new circles and structures were formed. These new structures, practices and circles partly complemented, partly avoided, and partly subverted those previously existing. In the case described, it turned out that Narrative Transformation in fact helped to explicate and work on assumptions and clarify many things, for example interests. It also supported the taking of practical steps beyond mere reflection.

One remarkable, possibly unsurprising, but nevertheless important experience has been the discovery that chiseling out what some of the problems, situations and constellations were all about was very delicate in terms of the power relations within the research group and beyond. The Narrative Transformation process required initial trust, but also brought about extreme closeness and trust. However, this level of closeness might not always be desired in work constellations.

So far, Narrative Transformation has not extensively and systematically been used in the development of *new products* (IT work means) from scratch; this remains for future work.

#### *Investigating and evaluating applicability and success*

The issue of how to investigate the applicability of inquiry and design methods from the standpoint of the subject, such as Narrative Transformation, needs to be further framed and explored.

In a sense, Narrative Transformation is the kind of method one might experience as useful in situations where one can describe the issues, the process and the results for others, e. g. in research papers - supporting others to evaluate how their issues are different and similar and how they would like to use the already conducted process as a source of inspiration. Although in terms of some epistemological approaches this might sound weak, in terms of interrelating practical experiences and research it may turn out to be quite powerful. The author adheres to an epistemological standpoint where the research actors' preferences and lifestyles (see below) on the one hand, and research approaches, concepts and methods on the other, are seen as closely interlinked.

The author sees the extent of clarification experienced by the participants in coming up with appropriate (mixes of) work means, practices, structures, assumptions etc. as an indicator of the applicability and suitability of methods such as Narrative Transformation as part of dealing with conditions etc. (e. g. of doing distributed research work in a delicate institutional setting and within a »fragmented work environment« that turned out to have far more ramifications than initially expected).

A suggestion for investigating and assessing the applicability of methods such as Narrative Transformation is that the participants regularly (e. g. every 3 months) do a self-assessment as to which extent the method has been instrumental in achieving (or discarding, in any case: in

dealing with) what they have tried to achieve. In many cases, it may not make much sense to make any kind of comparison between processes with and without the method - because all situations of developing and using IT work means and their embedding are so specific that there will always be commonalities and differences.

### **Narrative Transformation, Activity Theory, Critical Psychology and Memory Work - concepts and historical background**

Narrative Transformation was developed by the author and some of her colleagues in Berlin, Bonn, Sankt Augustin and Copenhagen, as part of teaching and research projects. It builds on concepts presented by proponents of Critical Psychology, a school of thought in the tradition of Activity Theory that was most influential in the 1970s and 1980s (cf. Holzkamp 1983; an introduction in English is provided by Tolman 1994). Critical Psychology, which was mainly developed at the Free University in West Berlin, can be seen as an attempt to further develop Activity Theory. In the following sub-section a brief overview is given of those Critical Psychology concepts that have been especially relevant to the development of Narrative Transformation.

#### *Critical Psychology and Activity Theory as background of Narrative Transformation*

An important notion for establishing Critical Psychology's research framework has been the notion of *present historicity* (gegenwärtige Historizität). Phenomena as they are currently present always have a history, and usually their historical reconstruction helps us to understand their features, internal organization and their relation to other phenomena (Holzkamp 1983, chapter 1). In Critical Psychology, two kinds of historically guided inquiry have been elaborated and undertaken as methodological frameworks: historical-empirical inquiries (functional-historical analysis / funktional-historische Analyse and societal-historical analysis / gesellschaftlich-historische Analyse) and topical subject-scientific empirical inquiry (subjektwissenschaftliche Aktualempirie).

The *historical-empirical inquiries* are phylogenetic and societal-historical reconstructions of the human psyche's distinctive features (ibid, chapters 2 - 7). One of the results of these phylogenetic reconstructions is that methods for inquiring about human individual phenomena that fully utilize the human potentials - and hence are most appropriate - proceed from the standpoints of the scrutinized (and hence somehow involved) individuals. The methodological framework for doing this is called *topical subject-scientific empirical inquiry* (subjektwissenschaftliche Aktualempirie, ibid, chapter 9), where research is conducted from the standpoints of the affected/involved target subjects on currently occurring phenomena *of relevance to them*. The appropriate methodological steps of *topical subject-scientific empirical inquiry* build on, and are delineated by the results of the *historical-empirical inquiries*, particularly those regarding *human agency* (Handlungsfähigkeit, see below).

Conducting research based on Critical Psychology's methodological considerations has resulted in, among other things, a system of research concepts (categories). The historical-empirical method serves to establish, substantiate and delimit research categories, e. g. by using accounts of the phylogenetic emergence of the human species. The resulting conceptual apparatus is intricate and differentiated; it captures and expresses the historical-empirical results (viz. the historically-empirically supported hypotheses), regarding events, sequences, features, affordances and relations of phenomena in organisms. For the purposes of this (necessarily reduced) sketch of Critical Psychology, I will only depict particular aspects of the meanings of selected Critical Psychology concepts that have been relevant for the development and use of Narrative Transformation, namely agency, subjectivity, reasons, grounding/groundedness, meaning, functionality and objectification-appropriation (here, only in their human specificity).

*Agency* (ibid, especially chapters 6.3 and 7; in English sometimes referred to as "action potency") denotes the kind, and extent, of presence and personal availability of, possibilities for acting and influencing phenomena in a way that is relevant for the individual. Each person's agency corresponds with the relation between, on the one hand the general pool of possibilities (and restrictions) that are available and attainable under particular historical-societal circumstances and on the other, the way these given circumstances with their generally available possibilities are subjectively apprehended by the person.

For the purposes here I will refer to Critical Psychology's concept of *subjectivity* (ibid, pp. 233 ff) only as a concept that denotes differences between individuals, e. g. in their ways of experiencing, interpreting, reasoning and having reasons. Subjectivity (including subjective reasons) is grounded in each person's respective specific socio-material conditions - but cannot be derived from studying them together with the person's movements, expressions etc. Hence, according to Critical psychologists such as Holzkamp (1983), it is never possible for one person (or group of persons) to appropriately grasp, predict or change another person's subjective experiences, reasons or »world«. Rather, this has to be reconstructed and »approximated« from each subjective standpoint, possibly by interleaving several individuals' perspectives in *intersubjective* exchange (ibid, pp. 233ff and chapters 7 and 9).

In Critical Psychology, the concept of *meaning* (always meaning related to a specific phenomenon; ibid, pp. 172-174 and chapter 6.3) is closely linked to purpose and denotes what »one« can do (with this phenomenon). Two aspects of meaning have to be distinguished: meaning in its generalized aspect and in its specific aspect. *Generalized meaning* refers to the »societal average« meaning, the prevalent, widespread or common meaning of a phenomenon. *Specific meaning* (ibid, chapter 7.5) denotes the societally mediated meaning of a phenomenon in contingency to specific circumstances: location, historically specific constellations and situation as subjectively experienced by specific individuals or groups. Here, it is

central to keep in mind that most phenomena of every-day relevance have been created by humans, exactly for serving specific purposes (or sets of purposes). A purpose is usually a purpose for a person or a group of persons themselves; but often it is also expected that this purpose is generalized in the sense that the purpose will emerge as an issue for other people as well. People then create a new phenomenon (e. g. artifact, work means, computer application) - one that serve this purpose that has not yet been served before - for future application by themselves and possibly by others, and this new phenomenon may later be used by other people who face the same limits and who have not yet had anything available that has served this purpose.

When considering Critical Psychology - as practiced mainly in Berlin and including the specific effort of Memory Work - a contradiction becomes apparent. On the one hand, efforts have been geared toward improving participants' conditions, their circumstances and their quality of life. But on the other hand, the development and use of material artifacts, e. g. *work means*, as a central part of people's lives, has largely not been addressed as part of the conditions and circumstances investigated by Critical Psychology project participants in their investigations from the standpoint of the subject. Especially, *developing* and harnessing material artifacts has largely been left unaddressed.

The concept of *functionality* of means (e. g. computer applications) as used here (and elsewhere, cf. Törpel 2003,2004) and, as noted, that has not yet been a focus of Critical Psychology, is always related to some specific means, in a way that denotes the meaning of the means: what can be done or achieved with a means, what purposes it serves. The central point of this concept of functionality is that what can be done »with« a means, is often not entirely »contained in« the means itself, even though functionality, especially of a computer application, often becomes attributed to the artifact itself (for critiques of attributing distributed functionality to individual artifacts see e. g. Suchman 2000, Latour 1999). Instead, the actual »functionality-in-active-use« that is ascribed to the means emerges in an interplay between the means (e. g. computer application) and other (historically grown, co-developed and differentiated) phenomena, such as

- further devices, work means, artifacts, infrastructures which are actively in use and in which a new means becomes integrated,
- characteristics of the actors (or actor groups) who want to develop, introduce, harness, make use of etc. a specific new means,
- purposes, needs, desires, agendas, objectives of the actors for the development, introduction etc. of the means,
- the actors' specific practices and
- the involved actors' social structures in which they act, e. g. organizations, circles.

In this way, the functionality attributed to computer applications is framed as always being a *distributed and dynamically evolving functionality in use*.

The double concept of objectification and appropriation becomes relevant in connection with any activity-theoretical consideration of meaning (of phenomena, artifacts, means, computer applications created by humans; cf. Holzkamp 1983, pp.176-178). Phenomena (as created by humans) and their meanings develop further over time, or rather: humans develop them further. This may be seen both from a generalized and a specific point of view. Any individual human or group of humans does not have an alternative to acting (also: perceiving, experiencing, interpreting etc.) from their subjective standpoints. A phenomenon may have a generalized meaning - but there is no authority (such as a »generalized meaning assessment committee«) that ultimately and objectively can assess, arrive at a conclusion and enforce what this generalized meaning is at one given point in time. Informed guesswork (interpretation) cannot be avoided. Nevertheless, when someone creates a phenomenon (e. g. a means) that serves a specific purpose, helps do a specific job, overcome a limit - after the experience of not yet having been able to serve this purpose, get this job done - this often implies that the creators assume that this limit will be faced by others as well. Yet - nobody can ever be sure that another person's or group's specific circumstances provide in fact »a case« of the kind of circumstances that resulted in the development of the original phenomenon (solution, artifact, means, computer application). This can only be investigated by the new human or group themselves.

The local specific inquiry and the harnessing of the phenomenon (artifact, means, computer application) in its meaning/functionality is what I refer to as *appropriation* (building on Leont'ev 1973, Leontyev 1981, Holzkamp 1983). The altering and codifying aspect of discovering, harnessing and realizing new possibilities, meanings, functionality of phenomena that results in new (versions of) phenomena that serve new generalized purposes is what I refer to as *objectification* (building on Leont'ev 1973, Leontyev 1981, Holzkamp 1983).

The double concept of objectification and appropriation, as further developed by Critical Psychology is an important reference concept for Narrative Transformation. Narrative Transformation can be seen as one operationalization and practical guide for responsibly objectifying and appropriating notions related with or attributed to the functionality of computer applications. Narrative Transformation is the attempt to support reflective and creative development practices of computer applications toward the improvement of the participants' quality of life. In carrying through Narrative Transformation, an objectifying process is interpreted as a process of giving notions, concepts, practices that are in use a more »solid«, durable, generalizeable and pervasive form when they become incorporated in computer applications than when they remain in other forms, such as oral or written forms. Regarding the aspect of appropriation, Narrative Transformation is the attempt to support practices of

understanding, utilizing, re-capturing, re-inventing, questioning etc. of existing IT work means (durable, generalizable and pervasive forms of notions, concepts and practices that are in use) that are in use.

#### *Memory Work and Narrative Transformation*

The specific procedure of Narrative Transformation represents a further development of the techniques of Memory Work, one of the attempts to operationalize the programme of subject research (Subjektwissenschaft) in Critical Psychology (cf. Haubenreisser & Stöckmann 1993, Haug 1999). Memory Work was initially developed in the 1970s and has been extensively applied and differentiated, especially by Frigga Haug and her co-workers and students at the Free University in Berlin and, later, at the University of Applied Business and Political Science (Hochschule für Wirtschaft und Politik) in Hamburg. Memory Work, whose initial target groups were women's groups in Germany in the 1970s, turned out to be instrumental for clarification and practical action toward a better life as part of an empowerment process. Memory work has not been devised for directly supporting the development of *artifacts*, or *work means*; yet, this paper demonstrates its potential as a suitable basis for developing Narrative Transformation.

#### *Lifestyle corresponding with Critical Psychology*

In the view of the author, a lifestyle corresponding with the assumptions and approaches in Critical Psychology is one that is geared toward finding, exploring, and realizing possibilities of fulfillment and enjoyment that are within and beyond the current possibilities. This includes becoming familiar with, grappling with, and overcoming one's own limits, as well as (and in relation to) the limits of the currently existing conditions of one's life. In most societies this also implies the need to address those relations of power that on the one hand »build a frame« around people's every-day lives and on the other manifest themselves as immediately experienced, concrete, relations of possibilities and restrictions.

Hence, a method such as Narrative Transformation is primarily geared towards improving the participants' quality of life by extending spaces of possibilities, in the sense that wider conditions are improved (or at least taken into consideration) as part of the improvement measures. As in some other research programmes (e. g. ethnomethodology), the research programme of Critical Psychology requires professional researchers to be interested in those of their own practices and life conditions that they at least partially share with other people who are not professional researchers, and whose practices and conditions are scrutinized during the research. People who participate in research, but are not professional researchers, must be considered as either already qualified to be co-researchers or to receive support by the professional researchers in their progress towards becoming co-researchers (Holzkamp 1983, chapter 9).

In this sense, a lifestyle in accordance with Critical Psychology is one that is very research-oriented, not necessarily in the sense of academic research, but in the sense that people are interested in understanding, and

possibly overcoming, the current conditions, practices, limits and possibilities of their every-day lives. This implies seeking to understand how current phenomena have evolved and how, given the way they are, they might further develop. It also implies seeking to discover how things could be totally different than they currently are; how oneself, maybe together with fellows (e. g. peers, colleagues, allies, fellow-sufferers), could influence historical trajectories in a beneficial direction. From this viewpoint, *causes, historical trajectories* (as in both Critical Psychology's historical-empirical approach and its research approach from the standpoints of the subject, subject research) and *reasons* (as in Critical Psychology's research from the standpoints of the subjects) are all assumed as important constituents of individual and societal reality. In short, a lifestyle that corresponds with Critical Psychology can be seen as inquiring, communicative, alliance-oriented, understanding, improvement oriented, active, activating, fulfillment oriented, and full of respect towards, history, feelings, meanings and reasons (of one's own and others).

#### **ACKNOWLEDGEMENTS**

I would like to express my gratitude to Hans-Jörg Burtschick and Steve Harris for having been inspiring discussion partners and constructive readers of drafts in the process of preparing this paper. This paper has had a longer history, and the collaboration with a number of colleagues has made the current version possible. Among these colleagues are: Steffen Budweg, Hans-Jörg Burtschick, Steve Harris, Birgit Huber, Ulrich Mill, Volkmar Pipek, Meik Poschen and Alex Voß. Thanks also to the participants of the ATIT workshop.

#### **REFERENCES**

1. Bentley, R., Horstmann, T & Trevor, J. (1997). The World Wide Web as enabling technology for CSCW: The case of BSCW. *Computer Supported Cooperative Work: The journal of Collaborative Computing*, 6, 111-134.
2. Bjerknes, G., Ehn, P. & Kyng, M. (1987). *Computers and democracy - a Scandinavian challenge*. Aldershot, UK: Avebury.
3. BSCW (2004). <http://bscw.fit.fraunhofer.de/> (page last accessed November 22, 2004; last BSCW system version November 2004).
4. Carroll, J. M. (2000). *Making use: Scenario-based design of human-computer interactions*. Cambridge, MA: MIT Press.
5. Dawidow, H. E. & Malone, M. S. (1992). *The Virtual Corporation. Structuring and Revitalizing the Corporation for the 21st Century*. New York, USA: Harper Collins.
6. Greenbaum, J. & Kyng, M., (eds.) (1991). *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.
7. Haubenreisser, K & Stöckmann, M. (1993). *Der andere Blick - Erinnerungsarbeit als Methode im Bildungsurlaub*. In: F. Haug & E. Wollmann (eds.),

- Hat die Leistung ein Geschlecht? Hamburg: Argument Verlag, 139-172.
8. Haug, F. (1999). Vorlesungen zur Einführung in die Erinnerungsarbeit. Berlin: Argument.
  9. Holzkamp, K. (1983). Grundlegung der Psychologie. Frankfurt a. M.: Campus.
  10. Kleinen, B. & Herczeg, M. (2003). Handling material in learning activities. The Knowledge Infrastructure KOIN. Proceedings of ED-Media 2003, Honolulu, Hawaii, USA, 640-643.
  11. Kuhn, S. & Muller, M. (eds.) (1993). Special Issue on Participatory Design. Communications of the ACM, 36(4).
  12. Latour, B. (1999). Pandora's hope: essays on the reality of science studies. Cambridge, MA: Harvard University Press.
  13. Leont'ev, A.N. (1978). Activity. Consciousness. Personality. Englewood Cliffs, NJ: Prentice-Hall.
  14. Leontyev, A.N. (1981). Problems of the Development of Mind. Moscow: Progress.
  15. Mowshowitz, A. (1997). Virtual Organization. Communications of the ACM, 40 (9), 30-37.
  16. Powell, W. W. (1989). Neither market nor hierarchy: Network forms of organization. In: B. M. Staw & L. L. Cummings (eds.), Research in Organizational Behavior, vol. 12. Greenwich, Connecticut: JAI Press, pp. 295-336.
  17. Schuler, D. & Namioka, A. (eds.) (1993). Participatory Design: Principles and Practices. Hillsdale, NJ: Lawrence Erlbaum Associates.
  18. Suchman, L. (1994). Working relations of technology production and use. Computer Supported Cooperative Work, 2, 21-39.
  19. Suchman, L. (1995). Speech acts and voices: Response to Winograd et al. In: Computer Supported Cooperative Work, 3, 85-95.
  20. Suchman, L. (2000). Human/Machine Reconsidered. Published by the Department of Sociology, Lancaster University at: <http://www.comp.lancs.ac.uk/sociology/soc040ls.html> or <http://www.comp.lancs.ac.uk/sociology/research/restopic.htm#spatiality>. Last accessed: November 26, 2004.
  21. Törpel, B. (2000). Self-employed labor meets codetermination - participatory design in network organizations. In: Cherkasky, T., J. Greenbaum, P. Mambrey & J. K. Pors (eds.), Proceedings of the Participatory Design Conference, Nov. 28 - Dec. 1, 2000, in New York, NY, USA. Palo Alto, CA: CPSR Press, 184-191.
  22. Törpel, B. (2003). Interest and Narration in Applied Information Technology - A Strange Combination? Or: From Freelancer Networks to Stories for Participatory Design. Second Tampere Conference on Narrative, Ideology and Myth, June 26-28, 2003. <http://www.uta.fi/conference/narrative/papers.html>
  23. Törpel, B. (2004). Forming circles and making things happen - Designing functionality that supports cooperative work in fragmented work environments. In: Proceedings of the 27th Information Systems Research Seminar in Scandinavia, IRIS 27 - Learn IT, Know IT, Move IT - August 14-17, 2004 in Falkenberg, Sweden.
  24. Törpel, B., Pipek, V. & Rittenbruch, M. (2003). Creating heterogeneity. Evolving Use of Groupware in a service network. In: Andriessen, E., E. Heeren, M. Hettinga & V. Wulf (eds.) Computer Supported Cooperative Work - The Journal of Collaborative Computing, special issue on "Evolving Use of Groupware", 381-409.
  25. Törpel, B. & Poschen, M. (2002). Improving Infrastructures by Transforming Narratives. In: T. Binder, J. Gregory & I. Wagner (eds.), Proceedings of the Participatory Design Conference, Malmö, Sweden, 23-25 June 2002. Palo Alto, CA: Computer Professionals for Social Responsibility, 248-253.
  26. Törpel, B., Wulf, V. & Kahler, H. (2002). Participatory organizational and technological innovation in fragmented work environments. In: Dittrich, Y., C. Floyd, R. Klischewski (eds.), Social Thinking. Software Practice. Cambridge: MIT Press, 331-356.
  27. Tolman, C. W. (1994). Psychology, Society, and Subjectivity: An Introduction to German Critical Psychology. London: Routledge.
  28. Voß, G. & Pongratz, H. (1998). Der Arbeitskraftunternehmer. Eine neue Grundform der Ware Arbeitskraft. Kölner Zeitschrift für Soziologie und Sozialpsychologie, 50, 1, 131-158.

**Appendix: The full initial episodes from the illustrative practical example of the use of Narrative Transformation in a research group**

**Episode 1:**

Now, where is the checklist? Aha, here. OK. ToDo. Yep. Well – Underwear, T-shirts, socks, pants, washing stuff ... hmm ... survival has been guaranteed. When again does the train to A. [city] leave? Shit – I must have put the note into the scrap paper and that's already downstairs – not, of course, the general garbage, maybe I should also do the dishes? I bet B. [flatmate] won't eat up the tomatoes. Apropos – do I have a printout of the C. [conference] paper? Actually, I could read it on the plane. Have to turn on the notebook anyway because of the train. Where actually do I have the Bowker & Star book? In D. [city abroad]? No, here it is. Actually, I have too much hand luggage, I better take the Luhmann, it's less weight. Apropos – I wanted to write a mail to E. [colleague, mentor] at last. Again, I haven't properly said goodbye to the colleagues from the control room, they don't know where I am, they'll think again: "the lazy loafer". Anyway.

Quickly call F. [partner], otherwise she'll feel neglected again. Should I take the mobile phone with me? No, not even more hand luggage. Rather take the technomethodology paper. In any case don't forget the keys of the institute. Well – now still only turn on the computer, then I go to sleep, and tomorrow I can do the rest.

### **Episode 2: A day in "G" [research group]**

Mid-week. Early afternoon, he is just back from school.

Most duties had been done, now he wanted to take care of his work for G [research group]. First sorting out, what had been discussed last time, what had been agreed upon, planned. Which tasks had he taken and which ideas and projects planned for himself. Where were the minutes and where the due dates. He is not in the office after all.

OK, all set, just begin. Set priorities. First the important and urgent things, don't lose track of the important but not urgent things.

Now, he has done the tasks he agreed to do at the last meeting, where were again the previous topics which he had planned and postponed? And what's up for tomorrow? Simply just look in the BSCW, the third time today. No agenda for the next meeting, but a new draft of the paper of her, his colleague. Hmm, what has changed, what is new? And where does she wish feedback? Simply just ask her.

Wow, he has reached her and she has time again (where does she take it from and where is her leisure?). Time for his questions, suggestions and new reflections. They argue, plan, philosophize and many images in his head become clearer, mark off and show new paths which want to be filled and pursued.

Puh, that was intense. And very constructive. Exhausted, he writes down a few thoughts, records them for further treatment and pursuit. Now quickly write down the clarified topics for tomorrow, propose it to the others as agenda and then: first cook food. Other duties are waiting. He quickly once more checks his eMail: Already feedback from him, the [other] colleague. That feels good. Tomorrow everything will go on.

### **Episode 3: Logo? - Logo!**

As virtually always, his workday began by turning on the computer and getting his emails transferred. During this day he'd have a few chores to get done where any delay would be difficult. Hence he hoped that he'd quickly finish his communication - and hence that there'd not be a particularly important time-critical message.

"Quickly finishing one's communication" it echoed in a low voice in his head while he was reading the mail from H. [city]. His feeling told him that communication was quickly "finished" [terminated once and for all] when it didn't yield the desired results. In this case, he himself and his working group had not kept an agreement, an arrangement set up at the last retreat of his project at large: the logo competition...

Everybody had agreed to perform a logo competition via BSCW in order to find a project logo at last. And BSCW provides all possibilities: whoever wants to loads up a logo, everybody scores it, and the creator of the most or highest scored logo may enjoy a bottle of champagne (which I. [colleague] from his working group had donated after all). So far the theory.

The whole thing hence rests upon a self-responsible participation that appears quite simple - by aid of a well-suited technical system. Why had nobody from his group participated, why hardly anybody from the other project partners and why above all not he himself? At least they could have scored the two loaded-up logos - but during the time before the deadline it had gone unnoticed. And therefore this mail from H. [city]: astonishment because of their non-participation.

He shook his head and was angry at himself. Back to the agenda.

### **Episode 4: All for all? All for one?**

In the G. group [research group] they had agreed to, after the weekly work meeting, either discuss content-related work of the individuals or jointly work on the topic "discursive design".

At the next meeting a text by J. [research group member] was to be discussed. The text was a short sketchy project description of her dissertation which she wanted to send her potential supervisor. Jointly discussing the text was currently not possible because, over a period of several weeks, every week important reasons kept them from discussing her text. For example, discussing the questionnaire that was to be generated for a project took longer as that enough time remained for discussing her text. Or not everybody was present. Or there were other reasons.

For some time, K. [colleague and flatmate] and she discussed the text at home, and this way she got impulses for the revision. They anyway quite often talked about their research endeavors.

L. [colleague, research group member] and she set a phone appointment, and then during the phone call talked about the text for a long time. Now she had further clues for how she could improve the text. During the long conversation over the phone they, beyond this, the spun research programmes. It had again been fun and very productive.

At some point in time, I. [senior research group member] and she could also take time together for discussing the text. Again, she got useful impulses.

She was very pleased with the feedback and vigorously revised the text.

Had they discussed the text in the G. group, then maybe they overall would have spent less time discussing the text. She asked herself if the feedback she would have received in the group was comparably thorough. Or if the group comments had even been better because everybody could have inspired each other? She could not tell.

# ActAD in information systems analysis and design: Academic reflection

Mikko Korpela

HIS R&D Unit, University of Kuopio, PL 1627, FI-70211 Kuopio, Finland  
mikko.korpela@uku.fi

## ABSTRACT

This short note provides background information and an academic reflection on a portfolio of four methods for information systems analysis and design, which are based on Activity Analysis and Development (ActAD).

## Keywords

Activity theory, information systems, software systems, analysis, design, methods

## HISTORICAL BACKGROUND

In the mid 1980s, a national doctoral education programme for Information Systems (IS) was established in Finland. One of the doctoral students, Mikko Korpela, wanted to study the IS development practice in Africa, and needed a theoretical and analytical tool for grasping “the role of technology in work” and “work practice in social context”. Like Kari Kuutti of the same programme, he identified Activity Theory (AT) and specifically Developmental Work Research (Engeström 1987) as the most suitable basis available for that purpose.

Since there was a need to adjust DWR to the specific needs of IS research and to expand it in various directions, the modified version was tagged Activity Analysis and Development (ActAD) to avoid confusion (Korpela et al. 2000, 2004). We do not try to repeat these papers here – instead, a proof copy of the latest paper presenting ActAD in the IFIP WG8.2 conference in July 2004 is attached (not exactly the published copy, only for the workshop participants!)

One of the directions of expansion is “zooming out” into wider scopes – work activities in organizational, economic, social, cultural and political contexts – which was elaborated on in the doctoral theses of Korpela (1994), Anja Mursu (2002) and Abimbola Soriyani (2004), and summarized into a research agenda (Korpela et al. 2001a). This branch of research aims at “contextually aware” methods for practitioners particularly in African contexts. Mursu’s work on the sustainability of information systems (Mursu et al. 2004a) is approaching the state of methodological guidelines, which however have not been tried in practice so far.

Applying the general activity-theoretical frameworks and work development methods to the specifics of information systems started in the Finnish-Nigerian INDEHELA programme and continued in the PlugIT project in

healthcare in Finland. Theoretical papers were published on IS development as an activity (Korpela et al. 2002b) and on the chain of activities from IS use through IS development to IS research (Korpela et al. 2002a). This branch of research operates often on the level of activity networks, peeping into the inner affairs of individual activities as well but mostly dealing with the collective aspects therein. Marika Toivanen, Heidi Häkkinen, Anja Mursu and colleagues have worked towards practicable methods for practitioners in the areas of exploring a “grey area” (Toivanen et al. 2004), feasibility studies identifying information needs within and around an activity (Häkkinen et al. 2004), as well as applying usability design on the activity level (Mursu et al. 2004b).

The research and methods described so far are mainly descriptive, applicable to IS needs analysis. However, activity theory in general and ActAD in particular can also be used for constructive purposes, i.e. to support the design of organizational information systems and the design of software products that can be used in organizations. To this end, there is a need for “zooming in” – from collective aspects of activities towards individual actions and the way software artifacts can be used in and between actions.

The agenda for “zooming in” was presented in the 2nd Nordic-Baltic AT conference: the gap between activity-theoretical IS methods and mainstream software engineering (SE) methods can be bridged by relating “object-to-outcome transformation” with “business process” and “technologically mediated action” with “use case” (Korpela et al. 2001b). Anne Eerola, Marika Toivanen and Annamari Riekkinen with colleagues have worked towards transforming this general idea into more precise understanding of the relations between AT/IS and SE concepts as well as into practicable methods for practitioners. This work has been conducted as a side process in the PlugIT project in single case studies. A major research project, ZipIT, has been proposed to carry on the methodological development in full scale in 2004–2007.

## DESCRIPTION OF THE METHODS PORTFOLIO

A portfolio of four specific methods for specific IS analysis and design purposes is presented in AT IT 2004.

Method 1, presented by Anja Mursu, focuses on analyzing the present state of one activity in more detail, possibly as a subtask within the other methods. The ActAD framework is used as a checklist, adapted to the specific

activity, in group discussions with the actors of the activity. The outcome is a description of the process, outcome, actors and means of the activity as well as its mediated relations with neighbouring activities. This is the core method of applying the ActAD framework of concepts as a lens.

Method 2, presented by Marika Toivanen, is for “requirements exploration” in previously weakly understood settings, e.g. when several uncoordinated activities take place between or outside of formal organizations. Thematic interviews and observation are used for producing a description of the current activity networks, services and processes, the most important challenges for development within or between activities, as well as an outline of the target state that can be achieved by organizing the work in new ways or by introducing new pieces of software.

Method 3, presented by Heidi Häkkinen, is a feasibility study method used for identifying and describing the needs for improving the information flows (information management, organizational information systems) in settings that are more formalized and better understood, but still involving several activity networks and organizational boundaries. Most often a feasibility study is conducted within an organizational unit. The core of the method consists of two multi-actor meetings of about two hours each. The outcome is usually recommendations about which actual development projects are needed – organizational changes, training, software procurement, software development, etc.

Method 4, presented by Annamari Riekkinen, proceeds from activity analysis into designing the architecture and the user interface of a new or improved software product, the need for which has been identified through methods 1, 2 or 3. The analysis is taken into more detail by describing the current object–outcome process in two ways: as a scenario (a narrative of the actions needed by different actors to produce the outcome) and as an activity diagram according to the Unified Modeling Language (‘activity’ in UML refers to a series of interactions by an actor with a computer system). The targeted new way of working with the new software is also described as a scenario, and a concept analysis is performed. The functionality of the system is defined with the help of the activity diagrams, and it is described as UML use cases. The concept model and use case descriptions are the necessary input to the Cheesman-Daniels procedure of software component design, while the target scenario is an input to user interface design according to the GUIDe procedure.

The user interface and architecture design processes of method 4 run in parallel and could be seen as two different but interwoven methods.

Taken together, the portfolio of four methods covers major aspects of IS development, from needs analysis through requirements specification to software product design. The portfolio should be supplemented by methods that address the introduction of software systems (new or modified) into organizations.

## **PRACTICAL EXPERIENCE**

As the historical description reveals, the methods are in a “prototyping” phase. They have been developed in single practical cases, but not validated by trying them in other cases. There is somewhat more experience on the core method of analyzing the present state of an activity (method 1), but so far only by researchers.

## **RELATION BETWEEN METHODS AND THEORY**

All the methods rely heavily on the concept of activity as a systemic entity that intertwines collective and individual aspects, human and technological aspects, dynamic and structural aspects, as well as holistic unity and contradictions within the same framework. Activities are regarded as objective, real-life phenomena, not just theoretical constructs. Therefore the theoretical model of activity should be constantly tested by practice, and modified to better reflect what activities are in reality.

Since activities are purposeful, methods that target at supporting activities by technological means have a built-in focus on the purpose of work. Unlike mainstream business process reengineering approaches however, activity theory brings the actors into the limelight as human beings, not just as process resources. The methods in our portfolio may currently rely too much on the design experts, i.e. the participative and cooperative aspects of the methods may need to be strengthened.

The three levels of activity networks – activities – actions appear meaningful and call for different methods. When “zooming out” to the level of activity networks, activity theory needs to be complemented by inputs from organizational theory, information systems, information resources management, and so forth. When “zooming in” to the level of actions, inputs are needed from ergonomic work analysis, design sciences, software engineering and so forth.

[discussion about the possible problems and limitations, applicability, adjustability etc. of the methods – in the workshop!]

## **ACKNOWLEDGEMENTS**

Heidi Häkkinen, Anja Mursu, Annamari Riekkinen, Marika Toivanen and others have greatly contributed to this paper.

The paper is based on research funded by the National Technology Agency Tekes through the PlugIT project, grants no. 40246/02 and 90/03 (2001-2004), as well as research funded by the Academy of Finland through the INDEHELA-Methods and INDEHELA-Context projects, grants no. 39187 (1998-2001), 201397 (2003) and 104776 (2004-2007).

## **REFERENCES**

ActAD: Activity Analysis and Development.  
[www.uku.fi/tike/actad](http://www.uku.fi/tike/actad)

Engeström, Y. *Learning by expanding: An activity-theoretical approach to developmental research*. Orienta-Konsultit, Helsinki, 1987.

Häkkinen H, Korpela M. A participatory assessment of IS integration needs in maternity clinics, in *ITHC 2004: IT in Health Care: Sociotechnical Approaches*, Portland, OR, 13-14 September 2004. Forthcoming.

INDEHELA: Informatics Development for Health in Africa. [www.uku.fi/tike/indehela](http://www.uku.fi/tike/indehela)

Korpela, M., Soriyan, H.A., and Olufokunbi, K.C. Activity analysis as a method for information systems development: General introduction and experiments from Nigeria and Finland, *Scandinavian Journal of Information Systems* (12:1), 2000, pp. 191-210.

Korpela, M., Mursu, A., and Soriyan, H.A. Two times four integrative levels of analysis: a framework, in *Realigning Research and Practice in Information Systems Development: The Social and Organizational Perspective*, N.L. Russo, B. Fitzgerald and J.I. DeGross (eds.), Kluwer Academic, Boston, MA, 2001a, pp. 367-377.

Korpela, M., Eerola, A., Mursu, A., and Soriyan, H.A. Use cases as actions within activities: Bridging the gap between Information Systems Development and Software Engineering, abstract, in *2nd Nordic-Baltic Conference on Activity Theory and Sociocultural Research*, Ronneby, Sweden, 2001b, p. 51.

Korpela, M., Mursu, A., Soriyan, H.A., and Eerola, A. Information systems research and information systems practice in a network of activities, in *Social Thinking - Software Practice*, Y. Dittrich, C. Floyd, and R. Klischewski (eds.), MIT Press, Cambridge, MA, 2002a, pp. 287-308.

Korpela, M., Mursu, A., Soriyan, H.A., and Olufokunbi, K.C. Information systems development as an activity, *Computer Supported Cooperative Work*, (11:1-2), 2002b, pp. 111-128.

Korpela, M., Mursu, A., Soriyan, A., Eerola, A., Häkkinen, H., and Toivanen, M. Information systems research and development by activity analysis and development: Dead horse or the next wave?, in *Information Systems Research: Relevant Theory and Informed Practice*, B. Kaplan, D.P. Truex III, D. Wastell, A.T. Wood-Harper, J.I. DeGross (eds), Kluwer Academic, Boston, MA, 2004, pp. 453-470.

PlugIT: Healthcare Applications Integration. [www.plugit.fi](http://www.plugit.fi)

Mursu A, Korpela M, Soriyan A. A generic framework for analyzing the sustainability of information systems, in *Proceedings of the Americas Conference on Information Systems*, New York, NY, August 2004a. In print.

Mursu A, Korpela M. The usability of an activity: How to combine usability design and activity development. Accepted to *3rd Nordic Conference on Cultural and Activity Research*, Copenhagen, Denmark, 3-5 September 2004b.

Soriyan, H.A. *A conceptual framework for improved information systems development methodology for educational and industrial sectors in Nigeria*, PhD thesis. Obafemi Awolowo University, Ile-Ife, Nigeria, 2004.

Toivanen M, Häkkinen H, Eerola A, Korpela M, Mursu A. Gathering, structuring and describing information needs in home care: A method for requirements exploration in a "gray area", in *MEDINFO 2004: Building High Performance Health Care Organizations*, Proceedings, San Francisco, CA, 7-11 September 2004. IMIA, 2004. In print.

# Analyzing the present state of an activity

Anja Mursu

Department of Computer Science, University of Kuopio

P.O.Box 1627, 70211 Kuopio, Finland

anja.mursu@uku.fi

## ABSTRACT

This paper presents an activity analysis method called Activity Analysis and Development (ActAD). The method can be used by practitioners when they want to develop their own work practices, but the fundamental idea of the method is to combine information systems development (ISD) and work development. The purpose of ISD is to facilitate some work activity by means of some software. Thus we see ISD as a boundary-crossing temporary activity at the border of two departments or companies, concerning both practitioners (users) and IS developers. The ActAD method provides a tool for both users and developers to discuss, analyze and further develop work activities within ISD process. The method itself is quite general in nature, it instructs the procedure how to proceed. In each case, more detailed questions and an application must be considered (see Toivanen, Häkkinen and Riekkinen papers in the workshop).

In this paper we introduce the ActAD method, and how it can be applied in work analysis and development. The outcome of using the method is a rich picture and an understanding of an activity at the present state. The system requirements should be based on that picture.

## Keywords

work development, information systems development, activity theory

## INTRODUCTION AND MOTIVATION

Information technology (IT) is part of information system (IS), which provides services or tools to an activity, which for its part provides services or products to its clients. When we introduce new technologies, it has a consequential impact on the activity in question. Thus the analysis of an activity in most cases is an essential start in order to create successful and sustainable technological solutions. The needs for technical improvements should be based on the needs in an activity. The principal idea is that information systems and work activities should be developed at the same time with the same goals. However, system developers and designers, who facilitate other people's work by technology, do not have tools for work

development, and practitioners usually are not capable of providing IS requirements based on work development efforts.

We have applied activity theory and specially Developmental Work Research by Engeström (1987), and tried to adjust DRW for specific needs of IS research (Korpela et al. 2004). Accordingly we have elaborated our modification, for example by creating more 'contextually aware' methods for practitioners. The method discussing in this paper, the ActAD method, is a general method for work development, and the case introduced here is quite descriptive. The ActAD method is based on a framework, which can be used as a checklist, adapted to each case. The framework is introduced briefly in this paper.

## The purpose of the method

With the ActAD method we present here, users and designers can together analyse an activity and have mutual understanding, what are the problem areas, what areas should be facilitated with some technical tools, and what is the vision of an activity in future. The basic method includes the following steps, which can be embedded to IS development process or conduct independently:

1. Identify the central activity
2. Structural analysis
3. Developmental analysis
4. Work development

## OUTCOME OF APPLYING THE METHOD

The outcome of applying the method is basically an understanding of the activity where new technologies are considered. Technical solutions can be assessed better, if we understood activity that is using it. In addition, the impact of the new IS is understood better and the work can be developed accordingly.

The concrete outcome is an activity analysis report, including descriptions of different elements of an activity and a network of activities, diagrams and drawings, problems descriptions, new ideas and so for, resulting in activity development requirements. The development work can then continue to more detailed analysis concerning technical solutions and resulting in system requirements.

## WHEN METHOD IS SUITABLE AND BY WHOM

The most logical part in IS design process for activity analysis is in the early phase of a requirement analysis or a feasibility study. Naturally the method can be used in pure work development process, but our viewpoint is in IS

development and work development concurrently. Thus the place for an activity analysis and the ActAD method is when organization has decided to improve some activity, and probably with some technical solution. After such decision, it is most advisable to conduct an activity analysis so that the possible impact of a new technical solution is intended.

The ActAD method is most suitable when the activity is a collection of several people, tools, applications and so for. In other words, the method is suitable when we want to develop information systems (IS), which is regarded as something fundamentally different from a software package as such. An IS refers to the information processing and management processes within an organization, and it includes both technical solutions and social elements. This IS provides services to an activity which is to be analysed. The method is also suitable when several activities are working on the same object, creating a network of activities.

### Development process and intended users

The method is most suitable when user participation is possible and required in ISD. However, the development process should be work and information systems development, not product development or off-the-shelf implementation.

The best way to apply this model is when users and designers do the job together. However, the aim is to provide a method that users can apply by themselves so that as a result they would have kind of a 'shopping list' when they contact system delivers.

The competencies that are required are not demanding. Users should be expert in their work processes, and designers should know suitable information technologies. In other words, the requirements are common professional knowledge, since the method is supposed to be practical and well documented to be applied by anyone who wants to analyse some work.

### SHORT INTRODUCTION OF THE METHOD

The method is based on a framework named Activity Analysis and Development (ActAD), illustrated in Figure 1. The background of the method is in Activity Theory and Engeström's work development model (Engeström 1987).

The background of the ActAD framework and its elements has been explained in Korpela et al. 2004. However, in the following the elements in Figure 1 are described briefly:

1. The outcome is what defines an activity.
2. The object is a starting point, which is transformed into the outcome during the work process of the activity.
3. Various actors and actions are needed to transform the shared object into the outcome.
4. The actors have different means of work at their disposal, to act on the object. The means can be individual or shared, and they can be physical artefacts or abstract tools like knowledge, skills, expertise and so for.
5. The actions of the individual actors need to be organized through means of coordination and communication.

6. All the actors taken together act as a collective actor, like a team or a project group.

7. All the elements must fit together in a systemic way, characterized as the mode of operation of the activity. This mode has a historical background, which is relevant in order to understand the present and to plan for a future.

8. The elements of an activity are produced by some other activity, and the outcome of an activity is intended for some other activity. These production / service relations between activities creates an activity network

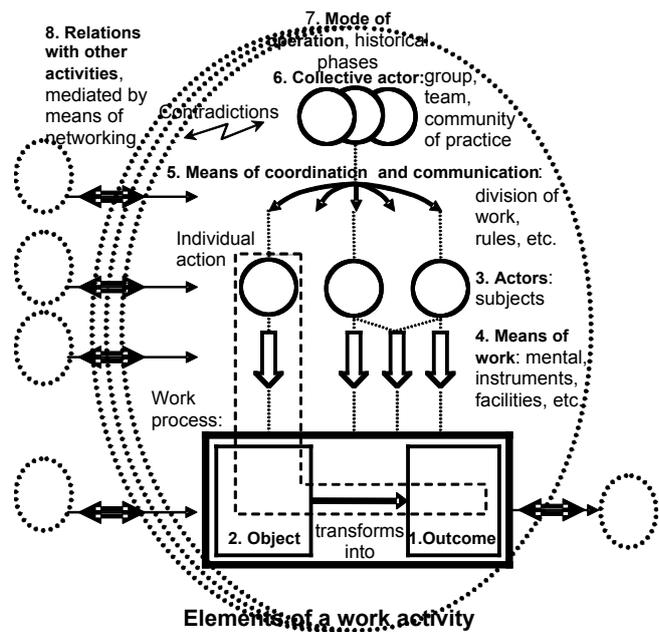


Figure 1 The elements of ActAD model.

Information technology can be used in various roles within activities – either as a means of working on the object, as a means of coordinating the collective process, or as a means of networking. These means are usually developed without analysing the others, or the activity as a whole. In the following the steps how to conduct activity analysis are presented.

### PROCEDURE OF APPLYING THE METHOD

The model can be used as a checklist when analysing activity. Table 1 gives an outline of questions that can be discussed during the analysis. The questions should be elaborated according to case, meaning that in each case relevant issues and questions vary. However, the basic elements are the same.

The elements and questions in the checklist (Table 1) can be discussed in interviews or focus group sessions. When practitioners want to conduct activity analysis by themselves, they can organize a workshop where the questions are discussed. The following steps have proved to be workable when analysing activity and its network.

### **I Phase: Identify the central activity**

An activity comprises a number of people working on something shared in an organized way – not necessarily at the same time and place – to produce a joint outcome.

By looking at what are the ‘shared objects of work’ and ‘jointly produced outcomes’, the most central activity in the case and the various people who work on it are identified.

#### *Prepare interviews, outline questions*

Prepare an interview, a focus group or a workshop. Outline questions to be discussed based on the checklist provided in Table 1. The aim is to find out a present state of an activity, but it is not prohibited to start discussing new ideas.

### **II Phase: Structural analysis**

The outcome of the first phase is the list of people (or groups, like nurses, doctors etc.) and a blueprint for discussed questions. The model can be presented for participants by an illustrated picture, where different elements are presented as real examples of an imaginary activity (see an example in Appendix 1).

#### *Organize interview, focus group or workshop*

Organize the meeting. There must be one person leading the discussion and somebody to make notes. There are two options for this phase:

1. just go through the questions, without trying to illustrate the activity anyhow. It is recommended to record discussions.
2. go through the questions and at the same time try to create an illustrative picture of the activity according to Figure 1. Use wall graphs, brainstorming or whatever method is suitable for your situation.

The purpose is to let people involved discuss the composition and linkages of the central activity and to create a good and rich picture of the activity. It may be useful to alternate between different individuals or professional groups thinking about the questions alternately on their own and then together.

### **III Phase Developmental analysis**

The result of the previous phase is the punch of notes, figures, tapes, wall graphs and so forth. This material must

be composed and analyzed to a holistic picture of the activity.

#### *Analyze interviews, focus group or workshop*

Go through the material you got in the first session. Prepare or revise wall graphs and verbal descriptions, and report the found problem areas.

#### *Organize a second round for analysis*

Go through the composed results with the same people you had in the first interview, focus group or workshop. Use wall graphs, descriptions, figures etc. Discuss and describe also the following questions:

1. **History:** How has this activity and network emerged and developed to what it is now? Can you identify some phases or stages in the overall development?
2. **Problems:** What kinds of weaknesses, deficiencies, and imbalances there are within and between each of the constituent parts analyzed before? Do the parts fit together well?
3. **Potential:** What kind of strengths and emerging new possibilities there are within and between the constituent parts analyzed before? What kind of a new mode of operation of the whole setup could be strived after – what would be the desired next stage in the historical development of this activity and the network? To achieve that stage, what improvements are needed in and between the various parts?

#### *Make corrections and finalize the analysis and reports and get verifications*

Make the final reports with improvement requirements and verify the report by the people involved.

### **IV Phase: Work development**

The actual work development can continue independently or it can be integrated to IS development process. In the work development phase new tools are developed, new knowledge is acquired, people are educated, processes and relations are improved, as identified in the previous phase. It is better to start with experimentations in a limited setup, but keeping all the stakeholders involved. The overall goal must be kept in mind through the process. The development of information system needs specific tools, but it is a different process. However, iteration is a good principle also in work development.

<b>The outcome or service</b>	What is the outcome or service? Who is the client who use the service and for what? Define the outcome, service and clients.
<b>An object and process</b>	What is the object to work on? What is the process from goal to outcome? Define object and goal, and process.
<b>Actors, people</b>	Who are the people, actors, within the activity? What are they doing, what are their roles, where are they come from? Define actors
<b>Means of work</b>	What kinds of tools are needed? What kinds of professional skills, methods, standards and so for? Where do all the means of work come from? What information is needed, and where this information comes from? Define means of work.
<b>Means of coordination and communication</b>	How do people within the activity communicate with each other, by which means? How is work divided and distributed? Where the rules and other means come from? Define the means of coordination and communication within the activity
<b>Collective actor</b>	How is the work organized, what is a hierarchical organization around the activity? How loose or tight is the group or team?
<b>The mode of operation</b>	Summarising all the previously said, what is the mode of operation? What is the context the activity is in?
<b>The network of activities and the means of networking</b>	Who and what are the people and other activities that are needed to be in contact (stakeholders), why and how? How do people within the activity communicate with people or organizations outside the activity? Where are organizational boundaries and how they are related to the whole service chain? Define the means of coordination and communication between activities
<b>The wider context</b>	What is the formal organization? What is the economic, social and political context? What are the financial relations?

Table 1. The checklist for activity analysis

#### AN ILLUSTRATIVE EXAMPLE OF USE THE METHOD

We used this method in the INDEHELA-Method project where we analyzed information system development activity in Nigerian software companies. We conducted four case interviews in three companies. In each company, the purpose was to analyze the ISD activity as it was at the time, not to conduct any work development as such. The interviews were in two parts: first to get knowledge and then go through the results together.

We started the interviews by explaining the method with an illustrative picture (Appendix 1). With the help of the picture, we could explain them the issues that we are interested in their company. We asked them to choose one project in the near past to be used as an example of their work. We then started to ask questions based on the checklist presented in Table 1. An example of modified questions is presented in Appendix 2. The interviews in general were positive and encouraging.

After the first interviews the results were analyzed using the ActAD model. We took one element at the time and created wall graphs. The result was a punch of pictures with descriptions and with these we went back to the companies.

At the second round we went through the wall graphs with the same people. The model worked well to direct the conversation and also to inspire further consideration. Unfortunately, we couldn't manage to organize group discussions to compose development requirements and the

historical dimensions remained weak. However, we got a good picture of the present state of an activity in the software companies in Nigeria, topic that has not been studied before. In the following we illustrate briefly the result of one interview (Mursu et al. 2002).

#### A case of ISD as an activity in a small Nigeria software company

Gamma Corporation was established in 1988 for IT training, software development and engineering. The company now has 35–40 employees altogether in Lagos and Abuja. The organization is hierarchical, divided into software solutions, training and consultancy units, as illustrated in Figure 2. Besides the IT experts, there are secretaries, some drivers and security men. We interviewed the executive director of software solutions, and two analysts.

**Object.** The customer in the case was a bank. The bank was in the process of re-engineering its business processes (BPR). One of the aspects to be renewed was human resource management (HRM), for which the company Gamma was chosen as a provider. The bank had an old system for HRM, which was developed in-house some years earlier. Gamma had also developed an HRM package, which had been customized to a number of organizations. The project was commissioned in mid-1998.

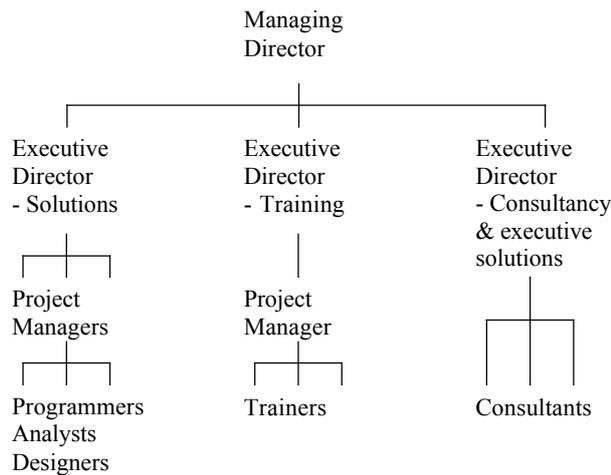


Figure 2: The organizational structure of the company

**Collective actor.** After the contract was signed by Gamma and the bank, they created a project organization. It consisted of the steering committee and the implementation team (Figure 3). The steering committee consisted of heads of relevant departments, head of audit and head of IT-department from the bank. Gamma's project manager attended steering committee meetings as a consultant; he was not formally a member of the committee. The steering committee confirmed a project plan, which included a project schedule, all the tasks, activities and resources. Overall, the steering committee had two main tasks: to solve financial problems and to take care of human resources in the project.

The implementation team included people from both Gamma and the bank. From the bank there were a team leader (one of the senior users), users, bank's own IT-people and the audit representative. From Gamma there were analysts and programmers.

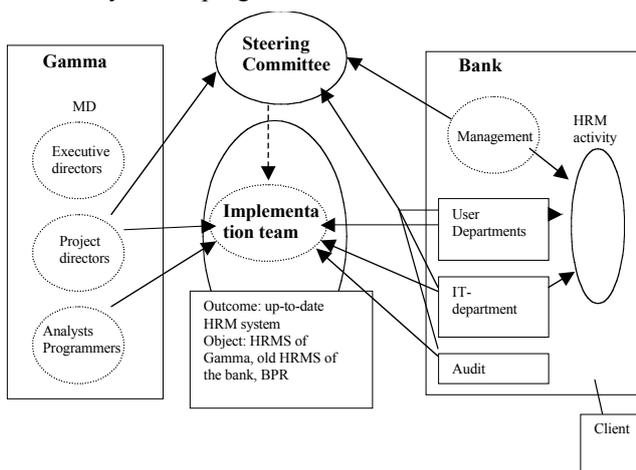


Figure 3: Project organization of HRMS project

**Actors.** In Gamma, 3–5 technical persons were involved in the project at various times. All technical staff had at least a Bachelor's degree in Computer Science, some had post-graduate degrees. Since the company is also a training

institution, they made use of the opportunity to train their staff on relevant courses in-house, and occasionally outside.

According to the executive director, it is important that the people in the implementation team have 'right skills'. It means that Gamma expected users to be experts when it comes to their work. The customers expected that people from Gamma are experts when it comes to technology and application. Both of them should be cooperative, capable to communicate with each other. The customers should also be willing to learn new technologies. The expectations are not always met. The team leader's role is very crucial and his attitude sets an example to other users.

**Work process.** Usually projects are supposed to begin with a requirement analysis, but in this case the requirements were initially based on the old system although mid-stream, unexpected new set of requirements were given as a result of the on-going BPR. The development and customisation of the HRMS was like a rapid prototyping done module by module. They started by converting their own system to the client's environment. The IT professionals in the bank were helpful as they knew the functionality of HRM and understood computerization work quite well. The modules went through two test phases. First there was a quality assurance test in terms of documentation, screen design, error messages etc., done by Gamma. Then the users tested the modules. In that phase there emerged new requirements for some of the modules. After acceptance the modules were moved into the production environment and the implementation was started. The most demanding aspect in the implementation was to get the old data into the new system, because of the lack of compatibility between the old system and new system. Data in electronic form was transferred using scripts. But the computer crashed in midstream, and they lost some of the data since there was no backup. Thus they had to 'fill the gaps' in the system manually, and that was slow.

During the first interview in the end of 1999, the system was in use, but the users were still in the process of getting the data into the system. In addition, since the BPR was still on-going, there was one module under construction. The situation hasn't changed much within a year. The second interview was done in the end of 2000, and the system was still in the implementation phase because of the new requirements. The process had taken longer than was initially expected. Accordingly, Gamma had given only quick training to the key staff that were involved in entering data into the environment.

**Means of work.** The technology used by Gamma is Oracle, Unix and Java with Windows workstations. Medium and small customers use SysBase and Microsoft databases because of the cost factor. Software development is based on high-level tools like Oracle Designer, Visual Basic and JBuilder. Standards for screens, security, on-line help, documentation, reports, etc. are more heavily used when they are developing from scratch – in customizing an existing package the tools guide the work. Database design is considered very crucial; they have very strong standards in that area. On project management methods they don't

want to be too strict, just to keep the project on track, “a few guys who come together and get things going”. For technical documentation they have specialists who are responsible. The bank had some standards for the interface, security, integrity (company standards) and on-line help.

**Means of coordination and communication.** Communication and coordination in the project was very much handled by meetings. The project plan was the main document driving the process. The communication and coordination within activities and the means of networking between activities are presented in Figure 4.

The steering committee had meetings once in a month, sometimes every two weeks. In their meetings they checked the project situation against the project plan. The project team leader was the official link between the steering committee and implementation team. The consultant, who was the project leader in Gamma, was a link between the steering committee and Gamma.

The implementation team met regularly, approximately once a week formally. In case of a critical situation there might be sub-meetings, not for the whole group but the people concerned. Besides meetings, design and programming documentation distributed the knowledge among the group.

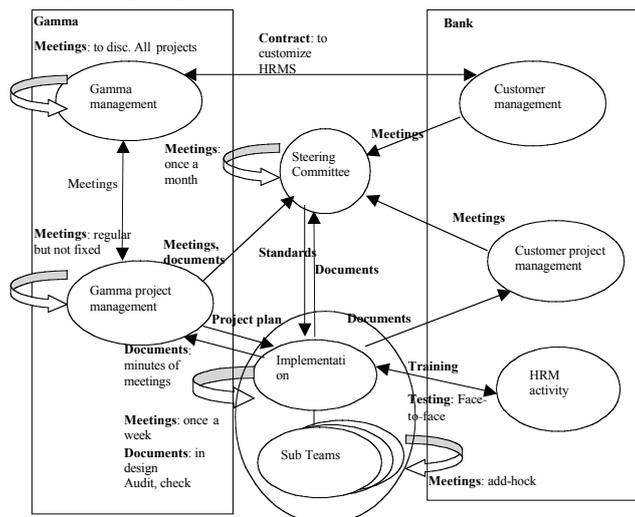


Figure 4: The means of networking *between* activities, and the means of coordination and communication *within* activities

The implementation team also met with users in testing and training situations. Usually a person who had developed a module took care of the training since he/she knew the situation. Only in special training Gamma used its training unit.

There were some checking points, ‘*crucial points*’, when the audit representative came and audited the whole process in terms of requirements, security and so. Still, in this case the auditing was not very formal.

**Mode of operation.** The project was organized to have very clear structure. Even if the operation was quite

hierarchical, it was very cooperative. The key persons met regularly and also informally in order to keep things going. The whole package was divided into modules so that the process of implementation and customisation was easier to control. Figure 5 presents a brief summary of the elements of the key activities of information systems development work by the implementation team and the steering committee.

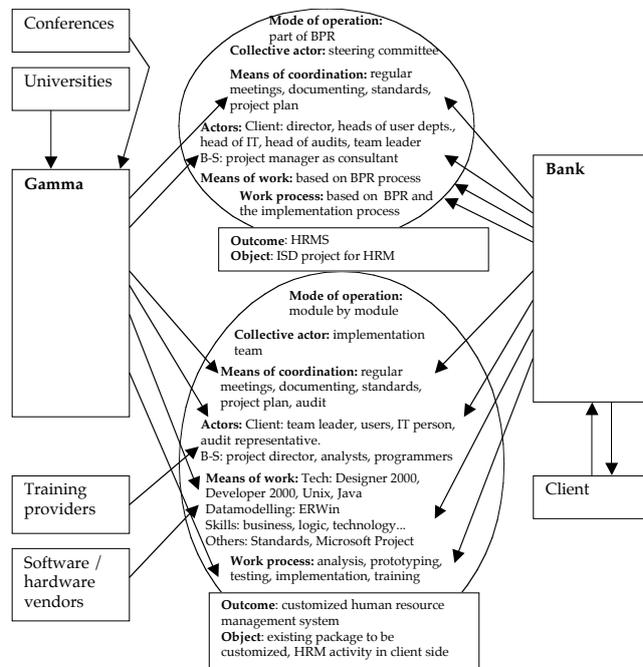


Figure 5: The elements of activity of the implementation team and the steering committee

Historically, Gamma experienced two main changes in the 1990’s. First, there was a shift from Cobol to databases in 1991. By that time the number of personnel had risen from seven to about 30 people. The other major shift was the movement to a client server environment in 1995. Both of these changes were triggered by their clients.

In summary, Gamma is a rather good example of the locally-owned small software company in Nigeria (c.f. survey of Nigerian software companies by Soriyan 2004). Nowadays Gamma pays a lot of attention to the high-level technical tools and standards of systems development as well as staff training. Flexibility and ability to react fast are regarded as a competitive asset, both in terms of the software technology and the project procedures. User satisfaction is highly appreciated, and managed through involving them closely in the projects.

**Risky business: problems and contradictions**

The main problems during the project were partly project-dependent, partly more general in nature.

Concerning the **object** of the project, human research management, problems are related to an old system and new requirements. The existing HRM system developed by Gamma was converted to client’s environment and customized to their initial needs. Thus the requirement

analysis was based on the old system, even if the bank was in the process of business process re-engineering. The bank was too eager to install a new system before a proper requirement analysis was done. That caused extra work during the process, since many modules had to be re-developed, and the final solutions of some modules were still under construction after long time. The old system as a basis for requirement analysis seemed to be a mistake. The initial schedule was unrealistic and all the dead lines were passed.

The project organization (**collective actor**) was quite normal for the Gamma, with a steering committee and an implementation team, both headed by the customer. Usually the role of the steering committee is strong, but in this case the executive manager evaluated it quite weak. This can be caused partly because of the on-going BPR process in the bank and its management, and partly because of inexperience of the management to realize the embedded changes caused by new information system. In addition, there were other stakeholders involved in the BPR, who affected the decision making of the client's management. One of them was an external consultant. Thus the situation was not very clear. Also the users were inexperienced to analyse their needs in the beginning of the process. This led to the situation of new requirements during the project. Gamma wanted to make the client happy and to get a good reference, so it wanted to be flexible.

What comes to **actors**, concerning the analysts and programmers in the Gamma, the main problems in general were in their defective skills and knowledge. University education is lacking behind the requirements of the companies, when it comes to new methods and trends, but also, according to the executive director, in basic skills like database planning and logical thinking needed when designing and analysing object activities. It takes time for the company to train people for these important tasks. Gamma was (and is) very careful who to send to meet customers. Inexperience of the users for information technology can easily lead to problematic situations. In this bank case, the problems were mostly caused by the inexperienced users to decide what they really wanted.

Problems concerning the **work process** were quite the same that has been discussed when analysing the problems with the object, focusing on the requirement analysis. New requirements had prolonged the process, and the plans and dead lines were not met. In addition, entering the data into the system faced problems with integrity and a machine crash, thus causing impatience among client's management. Also the training phase had not been started properly, because of the unfinished modules.

The **means of work** concerning technology, standards, skills etc. were not that problematic in this case. The possible deficiencies of domain knowledge by the Gamma people were helped by the experience of the IT department people in the bank.

The **means of coordination and communication** were organised as usual, and there were no obvious problems. Still, the role of the steering committee was not strong

enough to keep the coordination in hand, related to whole BPR process. The **mode of operation** was also organised as usual, with no obvious problems.

The problems faced during the process concentrated mainly to under specification of requirements and inexperienced users and their management. If we take more contextual or social viewpoint, the most visible problems that affect the business directly are *poor energy supply*, and *erratic and unreliable communication network*. Companies must put a lot of effort to guarantee steady electricity generation; they need generators and stand-by generators. The lack of fuel forces them to buy fuel from the black market, which is extremely expensive. Unreliable communication networks block a quick access of people and makes it difficult to arrange activities.

The *lack of resources for IT investments* is a problem of many organization in the society in question. Mainly big companies can afford to invest in software applications, e.g. banks, insurance companies and oil companies. Smaller companies have to struggle to survive and they don't have enough resources for IT investments. Thus software business in Nigeria is dependent on the big and/or international companies for the moment. In addition, it is quite difficult to find new customers, since data about the domestic market is insufficient. The *absence of official IT policy* by the government (so far) has not helped the development of a local software industry. The IT investments are incidental and the contracts are often based on relationships: '*man knows man*' is a Nigerian saying that influences business arrangements. In addition, corrupted managers or officials, or employees inclined to misuse the property of the company, can resist new systems and even sabotage the development process.

The competition is hard because of *importation of foreign packages*. This is of course true all over the 'globalised' world, but for example in Nigeria it is difficult to compete with well-known foreign companies. Firstly, it is difficult to find people who know the latest technological trends since they are not taught at universities. The latest technology and books are also sometimes difficult to get, even for a software company, and they are expensive. In addition, even if the first computers to Nigeria came in 1960s, the history of indigenous software development is short. Like the executive director of Gamma said '*we have no tradition in computer business*'. There have been cases where business ethics could have been better; the vendors had more or less a principle 'sell and run' (c.f. Korpela 1994). Thus many possible customers still prefer foreign rather than local software. Nowadays there is a registered association named Computer Professions of Nigeria (CPN). Also CoAN (Computer Association of Nigeria) is promoting the reputation of local software vendors. But '*Rome was not built in a day*', as our interviewee said.

The society at large around the company matters. The *political and economical history* gives root to the enterprise. The embarkation of Structural Adjustment Program (SAP) in the 1980s influenced the beginning of computerization in Nigeria, even if the program itself was not a success. According to the software company, being a

dictatorship or a democracy has not been affected directly the business. But in general the *environment* has its own meaning with e.g. overall safety and openness, possible riots or conflicts, general atmosphere, bureaucracy and so on. For example, the more governmental the partner is, the more difficult and bureaucratic the activity is.

## CONCLUSION

The ActAD model presented in this paper gives a basis for a method for analyzing the present state of an activity. The model can be used as a checklist when analyzing one's work activity. The structural analysis of an activity is recommended to conduct in a focus group session or a workshop. The basic idea in this paper is that work development is done within information system development process, in cooperation with users and designers, but surely the method can be applied in pure work development process as well.

In the paper we present one example where the method has been used as a research method when analyzing information system development activity in Nigerian software companies, a topic that has not been studied before. The example illustrates how the method works in activity analysis in practice. The method proved to be quite useful in guiding a semi-structured interview, describing an activity, and indicating problems. However, mainly because of a bit troublesome context and time limits, interviews and analysis remained descriptive and general, without a deeper action level analysis or work development efforts.

The ActAD method is a core method of applying the ActAD model of concepts as a lens. The method applies the concepts of activity theory and considers an activity as a systemic entity. The method presented in this paper is general and focuses more on contextual issues, 'zooming

out' as the case example illustrates. The more 'zooming in' applications can be found from papers by Toivanen, Häkkinen and Riekkinen in this same workshop. All these papers are connected to each others, and the connection is explained in the paper by Korpela.

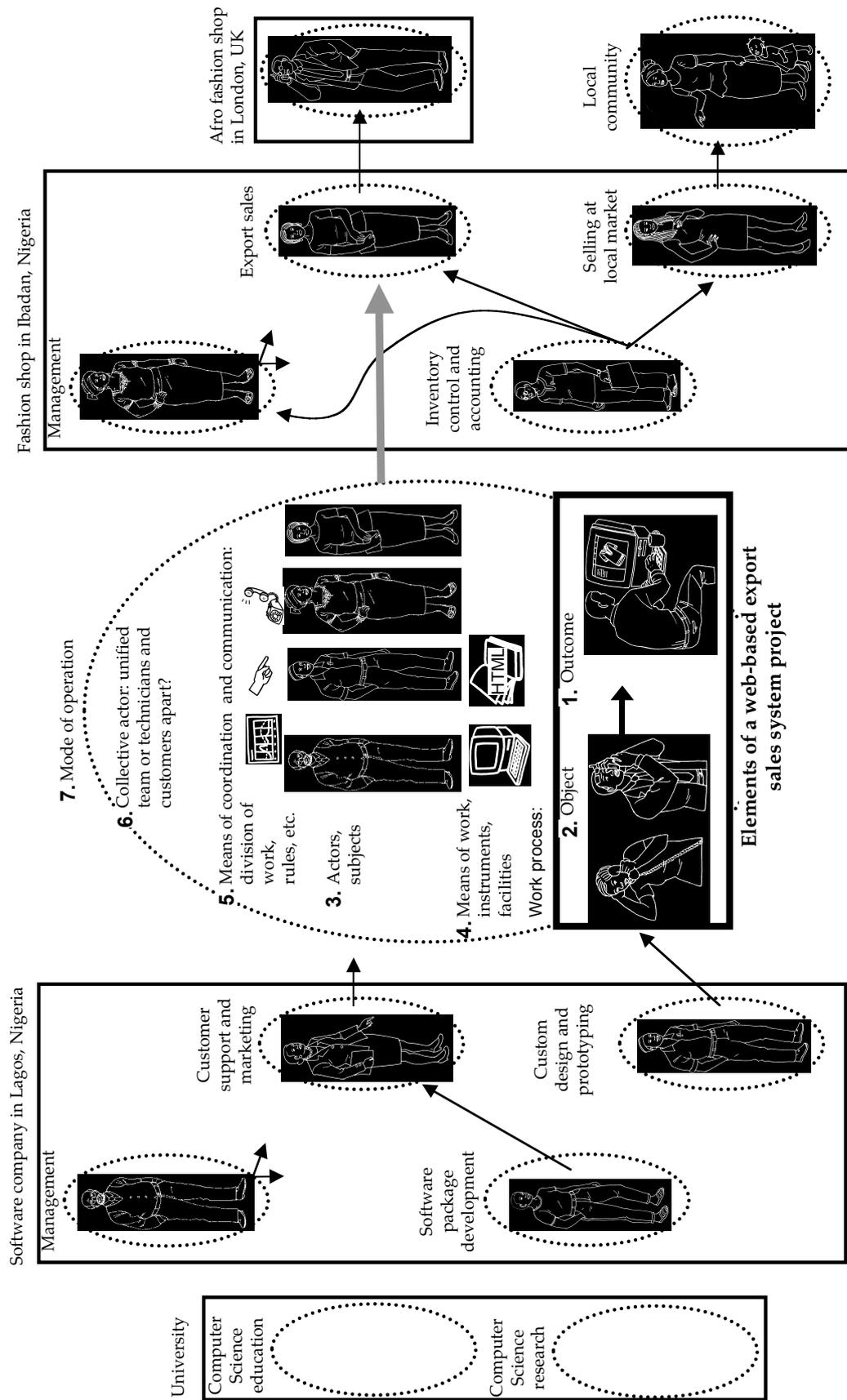
## ACKNOWLEDGMENTS

The paper is based on research funded by the Academy of Finland through the INDEHELA-Methods project no. 39187 in 1998-2001. I also thank Dr. Mikko Korpela, University of Kuopio, Finland, and Dr. Abimbola Soriyan, Obafemi Awolowo University in Nigeria, who participated in the project.

## REFERENCES

1. Engeström Y. (1987) *Learning by expanding*. Helsinki: Orienta konsultit.
2. Korpela M, Mursu A, Soriyan A, Eerola A, Häkkinen H and Toivanen M. (2004), I.S. Research and Development by Activity Analysis and Development - Dead horse or the next wave?, *Information Systems Research - Relevant Theory and Informed Practice*, Kluwer Academic Publishers: Boston, 453-471.
3. Mursu A, Soriyan HA, and Korpela M. Risky business - A case study on information systems development in Nigeria, in *Proceedings of the seventh IFIP WG 9.4 Conference on ICTs and Development : New Opportunities, Perspectives and Challenges*, (Bangalore, May 2002) Indian Institute of Management: Bangalore, 385-401.
4. Soriyan H.A. (2004), A conceptual framework for information systems development methodology for educational and industrial sectors in Nigeria, Dissertation. Obafemi Awolowo University, Ile-Ife.

Appendix 1: Illustration of a sample ISD project, used as research material in Nigeria (Korpela et al. 2002b, drawings Reetta Korpela).



<p>1. <i>The company:</i> Brief description and history of the company.</p>
<p>2. <i>The project:</i> What was the project about? What was the <i>outcome</i>? Who was the customer? For what reason they needed your service or product? How did you meet with the customer? How was the <i>object</i> of the project defined? Short description of the <i>process</i> (phases, schedule etc.)? How was the project organized? What were the roles and responsibilities of each team? How formal was the project structure?</p>
<p>3. <i>People in the project:</i> How many people were involved in each part of the project organization? How many men and women? What were their roles and tasks and responsibilities? What was the educational background of the members? What about experiences? Age distribution? What kind of people you want to employ? What is the procedure and how do you find these people? Does the company provide further education to employees? What kind? What other sources you use for getting knowledge?</p>
<p>4. <i>Working in the project:</i> What kind of professional knowledge or business knowledge was required? What kinds of tools were used? What kinds of methods were used? What kinds of standards were used? Who provides the tools, methods or standards? What was the level of documentation in each phase of the project?</p>
<p>5. <i>Communication in the project:</i> About meetings; how formal, how regular, who participates, what kinds of issues were discussed and decided? What about other, informal 'meetings'? How decision-making was divided? How were tasks distributed? Did you have any methods or technical means for communication? What was the reporting system? How was feedback distributed? Did everybody know each other in the project organisation? Did everybody know each others' roles? Did you meet outside the project?</p>
<p>6. <i>Communication outside the project:</i> Do you think that the user participation was adequate? How did clients participate the process? How often did you meet and why? How much did you rely on the customer in business issues? How was user training organized? What other stakeholders were involved? What was their roles? How did you communicate with them? How did different stakeholders impact the project? How would you describe top managements' role in the project?</p>
<p>7. <i>Sustainability:</i> Is the application in daily use now? Who is using it? Does it improve their services? How critical is the application for the company? How did you evaluate the appropriateness of the technology for the user organization? Did you give any support after the project? Do the customer have an IT department or IT experts? How experienced were the users for the new technology? What do you think are the most important issues for sustainable use of the new system?</p>
<p>8. <i>Overall:</i> What was good in the project? What would you do better or differently? What were the major problems? Anything relevant that we have missed? Any questions?</p>

# IT needs exploration

Marika Toivanen

Department of Computer Science, University of Kuopio

P.O. Box 1627, 70211 Kuopio, Finland

marika.toivanen@cs.uku.fi

## ABSTRACT

In this working paper, we describe how to utilize an activity-theoretical approach in scantily understood domains (for example multi-professional and multi-organizational domains). Our method is aimed for “requirements exploration” in previously weakly understood settings, e.g. when several uncoordinated activities take place between and outside of formal organizations. Thematic interviews and observations are used in order to produce descriptions of current activity networks, services and processes. Our method rises to the most important challenges of development within or between activities. An outline of the desired state can be achieved, for example, by organizing workflows in new ways or by introducing new pieces of software. We also present results achieved while applying this method to the home care case.

## Keywords

activity network; domain specification; IT needs

## INTRODUCTION AND MOTIVATION

The aim of our research is to develop a method for “requirements exploration” – gathering, structuring and describing requirements for information systems in previously weakly understood areas. The method should be useful in grasping the network of activities and the information needs specially, in a so-called gray area. The gray area means a hazy, unclear or unexplored area or problem domain. The holistic nature of the method is particularly important, since technology is not the only solution to the needs that will be discovered.

In this paper, we describe how information requirements can be explored in a scantily understood domain by using an activity-theoretical approach. The goal of our method is to make *the path from activities and processes to software requirements* more transparent and efficient, and easier to understand, by actors in domains, than in other existing methods. Activity-orientation takes care that new technology is introduced only if needed. The software is not designed before describing where it is useful, economic, and needed.

The key issue in this method is to understand the essence of the domain. Figure 1 depicts the sketch of the method. First, outline the domain as an activity (or activities), what elements are in and what are outside. Second, describe the information system in that activity (or in the activity network) and identify targets for developing. Third, if it is possible to achieve a better workflow with some software, then specify the requirements for that software.

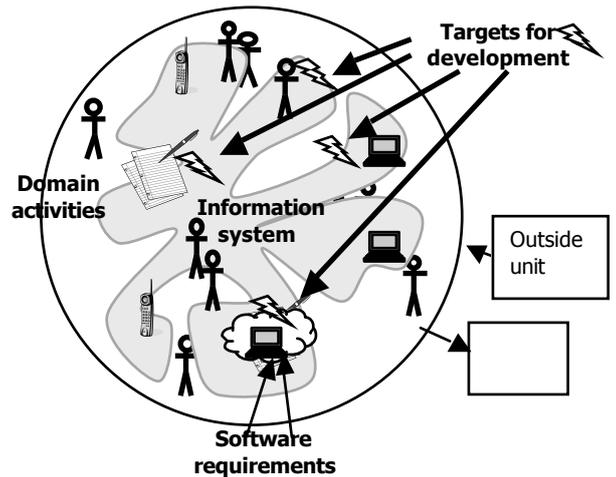


Figure 1. The problem domain can be seen as an information system within an activity network. Software requirements are specified according to targets for development.

## WHEN THE METHOD IS SUITABLE AND BY WHOM

The method is useful for early phases of design process, especially, for early requirements analysis, feasibility study and overview outlines. It is handy for gathering, structuring and describing information needs.

The method provides a holistic overview on the domain. From this overview it is possible to proceed into individual activities and specify them more profoundly.

The method is suitable for many kinds and especially for scantily understood, large, complex and unclear areas or systems. This method is advantageous if, for example, following questions arise: What systems are needed e.g. office automation systems, CAD/CAM systems, mobile systems, CSCW systems or something else? Do we need improvements in the work organization instead of new software?

The framework provides an opportunity to apply different viewpoints such as service chains, activity networks, and organizations involved in the services. The activity network consists of activities. Each activity is described in detail. The most essential functions in the organization and the needs for computer-based means of cooperation, communication and networking are emphasized.

Technical (computers, applications, databases, mobile applications) and non-technical elements (actors, papers, notes, bulletin boards and discussions) can be included in the descriptions, too.

Our method can be applied in in-house development, re-design and participatory design processes, especially when grasping the previously weakly understood areas. Systems analyzers and users can apply our method in co-operation. Not tested, but probably our method facilitates designers, usability specialists, and consultants.

Project managers (with users) of information system development projects can apply method to analyze, design or introduction of new software [with education, teaching, guide, handbook or mentor, who knows how take advantage of the ActAD framework].

The framework of the method provides a tool for communication and negotiation. This is suitable when the goal is comprehension of potential properties and requirements of future information systems and their integration. Hopefully, that fruitful communication leads to mutual understanding between different stakeholders, including service providers and customers, software vendors and software integrators.

Resources needed depend on complexity, wideness and exploration level of the problem domain. Targets for development is significant, too. This method is suitable for different domains.

### OUTCOME OF APPLYING THE METHOD

The outcome of applying the method is an answer to the question: *where are the IT needs?* You can find and point out spots of IT-needs in the domain and get insights into the application domain, including preliminary requirements specifications.

The outcome of the method has three purposes:

- Achieve the description of *the current* activity network, including services, processes and the software. This is for understanding the domain. (See, Analyzing the present state of an activity, by Mursu in this workshop)
- Find out *developing points* in the current activity network. What are the IT needs and developing points in the current information systems.
- Outline *desired states* in developing points and sketch how to achieve them.

This kind of outcome is needed in developing projects to identify information needs, to find out improvement possibilities, and to select the right way to carry out them. From the activity-oriented descriptions *the current state*, *the developing points*, and *the desired state* can be discovered. The desired state can be achieved by organizing the work in new ways, by maintaining software, or by introducing new pieces of software. Usually, these all are needed and they overlap with each other.

After figuring out the desired state continue, for example, in following ways: If work organizing is needed, then

describe action flows and change scenarios. If new pieces of software are needed, then describe software requirements.

The form of the outcome is mostly like descriptions in texts and diagrams which should be understandable by participants. Rich pictures are used to clarify texts. In addition, it is helpful to use pictures on a wall when discussing about the results, for example, in review and evaluation phases.

### SHORT INTRODUCTION TO THE METHOD

While elaborating this explorative method the basic default was that both *processes and data items* are needed to be considered. Another contribution came from *a path idea*. The path idea means figuring out paths from domain understanding to software specifications.

The path begins with focusing on the domain, to which we selected Activity Theory -based (Engeström, 1999) and Activity Analysis and Design framework (ActAD) (Korpela *et al.* 2000, 2004). Process chains modeling and information clusters descriptions are used in order to get detail descriptions of the problem domain. From software requirements specifications we selected use cases, including scenarios, and entity-relationships models, or conceptual class diagrams, which can be described by UML. For more details see Toivanen *et al.*, 2003. In this paper, we focus on the beginning of the path from domain understanding to software specification.

The method is iterative and has three phases:

1. Structuring and describing the problem domain
2. Aiming and focusing on the targets for development
3. Focusing on the software specifications

Depending on the problem domain or the targets for development applying the method can be stopped after any phase.

### PROCEDURE

#### I Phase: Structuring and describing the problem domain

##### *Feasibility study*

First, make a literature review on previous research on your domain in order to provide an initial understanding of the activity network around that domain. Keep collecting data about other projects around the domain during the project. This is helpful when thinking what can be done in developing points of your case.

##### *Interviews*

Find and select interviewees (the actors of the activity network), use your first sketch of activity network.

The following interview themes should be used:

- What are the objects, goals and outcomes of the domain (from both the collective and individual viewpoints)?

- What are the means of work, i.e. physical or mental instruments or facilities that are used in the actions?
- Who are the actors and organizations that one work with and what are their roles? Which professions, authorities, and abilities are present? Where do they come from? What kind of professional training is needed?
- What are the means of coordination and communication in the domain? What kind of rules and division of work affect daily activities? Where and how are the rules created and accepted? What kind of instruments is used in communication and how are they used? Where do they come from? Is information and communication technology (ICT) used, and if yes, what is its role?
- What are the means of networking? How are one's activities related with other activities in the network? Where are organizational boundaries and how do they affect the service chain?
- What is the collective actor like? What kind of groups or teams are there and what are their aims, responsibilities and functions? How are these teams formed and who comprises them?

Use the language of interviewees always when possible. Use scenarios to discover all essential elements, *i.e.* narrative descriptions of a customer case and the activity network in it. The scenarios can be based on literature. Remember, interviewee is an expert in the domain.

Documentation of interviews is cumulative until last interview has been made.

### Analyzing

Zoom in, zoom out, and be open.

Gathered information can be grouped, related, composed, decomposed, generalized and specialized, while producing general overviews or zooming into special cases or situations. In addition, clusters of information are emphasized as *concepts of work* and relationships between them.

Hopefully, you recognize patterns and relate the elements to the wholeness, *e.g.*, to find the chains and networks of services.

The present state of the domain should be sufficient thoroughly analyzed and the problematic areas pointed out. Further, it is worth to evaluate and discuss whether problems should be solved by, *e.g.*, organizational changes, staff training or new technology.

### Cross-check

After interviews and analyses, the results should be cross-checked with the actors. A practical tool for this is workshops of interviewees. Organize those workshops carefully beforehand.

### IT Needs

The problem domain is modeled as a *network of activities*

within and between organizational boundaries. The key activities are described in more detail. Finally, the outcome is "IT needs" -spots in the activity network of the domain described in documents and a poster.

## II Phase: Aiming and focusing to the targets for development

Focusing to the whole activity network is not effective. After developing points have been identified, the analyst should focus on those points. Work processes and information architecture are studied in parallel.

### Work processes

This method proceeds from activity networks towards software engineering as follows (Figure 2.)

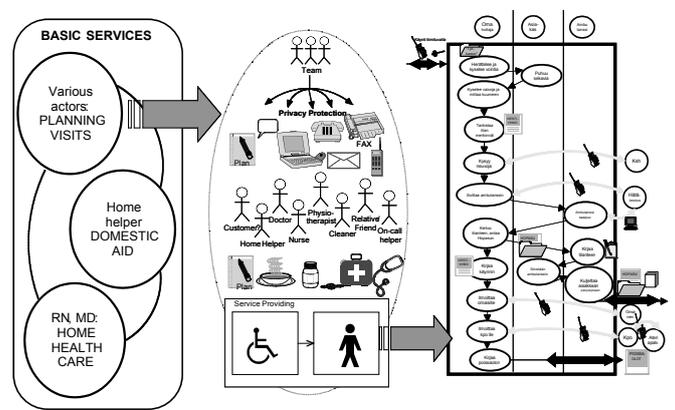


Figure 2. In the first phase, the activity network and selected activities are described in detail. In the second phase, processes of selected activities are described in order to understand processes and recognize lacking software.

The process of an activity shows how the object is transformed into the outcome through actions. This process is described in terms of actions of various actors over time. A combination of the activity framework and UML's "activity diagrams" and use case diagrams are proposed to be utilized in this task (the term 'activity' is used in UML, Unified Modeling Language, in a slightly different meaning than in AT). The purpose is to explore the *information needs* in more detail, and to identify "empty spots" for potential software; *i.e.*, the needs for currently lacking computer-based means of action, coordination or networking (applications or components).

### Information architecture

Answer to the questions: Where are the pieces of the needed information? What are concepts, structures and relationships between those pieces?

### What is done around targets for development?

What are the developing points in the current information systems? *The developing point* can be any element in any part of activity network, even some small elements attached to any part or context of network. It can be inside or outside of single activity or between activities.

Check your data collection, started in Phase I, if there are other development projects struggling with same targets. Clarify overlaps, dependences, and common resources in these projects.

### III Phase: Focusing to the software specification

In this phase the analyst specifies the *requirement specifications* of each piece of software identified in the previous phases. Here, the analyst utilizes architecture and action descriptions. Use cases, class diagrams, and sequence diagrams are useful, too. Not bound, but flexible.

The software requirements specifications should be in such a form that it is possible to utilize them in software design and implementation. Furthermore, actors should have a possibility to inspect the requirement specifications by themselves. Hence, the requirements must be presented in a clear and simple way.

More about this phase, see Riekkinen's method in this workshop.

#### CASE EXAMPLE – HOME CARE AS A DOMAIN

Our domain focus is in the health care, where information systems are usually developed for a single organization, or for information flow from one organization to another. In our case domain, multi-professional and multi-organizational home care, this approach is not very helpful. Moreover, home care takes place in “no-man’s-land”; outside the organizations’ infrastructure, in the customer’s home, and chances of technology are used scarcely.

Our task was to develop methods for grasping a previously unexplored “gray area” of inter- and extra-organizational activities, in search for requirements for possible new software products and their interfaces (Toivanen *et al.*, 2004). Home care was selected as the case domain. Our *explorative requirements analysis method* begins with developing a rough overview of the activities, networks and organizations involved, in order to identify major stakeholders. The number of social and healthcare professionals and other parties involved in providing home care services in this case study was astonishing.

The concept of *home care* includes both home help services and home health care. The home help services can be further divided into household aid and supportive services, such as transportation. Municipal health authorities have traditionally taken care of home health care, whereas home help services have been the responsibility of the municipal social welfare authority. Other actors in the field are, *e.g.*, the customer’s family, private services, and the non-governmental organizations. Earlier studies have identified shortcomings in the information flow and a need for better organized visits to the customers’ homes, so that the entire service system would support the customer’s wellbeing and autonomy.

#### First step – the overview

First, a literature review was made on previous research on home care in order to provide an initial understanding of the activity network around home care. The ActAD framework was then used to find and select interviewees (the actors of the activity network).

Representatives of key stakeholder groups were then interviewed, using a semi-structured interview guide that followed the activity framework but utilized terms specific to the home care domain instead of abstract terms such as “actors” or “means”. On the basis of the interviews, preliminary descriptions of the findings were produced and structured into documents and rich pictures on a wall (Figure 3). The findings were presented, considered and processed forward with the stakeholders in workshops.

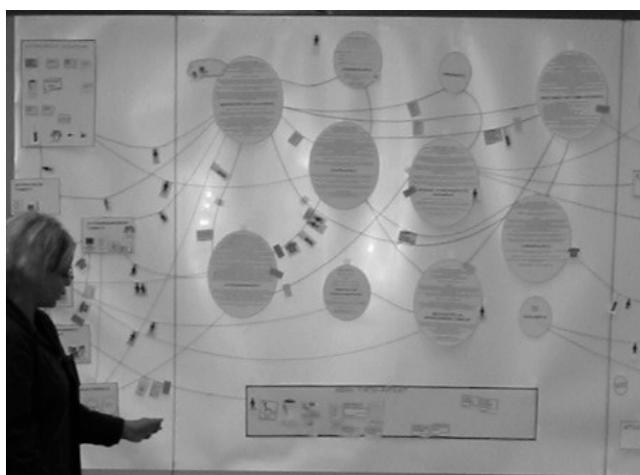


Figure 3. Rich pictures on a wall in a workshop

After the workshops, revised descriptions of the information needs in the *network of activities* (Figure 4) were constructed. The outcome was depicted as a systematic description of the activity chains in the domain. Core activities were identified and described down to the level of actions, where the “empty spots” (lack of software) and requirements for new software and their integration with legacy systems can be identified.

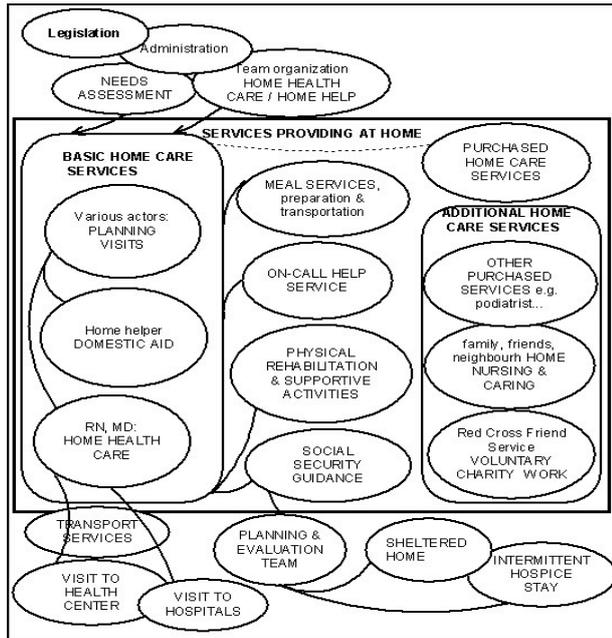


Figure 4. Example of outcome: Activities and their relations or network of activities

### Second step – focusing to the developing points

We focused to the main activities and described their processes in more detailed level. Using the acquired understanding, the purpose and the advantages of the would-be software and the first sketch for its architecture were derived. It was important to check also, what is happening in a municipal and in a governmental level and things like legislation, requirements of statistics etc.

### Third step – software requirements specification

The software specifications for IT needs were identified in this step. One of the key activities in Figure 4 is the planning visits activity, where the outcome is the plan of home care. We described possible use cases of the planning visit activity, composed software components needed. We also made sequence diagrams, which present interaction between existing information systems and the software components.

### CONCLUSION

The key tool of this method is ActAD framework (see academic reflection). Moreover, thematic interviews and

documentation of the interviews by modeling the domain as a network are needed; workshops and pictures on a wall are helpful to assess the outcome.

### ACKNOWLEDGEMENTS

I gratefully acknowledge the contributions of Heidi Häkkinen, Pertti Laitinen, Irmeli Minkkinen, Päivi Röppänen and Tuula Tuomainen in the PlugIT home care research group as well as the Kuopio Social and Health Center. I also thank my supervisors Anne Eerola and Anja Mursu for comments and criticisms.

The paper is based on research funded by the National Technology Agency Tekes through the PlugIT project, grants no. 40664/01, 40246/02 and 90/03 in 2001-2004.

### REFERENCES

- Engeström Y, Miettinen R, and Punamäki R, eds. *Perspectives on Activity Theory*. Cambridge, UK: Cambridge University Press, 1999.
- Korpela M, Soriyan HA, and Olufokunbi KC. Activity analysis as a method for information systems development: General introduction and experiments from Nigeria and Finland. *Scandinavian Journal of Information Systems* 12(1), 2000, pp. 191–210.
- Korpela M, Mursu A, Soriyan A, Eerola A, Häkkinen H & Toivanen M. Information systems research and development by Activity Analysis and Development: Dead horse or the next wave? In: Kaplan B, Truex DP, Wastell D, Wood-Harper AT & DeGross JI, eds. *Information Systems Research: Relevant Theory and Informed Practice*. Boston: Kluwer Academic Publishers, 2004, pp. 453-470.
- Toivanen M, Eerola A, Korpela M. From information systems requirements to software components - home care case. In: Sanna Laukkanen, Sami Sarpola, eds. *Electronic Proceedings of the 26th Information Systems Research Seminar in Scandinavia (Proceedings of the 26th IRIS)*, Haikko Manor, Finland, August 9-12, 2003.
- Toivanen M, Häkkinen H, Eerola A, Korpela M, Mursu A. Gathering, structuring and describing information needs in home care: A method for requirements exploration in a "gray area". In: Fieschi M, Coiera E, Li Y-C, eds. *MEDINFO 2004: Building High Performance Health Care Organizations*, San Francisco, 7-11 September 2004, p. 1398-1402 Amsterdam: IOS Press, 2004.

# Acquiring information about IS development needs from the viewpoint of an organisational unit

*Heidi Häkkinen  
Department of Health Policy and Management  
University of Kuopio  
P.O. Box 1627,  
FI-70211 Kuopio  
Finland  
heidi.hakkinen@uku.fi*

## ABSTRACT

Information is maybe the most important common factor in an activity system. Sometimes there is a need for producing preliminarily information about an activity system, the state of its information system (IS) and the most important development needs. This information may be needed quite rapidly for informing and guiding further information system acquisition or system development. A tentative description of the activity network, information needs, and user requirements should be acquired before any major changes are planned. Relatively rapid, yet participatory methods are needed in this preliminary stage.

In this paper we present a relatively rapid and low-threshold method for describing the activity network, its information system and the development needs and user requirements in it. The results show how the most important needs for development can be utilised in concurrent IS development and work development.

## INTRODUCTION

In some situations a tentative description of the activity network, its information needs, and user requirements would be useful, rather than a very thorough one. Such conditions may prevail e.g. when major changes like purchasing a new software application or making some major changes in the existing ones are planned.

This is particularly the case when the activity network and its overall information system, IS, defined here as the aggregate of processes, resources and tools of information management, computer-based or not, as in [1], is fragmented into several disconnected forms, software applications, rules, organisations and actors. Same data may be collected, processed and stored by multiple means and in multiple locations. In the absence of practical and mutual information channels, true cooperation and

collaboration between different actors becomes more difficult.

Most human activities consist of a chain or a network of organisations. Yet the design of software applications and other information and communication technology (ICT) tools still supports mainly profession- specific or task-specific work. The design of new artefacts and development of the existing ones should be focused in a wider context than just one user group or organization.

At least some of the methods used as the basis of IS and IT design and development should be easily and inexpensively utilised. In many cases it is better if the preliminary assessment is produced by the organisations themselves, rather than software vendors or quality consultants. There are two main reasons for this argument. Firstly, the objective, perspective and culture of the organisation should be taken into consideration in all development - not just costs, effectiveness and streamlining of processes. Secondly, it is the actors themselves who know best what kind of information they need in their work and with what tools and means they are ready to handle it.

## The starting points of the method

The starting point and theoretical basis of the method is an activity-theoretical framework called Activity Analysis and Development, ActAD (Figure 1).

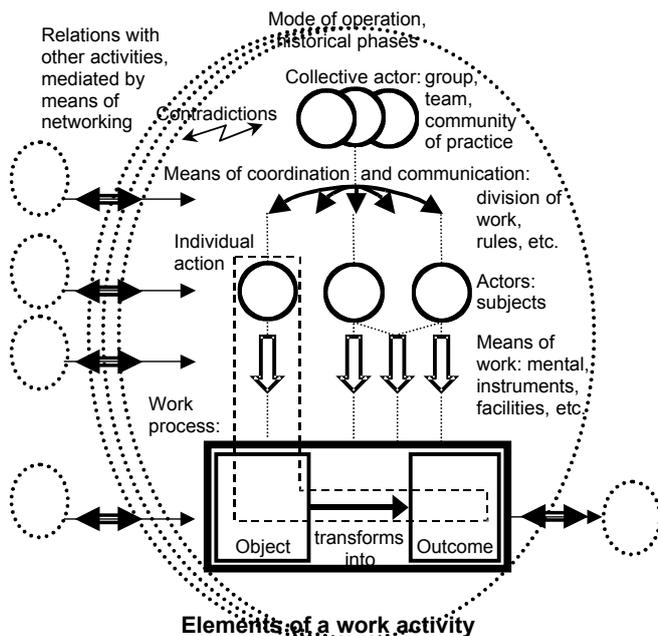


Figure 1: ActAD framework, structure and elements [2]

It is adapted from the traditional Engeström's systemic structure of work activity, as in e.g. [3]. In ActAD, the original AT "triangle" model has been remodeled in order to make it better fit IS development. ActAD has successfully been used as framework in different occasions and domain areas, e.g. [1] and [4].

### The aim of the method

The method is meant to be used in situations where a tentative description of the activity network, information needs, and user requirements should be acquired, particularly before major changes are planned or decisions about purchasing new tools are made in organisations. The ultimate beneficiary may be

The organisation should be able to utilise the results in the development of the activity, its information management and cooperation with the rest of the activity network. At least they should be used to inform and support mutual discussions and decisions concerning large investments in software applications. It should also be possible to use the results as groundwork for further software design, requirements engineering and system integration.

### Utilisation of the results

The mandator of the study may be almost anyone interested in developing the work activity and/or information management tools in it. The benefits can be directed to the service-providing organization(s), software companies, ICT solutions selling quality consultancies, or those who perform the actual work. Of course the customer, the actual citizen who uses the service in question, does also benefit,

but only indirectly. The direct beneficiaries can thus be divided into three main groups:

- The management level of the service-providing organisation: the employing, financing, organising, controlling and/or reforming agent.
- The vendor or consultancy level: an ICT vendor or software company with an interest and products (existing or planned) aimed to be used in the field of activity.
- The operative level: the representative of the employee, personnel in a single working place or company, occupational group, trade union etc.

### Prerequisites and required competencies

The method is meant to be used in the very early stage of

- development needs assessment, regardless of whether the initial idea of development is aimed at work activity or information system
- in the organisations: decision-making regarding purchasing ICT, hardware or training,
- in software companies: while designing new software, used as a feasibility study and a basis for requirements analysis.
- (in any of the above) when planning, preparing to or reacting to some major changes in the field of activity.

One of the main ideas in this method is that it should be used quite flexibly and with low threshold. A method used in tentative (and often quite poorly financed) needs assessment should not require much special skills or training. These include e.g. academic training in social research or IS research in general, or Activity Theory in specific, as well as IS research methodologies and Requirements Engineering training and skills.

A good background would be a basic knowledge of the field of activity in general and some experience in information system development [not much and not as a project leader or such - just an awareness of information management and an active participation in some project.

Another personal requirement is the will and ability to interact and communicate with people. The interviews are meant to offer a neutral and equal arena for everyone to participate, no matter what their role in the activity is. Therefore, a great emphasis has to be put on making the sessions as accessible, participatory and practical as possible.

### The outcome of applying the method

The outcome obviously depends on the starting point of the study in question. In general, the outcome can be divided into three categories:

- Description of the present state of the information system in question, including both the everyday information handling activities and the problems experienced in them. Also confirms the structure of the activity system and network.
- The nature of solutions to the problems, assessed by the actors themselves: the descriptions of the basic form and media by which the information management problems could be handled with.
- Generation of an enriched mutual understanding among the participants regarding the information system and the activity network in general. Provides organisational awareness and motivation for further information management development and work development.

### The form of the outcome

The outcome is a combination of both textual and graphic forms. Textual outcomes are the descriptions of the activity network, its information system and the problems found in it problems. Based on the transcriptions, activity scenarios can be further processed.

Diagrams and rich pictures are used to clarify the connections and needs. The rich pictures that are created in the group sessions are one example. These pictures can be clarified and condensed by transferring them to electronic form.

Also the information flows and use cases can be described, although not on a very detailed level. The central working processes can be roughly depicted as in Figure 2.

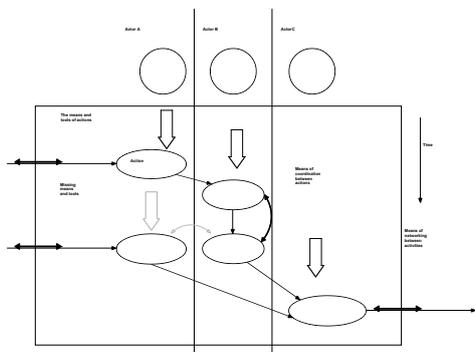


Figure 2: An example of depicting the work processes.

### WHEN & BY WHOM IS THE METHOD SUITABLE?

The settings for using this method contain preferably an aggregate of software applications and other ICT tools. Those tools may be partly interconnected, managed by various organizations and used by several user groups. The area in question should be relatively well known beforehand. This means that if there are no existing software applications or literature about the field of activity, this method is probably too light-weight and

rapid. There are better methods for handling or these “gray areas” (see e.g. paper by Toivanen in this publication).

The method can be used

- when there is a need in the organization(s) to improve customer services and working conditions
- before a new piece of software is to be designed for the domain, or before organizational changes,
- as an instrument for evaluation (after the implementation of ICT or organizational changes).

### PROCEDURE

**Firstly**, the structure and elements of the activity system (according to the ActAD model as seen in Figure 1) are roughly identified by using literature. The literature naturally depends on the field of activity. It may include, but does not have to be limited to, e.g., academic research papers, local, regional, national or company reports and statistics, various descriptions of the existing field of activity, etc.

Some information can also be acquired by observing or interviewing the actors. However, not too much time should be used into this. The point of this phase is not to accomplish a very thorough examination of the activity. It is sufficient when the activity system itself with its networking or participating organizations and other stakeholders within the activity network can be identified by name. With this preliminary knowledge the next phase, the group interviews, can be planned.

**In the second phase**, a multi-professional group is invited into two discussion sessions, each lasting for about 2-3 hours. The participants should be the actual people who work daily within the activity (as opposed to their managers). The sessions are held approximately two weeks from each other and preferably in some neutral location, at least for most of the participants.

The first session is a workshop for identifying and expressing the means and problems of information handling. The participants are asked to describe all tools and means of communication within the network and announce if there is any problem with them, great or small.

During the first session, all questioning and critique of present means and tools is welcome, but also explanations are required. The group consists of several people who may not share the same training or do not normally work together on a daily basis. The needs and problems of each and everyone need to be explicitly described, preferably with examples from the real world. The frequency and severity of problems are also assessed. Questions asked in the first session are:

- What tools does the information system consist of: what are the (both computerized and non-computerized) elements? What are the forms in which data are transferred?

- What is the state of IS: what kind of information-related contradictions are present in the work activity system?
- What are the problematic and non-problematic areas?

During the session, all findings are discussed within the group, explained with examples and illustrated with drawing and symbols on a large sheet of paper so that everyone can see it. An example of the drawing can be seen in Figure 3.



Figure 3: An example of a group drawing, situation after the first interview. The activity network with its components is depicted around its centre, in this case the customer.

The central viewpoint of the activity system is placed in the middle of the picture and the other actors and stakeholders are drawn around it. Connections (information transfers) between the actors are drawn as lines between the boxes. Sticky-notes and colour symbols are attached to the connecting lines according to the group's opinions.

After the data collection, a simplified summary of the picture is made. The picture elucidates the activity network and its information system for the participants, as well as the researcher(s) during and after the group session.

In the second session the focus is on generating solutions to the previously found problems. The solutions can present any kind of improvement: new software or hardware, software integration, new ways of networking, extra training for the staff, changing the location of people or their instruments, abandoning a custom or a rule, etc.

Questions for the second session are e.g.:

- How could the existing way of information management be changed?
- What kind of problems could be solved with software, including integration of existing

software? Describe the nature of the solution, not the technology or software application.

- In what kind of problems is software not needed or not accepted at all?
- What kind of problems could be solved by other means of work or coordination development?

In order to avoid making large amount of notes during the sessions, all discussions are recorded. The recordings can be fully or partially transcribed later, if needed. Also the drawings can be re-drawn in an electronic form, like in Figure 4.

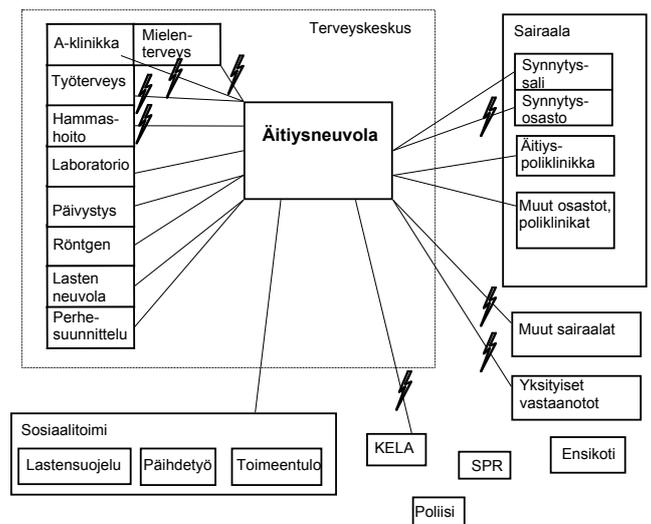


Figure 4: The same setting as in Figure 3, now depicted in an electronic form. The lightning symbolises a problem.

**The third phase** is the analysis and refinement phase. Found problems, as well as the solutions, are categorised. There can be several sets of categories. They depend largely on the main aim of the research. The existing information management may be categorised by directions of information flow, forms or tools, used frequency or criticality (priority) of the information transfer etc. The solutions may be categorised in relation to the existing tools and applications, priority of problem solving, in relation to the type of possible technology etc. The categories can vary from solution types (“Software wanted”, “No software wanted” and “Other development ideas”) to the direction of information needs, as shown in Figure 5.

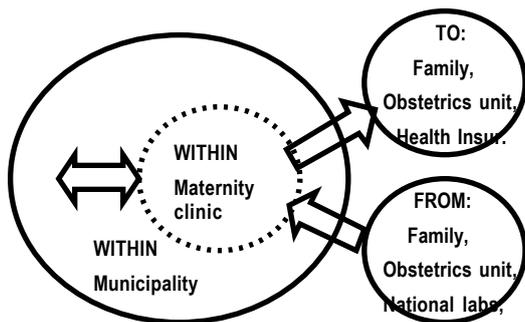


Figure 5: Results grouped according to the directions of information needs, an example from maternity care.

The form of results categorisation and presentation should not be limited by anything else except the participants' skills. However, in order to keep the method lightweight and rapid, not too much time or effort should be put into this. The most important aspect is that it is readable for those who are going to utilise the results.

## DISCUSSION

The activity system of the whole network could be developed by using this method in various organisations within the network, and by synchronising the results.

The data is collected in group interviews for several reasons. The most important is that it makes the activity system visible and concrete to several actors at the same time. Group discussions are a much more powerful way to generate innovative thoughts and collective learning and experiences of mutual understanding than, e.g. individual interviews or observational settings. Our experiences of using the method have proved these ideas correct. In the workshop, some of these experiences will be presented.

## ACKNOWLEDGMENTS

The paper is based on research funded by Tekes, the Finnish National Technology Agency, through the PlugIT project, grants no. 40246/02 and 90/03 in 2001-2004. The contribution of the research team members is gratefully acknowledged.

## REFERENCES

- [1] Korpela M, Mursu A, and Soriyan HA. Information systems development as an activity. *Computer Supported Cooperative Work* 2002; 11: 111-128.
- [2] Korpela M, Mursu A, Soriyan A, Eerola A, Häkkinen H & Toivanen M. Information systems research and development by Activity Analysis and Development: Dead horse or the next wave? In: Kaplan B, Truex DP, Wastell D, Wood-Harper AT & DeGross JI, eds. *Information Systems Research: Relevant Theory and Informed Practice*. Boston: Kluwer Academic Publishers, 2004, pp. 453-470.
- [3] Engeström Y and Miettinen R. Introduction. In: Engeström Y, Miettinen R, and Punamäki R, eds. *Perspectives on Activity Theory*. Cambridge: Cambridge University Press; 1999, pp. 1-16.
- [4] Toivanen M, Häkkinen H, Eerola A, Korpela M, Mursu A. Gathering, structuring and describing information needs in home care: A method for requirements exploration in a "gray area". In: Fieschi M, Coiera E, Li Y-C, eds. *MEDINFO 2004: Building High Performance Health Care Organizations*, San Francisco, 7-11 September 2004, p. 1398-1402. Amsterdam: IOS Press, 2004.

# Deriving software architecture from activity analysis

Annamari Riekkinen

HIS R&D Unit, University of Kuopio, PL 1627, FI-70211 Kuopio, Finland

Annamari.Riekkinen@uku.fi

## ABSTRACT

In this paper we introduce a method by which the software requirements are gathered and described by means of activity-based techniques. Further, we explain how the service-based component architecture is derived from the outcomes of the activity analysis according to Cheesman and Daniels procedure.

The method is an iterative process, which covers requirements gathering, analyzing and describing (i.e. requirements specification) and software architecture outlining. Therefore, from the point of view of the software supplier, the method is performed by system analyst, user interface designer and software architecture designer. Additionally, the database designer or other concept analyst is needed. From the client's viewpoint the users and the other domain experts are in important role during the requirements specification phase.

The method has been constructed on the basis of our software development case study, called Pakkanen, which aims at finding methods for the component-oriented software development. Pakkanen is one of the pilot projects in PlugIT-project, which focuses on software integration and its aim is to increase interoperability and lower the introduction threshold of health information systems in Finland by using open interfaces and component-based approach (PlugIT, 2001).

## Keywords

Activity theory, activity analysis, component, service, software development, requirement specification, software architecture, use case, concept model, UML.

## INTRODUCTION AND MOTIVATION

Many organizations are trying to allow Web access to existing systems or are integrating existing systems to deliver enhanced capabilities to their clients. Requirements for interoperability of the software systems can be seen as a consequence of the fact that nowadays software products and information systems are conceived more and more as the supporters of the business processes.

Today's component and web technologies (e.g. service-based architecture, middleware, component models and other standards for developing distributed software systems) provide plenty of possibilities to implement interoperable software systems, which are based on service-oriented architecture. At the moment they have widely introduced in software industry, but rather

poorly supported by suitable methods. What is not always obvious in such developments is that the business processes needed for service-based systems are usually different to the current processes (Cheesman & Daniels, 2001).

Our method is a combination of cross-scientific methods that illustrates how it is possible to proceed from activity analysis to software requirements identification and description so that the results can be exploited at the design phase. So, our goal has been on specifying the software requirements from the point of view of the following process, especially implementation design. Accordingly, the method is performed at the requirement specification phase and the early software design phase.

In order to understand the real needs of envisaged software it is essential to understand the activity domain. Also, since the development of component-based systems is based on the business processes, the activity-based approach is needed for specifying the software boundary and requirements. We adopted *an activity-based framework* (Korpela & al., 2004) for gathering, analyzing and describing the activity domain of software system. The selected method emphasizes that information systems and work activity is developed simultaneously.

In our opinion the user interface design plays an important role in concretizing the requirements for all parties in the development project. That is why we have adopted the GUID-e project model, presented by Laakso (Laakso, 2003), for user interface design. The model emphasizes that user interface design should be done at the early phase of the development project.

For architecture and component design we have chosen a method presented by Cheesman and Daniels. The method requires that software requirements are described in a form of use cases (functions) and type model (data). Both are described in UML notation. (Cheesman & Daniel, 2001) The C&D method is included in a component-based software development process framework, which in turn, is based on the RUP (Rational Unified Process).

With this method we take into the consideration the domain's working practices and their development. By means of the method we can acquire information and understanding on the scope and the domain activity of the software. The domain activity of the software means the activity that the software is part of. The scope on the other hand means all those activities that affect on the

software or vice versa. That is that scope includes domain, but it can also be larger than that.

The idea of the method is that we understand the real requirements of the software better if we understand the activity of its target domain. The method offers tools for information gathering, analyzing and describing. By means of the tools we structure and examine the activity domain. In addition, the method provides the techniques for specifying and describing the functionality of the intended software system and identifying the software components on the basis of concept modeling. The tools can be applied on different level depending on the requirement analysts' existing knowledge of the activity. In this description we assume that analyst-team does not know the activity that well. We also assume that there is some kind of technical software in use. In principle a big portion of software requirements can be derived from the existing systems but to understand the real needs of the intended software requires the understanding of the activity.

As the result of the method we have a service-based component architecture, which quite straightforwardly is derived from the well-understood software requirements. From the activity analysis viewpoint we arrive at the requirement specification that consists of a user interface specification, use case descriptions and a type model. We concretize the functions of the user interface (and software) with the help of high-level goal-based use cases. We define the goal-based use cases more closely and attach the requirements of the data and the other qualities to each use case. Type model is a description of the data that is used and maintained in the software. The model describes the concepts in details and the relations between them. Besides the requirement specification, this method generates different kinds of descriptions of activity.

### INTENDED USERS AND PRECONDITIONS FOR THE METHOD

Our method is purposed for component and service-based software development projects that aim at developing software and work activity at the same time. The users of the method are the persons in the analyst-team; requirements analyst, user interface designer, architect, test designer and technical expert. Here we mean the role of the participating actors, not the number. On the basis of Pakkanen-project, this is the best way to reach the mutual understanding and well-defined requirements. It is also essential that the experts of the domain and the end-users from the client organization take an active part in the requirement specification. It is important that all relevant actors from the domain that the software affects (such as users, technical experts and other possible domain experts) are represented.

A precondition for using the method is that the analyst-team is familiar with some field study methods (e.g. perform interviews and observations), which play an important role in the information gathering. Secondly, the know-how of the user interface design is required. In addition, the team should know the basis of the

Unified Modeling Language (UML), especially the use case and class diagram notation, by means of which the requirements are described.

### THE PROCEDURE OF APPLYING THE METHOD

The phases of the method is illustrated in the figure 1 by using the dark grey background. Before the requirements specification starts, a preliminary decision that domain needs software has been made. There is some information available about the intended software, for instance the purpose of the software, the central tasks of the software, and the essential information maintained by the software. Furthermore, at the beginning of the requirements specification phase, analysts may have different kind of information related to the domain; about existing software, people, tools and devices, working practices and even requirements for the new software. Nevertheless, requirements specification should not begin with analyzing the existing software requirements. Instead the adequate understanding of the domain is gathered in order to understand the actual software requirements.

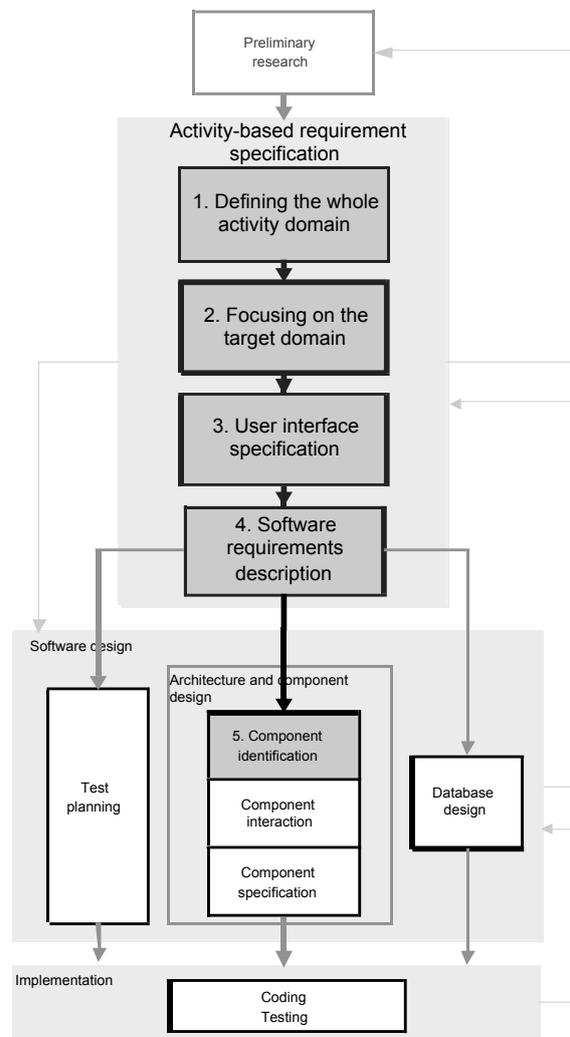


Figure 1. The method (the phases 1-5) offers techniques and tools by which the initial software architecture can be derived from the well-understood software requirements.

During the first two phases of the activity-based requirements specification, an understanding of the domain activity and the software's scope is composed. That is mainly done by analyzing the activity as a whole and by examining the most relevant work processes. Then, the actions and tasks where the software is involved in, are defined on the basis of the analyses. The results are modeled in process diagrams that describe the intended actions. At the same time with the activity examination the information on the domain is gathered into the concept model, from which, in turn, is derived a type model. The type model depicts the information that is used and maintained by the intended software. On the grounds of the process diagrams and the type model a user interface specification is outlined. The UI specification is defined and tested with the intended users or other specialists of the domain. After the client has approved the specification, the requirements are documented for the following development tasks such as architecture and component design, database design and software test planning.

The process proceeds so that first we figure out the current state. By analyzing the present activity we are able to identify the (development) needs and make proposals for solutions. The proposals will be reviewed with the client. When a consensus on the software's responsibilities is reached, the desired state descriptions are done.

Next we will present in more details how the method proceeds and what are the essential tasks and results of each phase. Even though the method is described here as a linear process, in reality the tasks and the phases are iterative and at least partly overlapping.

### 1. Defining the whole activity domain

At the first phase we form an overall picture of the domain. In other words, we garner a basic understanding of the domain as a holistic activity and, further, the role of the existing system in it (if we have one). Usually the requirement analysts have a lot of advance information on the activity domain that must be organized somehow.

In order to obtain the holistic view, we structure and depict all the relevant aspects of the activity according to the *ActAD analytical framework* (Korpela & al., 2004; see example in the figure 2). In this paper we use a term *work activity model* as a synonym for the ActAD framework. For information gathering we have adopted a *thematic interview model*, which has been formed in home care case on the basis of the work activity model (Toivanen & al., 2004). The idea of the interview model is that the interview themes are derived from the analytical framework so that the results, in turn, will be easier to link with it.

For the information gathering and analyzing all related assets, such as forms, instructions, checklists and the activity descriptions of the quality management system, are valuable. In the software re-engineering projects where, an existing system is available, the system and all documentation related to it are also an important

source of information. When the existing system is available we should, however, bear in mind that we are probably developing the system to a changed usage environment (e.g. web-enabled), whereupon only the copying the existing solutions is not enough. Naturally it is worthwhile to reuse existing assets, but while doing so, we must also understand that they are still suitable for the intended activity.

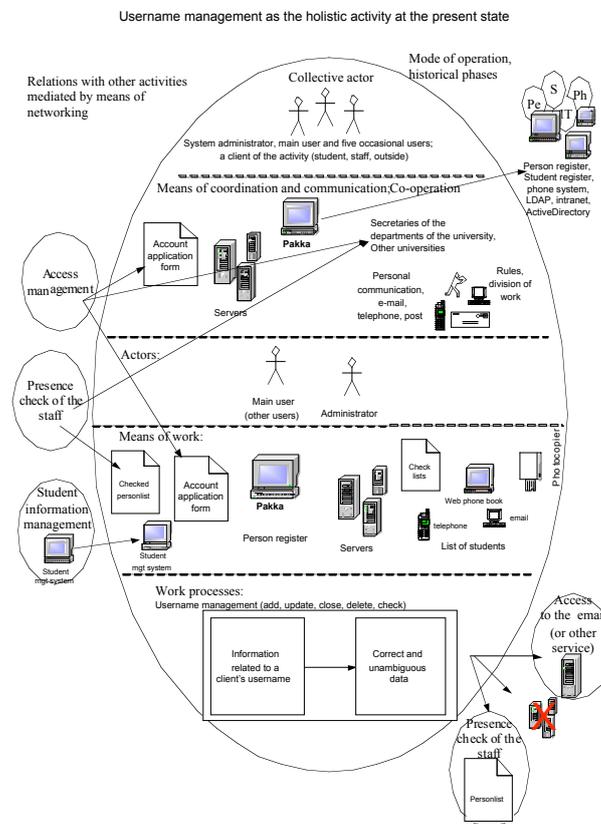


Figure 2. The whole activity - username management – captured in accordance with the work activity elements.

The main concepts of software engineering can be linked with the work activity model, for instance as follows:

- The *users* of the current system are placed in the *actor* category.
- The *functions* provided by the existing system are comparable to the user's *action*. However, when we describe user's current actions, we must not consider only on the features of the current system, but find out all the relevant steps in order to complete the action (see the phase 2: zoom-in on the individual work processes).
- The *existing interfaces* and other *inter-operability features* can be considered as *relations with other activities*. This kind of situation may prevail, for instance, if we have a system for managing the personal data of our organization that we would like to utilize in our new software system. These kinds of relationships and dependencies are usually documented in architecture specifications, but as well they can be included in the work activity model (i.e. the ActAD analytical framework).

- From the *database description* a number of useful concepts of the target domain can be derived. Additionally, the database structure indicates how these concepts relate to each other. The database description can be used as a basis of the *concept model*, into which all the terms of target domain will be captured during the requirement specification phase. Many of these terms have been able to be assigned in different *elements of the work activity* and vice versa (see figure 3). As the specification workflow progresses, the concepts depicted in the work activity model are included in the concept model. Notice that, at this phase the concept model does not have to be scoped to the intended system or even be detailed, since these things will be arranged later.

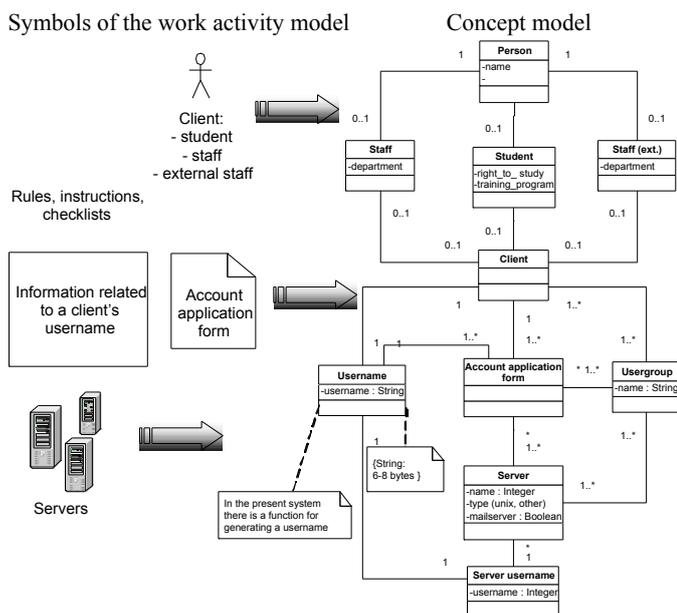


Figure 3. The figure illustrates how the symbols used in the work activity model relate to the concepts (classes) in the concept model.

It is obvious that the work activity model, as well as the concept model, will be completed as the process goes on. Anyway, on the grounds of the examination of the existing assets and on the basis of the thematic interviews we are able to form a rather extensive picture of the activity with the help of the framework (figure 2). Besides providing a holistic picture of the activity, the ActAD framework provides a checklist of key issues that need to be clarified (Korpela & al., 2004).

In addition to information structuring, at this phase we perform an expert walkthrough to a user interface of the existing software system. For this purpose we can utilize the existing heuristics (see e.g. Nielsen&Mack, 1994 and The Usability Method Toolbox) or the screens can be evaluated in general by the analyst-team. The aim of the walkthrough is to document the problems and the advantages of the present software from a usage viewpoint. The usability analysis document, which is created here, serves most the user interface design.

As a result of the phase the analyst-team have an overall picture on the activity and the idea of the issues that is needed to examine in more detailed. The results of the phase are documented as follows:

- Work activity model that describes the present state of the activity. Also the model of the intended state is been able to outline on the grounds of the preliminary information gathered at both the preliminary research and this phase. Both models will be completed during the requirement specification.
- Preliminary concept model.
- Usability analysis
- (Software requirement descriptions that will be analyzed later.)

Next we will proceed to study work processes more carefully.

## 2. Focusing on the target domain

It is obvious that if we want to support the work processes, we have to understand them. At the second phase we examine more profoundly the main work processes that are selected on the basis of the holistic activity examination. The processes to be studied more carefully are chosen, for instance, on the grounds that

- there are problems with incorrect or missing data in the present software system or work activity. This indicates to the problems of communications between the actors (or technological information systems).
- there has been introduced new work practices, technologies, tools etc. within the activity or activity network. These changes may have caused new (or changed) requirements for the intended software. Usually all effects can be identified only by understanding the required processes.
- the analysts are unfamiliar with the processes; what phases are included in process, who is responsible for which actions, how the process progresses, what tools and other means are used for works and communications, etc.

On the basis of the examinations of work processes the real software needs and requirements are identified. Another remarkable result is that by analyzing the processes they could be streamlined in many ways. For instance, by analyzing the processes we can indicate whether there are some actions to be automated (e.g. transfer some actions to systems' responsibility). We also will notice whether there are some unnecessary steps within the processes that could be removed. Further, we can point out some non-technological development needs within the activity or activity network. This refers to the problems that could be eliminated by developing the activity, e.g. by changing division of labor or by making instructions for work.

### Zoom-in on the work processes from the multi-actor's point of view

If there are any unclear issues on the current (or new) processes or communications, it is important to clear them up. By the multi-actor's viewpoint we mean that there are several than one actor (i.e. person), who participate in the work process. For instance, the processes within a certain activity, as well as the processes within the activity network, represent the multi-actor's viewpoint. The thematic interviews of different actors and walkthroughs with domain experts have proved to be suitable for clearing up the processes.

The processes are documented by means of the UML activity diagram notation (Fowler, 2000; Cheesman & Daniels, 2001), because it allows us to indicate also the responsibilities of the actors (i.e. who is responsible for which action). Additionally, we use the symbols of the work activity model to represent the tools and other means of working and communications. By doing so, there remains the traceability between the work activity model and the process diagrams (see figure 4)(Korpela & al., 2004). An essential advantage of the use of this notation is that users and domain experts will not have difficulties to understand it. Or if so, the textual descriptions can also be written to support the diagrams. Another important issue is that the diagrams are more expressive than traditional UML activity diagrams.

### Zoom-in on the individual work processes

At this level we pay our attention on user's workflows and the working practices. It is important not only concentrate on the interaction between the user and the current system, but also examine the user's action as a whole. This approach we have adopted from GUIDE-process model (Laakso, 2003), which has been applied to the user interface design in our case study.

By now, we may have found out the most interesting workflows that are the frequently occurring ones, and, the flows with broad coverage of the functionality of the current system. For figuring out, how a user really acts in order to complete an action, we use the observation of the real work. We select, for instance from two to five actions that we clarify in details. We will find out the work phases included in the workflow, the means of work needed and used, the interaction of present software and user etc. There are different methods for that how the observation is carried out. For instance the master-novice approach means that the real user of the current system tutors one of the development team member how to perform the selected actions. During the observation situation the other analysts stay in the background and observe how the working practices progress. Regardless of that how the observation is realized it is important that the observation situation is planned carefully in advance. Otherwise the results may remain inadequate.

The individual workflows are documented by using the same activity diagram notation, which is presented in the previous paragraph. The process diagrams may be completed with textual scenarios, if necessary. A

scenario means an informal written story about that how user acts to perform an action. All aspects related to the usability of the current system will be added in the document 'user interface usability analysis'.

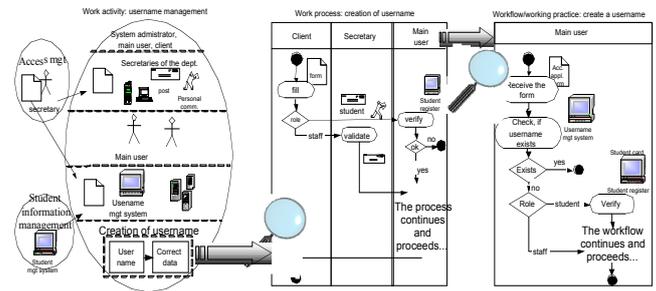


Figure 4. An example about how traceability is preserved through the different level of process diagrams. The diagrams from the left: holistic work activity, multi-actor's process diagram and user's workflow diagram.

We analyze all process diagrams, descriptions and development ideas carefully. During the field studies there usually appear lots of problems or development ideas that have been written down. By analyzing work processes with the help of symbolized activity diagrams, we notice quite easily the essential development needs and software requirements. The analysis task may be performed through a few iterations, since we start from general level and proceed towards more detailed issues. During the analysis process the results of each iteration are consulted with the expert's of the target domain. When we are satisfied with the results, we create new diagrams, where all the development ideas are considered. If the diagrams need textual support – and usually they do – we write them in *concept scenarios*. A *concept scenario* describes how user performs an action when a new or improved software system is introduced in the action. Notable here is that the concept scenario describes the desired state of an action where as the scenario explains the current state.

The analysis of work processes aims at defining exactly who are the users and which functions are the responsibilities of the intended software. On the basis of the high-level use cases (i.e. process diagrams + the concept scenarios) all users and their goals for using the software are documented into a user analysis. In the high-level use cases the interaction of the users and the software is described, but not yet in details. This will be done during the user interface design phase.

### Concept analysis

During the activity analysis, we have captured all concepts of the domain and the information related to them in a concept model. The diagram shows us not only the concepts but also how these concepts are related to each other, what kind of data the concepts include and all known rules and constraints related to the information. Although concept modeling is related to the information analysis, it cannot be separated from

the activity analysis. Actually, only by understanding the concepts of the domain and their relations with each other we are able to understand the activity.

Here we will concentrate on concepts that are relevant from the viewpoint of our system. This means the concepts and the information that must be managed by the system. During the concept analysis the concept model is transmuted into a type model. By a *concept model* we mean a software-independent conceptual model, which contains the information of the domain. The purpose of the concept model is to capture domain knowledge. According to Cheesman and Daniels the concept model is represented as a form of UML class diagram (Cheesman&Daniels, 2001; Fowler, 2000), which illustrates concepts as classes and relationship between the concepts as associations (with multiplicities). In addition, significant data related to concepts is depicted as attributes with or without types and other properties. Operations would not be used instead.

A *type model* is derived from the concept model so that it depicts precisely the domain information that is relevant to the scope of the intended software system. Thereby, although both models are depicted as a UML class diagram, their purposes are different. For example, in the concept model the class 'user' represents the concept of user in the domain. Instead, the class 'user' in the type model symbolizes the user from the point of view of the system, which typically is more precisely defined.

We create a type model by coping the concept model. After that we go through the concepts one by one and decide whether or not there is some significant information from the point of view of the intended system. Due to this procedure we add or remove elements until its scope is correct which means we have defined together with the domain experts what data is stored and handled in the intended software system. If it is difficult to scope the type model on the basis of concept model, we suggest that the concept analysis will be done simultaneously with the user interface design. This will happen so that, while designing the user interface screens on the grounds of the high-level use case descriptions, the work phases will be gone through one by one by thinking what kind of information is needed during the execution.

When we have a scoped type model, we add all required domain rules by writing some constraints (in natural or in some definition language), by introducing new attributes and by specifying the association role multiplicities (Cheesman & Daniels, 2001). The domain rules with constraints are largely captured at the earlier stages and depict, for instance, in the communication and co-operation category in the work activity model. The rules are found and identified from the existing system, all documentation related to the system and activity itself (e.g. technical documentation, forms and instructions) and through the field studies. Nevertheless, the additional rules and constraints may still come to existence. Especially during the user interface design, which usually at least partly overlap with the concept

analysis and the zoom-in on the individual work processes.

During the concept analysis we will also face the architectural issues, such as what kind of interoperability with other systems are needed and what other dependencies does the system have. These features can be expressed in a work activity model. In more details these features may be described in the user analysis. The figure 6 and 7 show how software's type model can be derived from the activity's concept model.

With the help of the type model we can check that all use case candidates are identified. We pore over the model with create/destroy check and association update check, i.e. with a list of the questions like (Cheesman & Daniels, 2001):

- Do the concepts get created or destroyed? If so, how does it affect the system? Does the concept have any attributes that might change?
- Do the relationships between the concepts change over time? If so, how they affect the system?

As a result of the whole phase two the analyst-team have more profound understanding on the work processes on the grounds of which the boundary of the intended software is specified. Here, at the latest, we write a high-level system envisioning statement (one concept of A4 or so) on the basis of all information we have. The outcomes are documented as follows:

- Short (textual) description of the intended software system: its purpose, functions, data, users and interfaces to other system.
- User analysis.
- Type model.
- (User interface) usability analysis.
- Process diagrams with textual descriptions on users' work processes.
- Process diagrams with textual descriptions on multi-actors' work processes.
- Updated work activity models.

### 3. User interface design

The purpose of user interface design is to concretize the intended software to the parties in the project: client, users, system analysts, designers, implementers, testers and so on. During this stage the software is defined and tested with the client. Notable here is that we do not write any software code at this stage. The method for user interface design is described in (Laakso, 2003). As the result of this phase we have a user interface specification where the functions of the intended software are described by means of the high-level goal-based use cases and "screenshots" of interaction.

### 4. Software requirements description

The purpose of this stage is to document the software requirements in appropriate form from the viewpoints of the following phases. During this phase we polish the main outcomes of the requirements specification phase. The requirements of the intended system are described

- in a user interface specification,

- in use case descriptions,
- in a type model.

By a *use case* we mean the description of intended system behavior from the user's point of view. In other words, a use case describes step by step how the user is intending to use a system to accomplish a single action or event. Use cases are documented by using textual stories in accordance of UML notation (Fowler, 2000; Cheesman & Daniels, 2001). Quality properties (e.g. performance and security aspects) are added to each use case statement. However, if the expectations consider the whole system these requirements are described separately.

Other useful outcomes of the requirement specification phase are as follows:

- short textual description on the intended software,
- user analysis,
- process diagrams with textual descriptions on users' working practices,
- process diagrams with textual descriptions on multi-actors' work processes,
- work activity model that describes the intended state of the activity.

### 5. Deriving the software architecture

At this phase we finally form the initial application architecture on the basis of the specification artifacts. The steps are faithful to the ones in Cheesman and Daniels procedure. That is why we also have adopted the concepts from them.

The application architecture is divided into four layers (see example in figure 11):

- *User interface*, i.e. the presentation layer.
- *User dialog*, which handles dialog logic.
- *System layer*, where components correspond to business system.
- *Business layer*, where components correspond to stable business concepts and are associated database.

We start by identifying the *business interfaces* and *components*. Then we continue with *system level interfaces* and *components*. When we have an initial set of *component specifications* for the both type of interfaces, we put them together and so we have the *initial application architecture*.

Business interfaces are identified on the basis of the type model, which we get at the requirements description stage as a result of the concept analysis. The type model contains a lot of information and business rules. Business interfaces are discovered by identifying core types in the type model and creating interfaces to manage them and their details.

A *core type* is a business type that has independent existence within the business (or core activity). All the other types provide details of the core types. The purpose of identifying core types is that we start to think about which information is dependent on which other information, and which information can stand

alone. The step is useful for allocating information responsibilities.

First of all, we create *business interfaces* that manage the information represented by the core types and their detailing types. The rule is that one business interface is created for each core type. Now we have some business interfaces in our type model so that we can start to assign *responsibilities*. The purpose of this step is to clarify which information will be managed by which interfaces. Evidently, each core type is allocated to its own interface. (In type model, allocation of types to interfaces is indicated by a solid diamond symbol, see example in figure 9). Then, if a detailing type only provides the detail for one other type, it is allocated to the same interface as that type. If the detailing type details more than one type but they are allocated to the same interface, the detailing type belongs to that interface too. But when a type details other types that are allocated to different interfaces, the decision is made on the basis of how tightly it is coupled with other types within the potential interfaces. This decision means that we initially indicate, which interfaces will be responsible of managing the *inter-interfaces references*, and where referential integrity will be maintained. (In type model inter-interface responsibilities are depicted with a thin arrow, see example in figure 9).

When we have allocated the responsibilities, we create *components specifications* for each business interfaces. The rule is that for each business interface is created a component specification. So we have identified the business interfaces and their component specifications. Now it is time to start thinking about *system interfaces*. The principle is that each use case represents one system interface and their operations are derived from use case steps. According to this technique each use case will be pored over step by step, and for each step we consider whether or not there are system responsibilities that are represent as one or more operations of the appropriate system interface(s). We have an illustrative example in figure 10.

Another way of identifying the operations is to derive them from screen designs, which are the artifacts of the user interface design workflow (see example in figure 12). Operation discovering on the basis of these images may be easier and faster than identifying on basis of the use case. Anyhow, the both techniques have their advantages and disadvantages.

The operation identification on the basis of the use case description is more laborious, because during the user interface design all use cases have already been pored over, and overlaps among the operations have been dealt with. Instead, here we have to do it (again). However, these two workflows may overlap with each other so that it may be worthwhile to begin to identify the operation on the basis of the use case descriptions rather than wait the completion of the user interface specification. The user interface technique may also lead too specify components from the point of view of reuse. On the other hand, the selected approach support the architecture model, which aim at improving the

interchangeability and reuse potential of each component.

Ultimately, it does not matter by which technique the operations are founded. More important is to place the created system interfaces into the suitable *system components* and assign the dependencies between the components.

The use-case-derived system interfaces are usually strongly overlapping and manage the concepts that have the same lifetime. That is a good reason for putting all of them into the same component specification. But, on the other hand, we must also consider the implications for component deployment and replacement. We want to be able to build and upgrade their implementations separately. So, at this phase we only make the outlining decisions. [Later, during the component specification workflow (not considered in this paper) we examine the components interaction in more details, as the result of which we have factored interfaces and component specifications.] In addition to create the component specification(s) we add all understood dependencies between the components. Also these dependencies will be validated during the interaction studying.

Now we are almost finished, only the existing interfaces, systems and other potential dependencies must be considered. If we have described them in the work activity model or high-level system envisioning, we are able to get them there. Otherwise we may utilize potential architecture models or other existing description.

At the final step we combine all these elements (i.e. component specifications) into the architecture. The architecture model is illustrated in figure 11.

## GENERAL CONSIDERATION ON THE METHOD

The activity analysis, particularly at the first time, may be time-consuming but rewarding. The software engineering process is iterative so we can re-use produced documentation later on next phases and in the following projects. Process models are also useful in maintenance of the software and in orientation of new employees on the target domain. In addition, they provide information and checklists for everyone in software development organization. One can always examine them when needed.

## PAKKANEN – AN ILLUSTRATIVE EXAMPLE OF APPLYING (AND DEVELOPING) THE METHOD

The purpose of our case study was to pilot component- and service-based techniques and suitable methods for component oriented software development. We did this by re-engineering a small software system, which had earlier been developed in our department for managing the university's usernames.

### 1. Defining the whole activity domain

We started our study by capturing all conceivable concepts of the domain (i.e. username management) into the work activity model. Because our team was not familiar with the technique, we did it with the help of

the activity analysis expert on the basis of our own understanding of the domain. As a result we had an initial work activity description that we started to fill-in by examining the existing system and its documents and other available assets, such as the account application form with the instructions.

After that we had the idea of the themes that we were interested in. We utilized the thematic interview model, in which the themes are structured according to the ActAD framework. Our themes may be expressed through these questions:

- Who are the clients of username management? Who need and for what purpose the username management.
- What are the object and outcome of username management? What is the purpose of the management system?
- What are the key actions within the activity? Which other work processes affect the activity? What are the object and outcome of each action?
- Besides the management system, what other tools are used and needed in the actions (e.g. the other systems, the forms, the checklists)?
- Who are the actors?
- What kind of rules and division of labor there are? What kind of instruments is used in communication? What co-operation modes there are?
- What is the collective actor like?
- Which are the previous and the following activities?
- What is the relevant data related to the system, and where does it come from?

In our case the selection of the interviewees was easy since the existing system only had a few users. The system's administrator and main user were interviewed, and the results were depicted within analytical framework.

At this phase we also started to collect the requirements into separate documents (i.e. the desired state descriptions). The requirements for technical usage environment, for the intended functionality and for the intended users were shortly delineated in a written description. The knowledge of the domain was started to capture in a concept model.

### 2. Focusing on the target domain

#### *Work processes*

Mainly on the basis of the analysis of the thematic interviews we decided to clear up and model four work processes from the multi-actor's viewpoint:

- Person leaves the university.
- Person changes a department.
- Person changes name.
- Creation usernames for new students (several persons).

First three processes were selected, because due to the interviewees in the existing system there were problems with the data that was not correct. In addition, there was

introduced an entirely new process for creating usernames for incoming students. We wanted to figure out, if there were new responsibilities to the envisaged system.

For information gathering we used the thematic interview. This time, besides the users of the existing system, we also interviewed two secretaries from one department. The results were described as process diagrams. Additionally, for process analysis, the main user gave us one process diagram from a department's quality management system.

From here we moved to study the user's workflows by observing the real work. We ended up two interesting actions:

- Create a username for a student.
- Change an existing username due to changing of a name.

The first one was selected because it was the most frequent action. The other was chosen on the basis that it covered the functions from deleting (or closing) the old username to creating a new one. The current workflows of the main user were depicted using the same activity diagram notation as in multi-actor's viewpoint descriptions. In addition, we wrote up the workflows into the scenarios.

The analysis of the present workflows was mainly performed in brainstorming session, which was held by the analyst-team. The main noteworthy result of the analysis was that we were able to streamline the main workflows. Especially the change of the username, which currently was an extremely complicated action, was simplified remarkably in the intended state description. Our example in figure 5 illustrates how the user's workflow "create a username" has been streamlined on the basis of the observation. Three phases have been removed from the workflow (1-3). In addition, few means of work have disappeared because of the development of the interoperability of the existing systems (4-5). The third development aspect is that one before manually made task has been changed to perform with the intended software (6).

Afterwards the results were described as process diagrams and concept scenarios that represent high-level use cases (e.g. figure 5: create a username as a user's intended work process). In textual description we offered more detailed information for instance on intended login mechanisms, checking and searching techniques and so on. These descriptions served as main inputs for the user interface design.

In our case, however, the descriptions were too detailed, because we had also specified user and software interaction precisely, which actually is the purpose of the user interface design. The user interface designer had difficulties to detach herself from the given models.

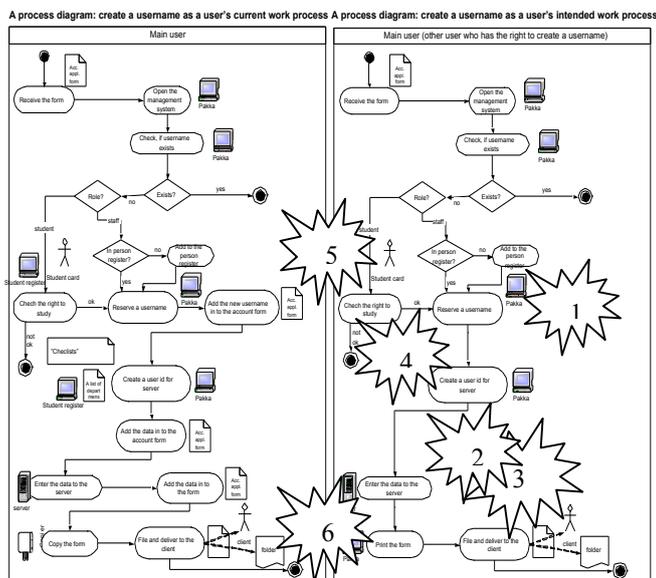


Figure 5. The diagram on the left describes the present state of the user's workflow "create a username". On the right hand the same workflow is described as the intended workflow.

However, the detailed use case description proved to be useful as the basis for the UML use cases, since the software's functions were able to be noticed on the process diagrams. So we concluded that all relevant information is needed in the high-level use case descriptions that are the inputs to the user interface design.

### Concept analysis

In figures 6 and 7 is illustrated how the type model was derived from the concept model(s) in our case study. The concepts of the initial concept model were gone through one by one by deciding whether there was significant information related to the system. Because we now had the envisaged system viewpoint, some concepts of the concept model were removed (see figure 6) and all known rules and constraints were added as attributes, as multiplicities and as written constraints, if they already did not exist there. In addition to information specification, we also examined the interoperability requirements with other systems. Finally, we ended up a scoped type model, which represented all the decisions that were made during the analysis (figure 7). For instance, we decided that the new system utilizes the organization's existing person register (i.e. personal data is not stored in the username management system but only a reference to person register is stored). Secondly the concept 'client' is named as 'user' and the concept 'username' is introduced as an attribute of the user. In addition, servers and user groups information is maintained within the intended system. Other notable aspects are the normalization of the relationship of the concepts, and other rules and constraints described in textual mode.

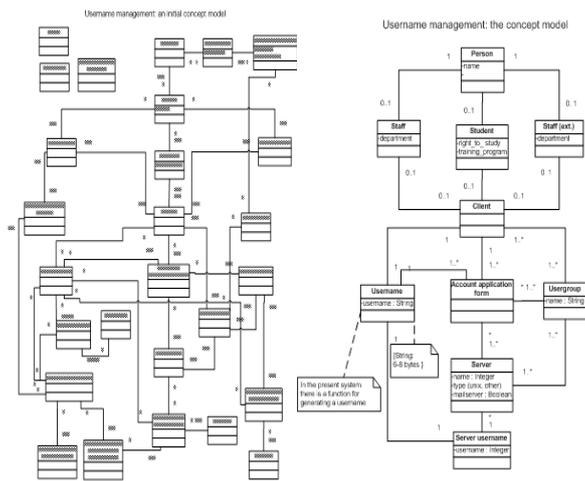


Figure 6. The concept model on the left includes all the concepts of the target domain (username management). The concept model on the right hand includes only the concepts that are relevant from the intended software's viewpoint.

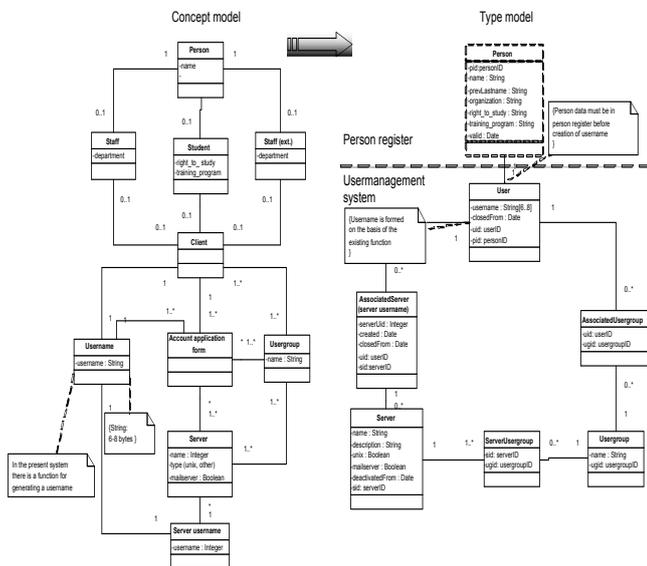


Figure 7. Here is illustrated how the concept model is transmuted into the type model.

### 3. User interface design

The user interface designer made an interaction design for every important use case. After that she draw the screenshots by means of which the functionality of the software was illustrated and tested with the main user. The screens had been designed without any programming. In figure 12 there is one screen example of screenshots of interaction.

### 4. Software requirements description

At this phase we updated our work activity model according to the intended work activity (figure 8) and we wrote (exactly completed) a high-level textual description of the envisaged system.

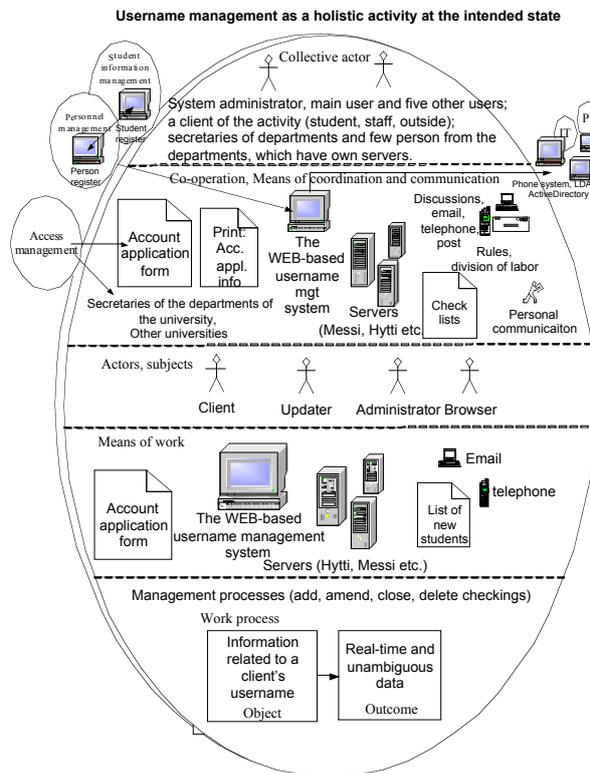


Figure 8. The envisaged activity - username management – captured within the ActAD-framework.

In our case the information of the domain is structured as follows:

**Activity:** username management (of the university).

**Collective actor:** people who handles or needs the correct data related to username.

**Co-operation, means of coordination and communication:** interest groups, means and tools for ensuring the correctness of the data, e.g. division of work, rules, communications, forms. Also the system's connections to other systems are depicted here.

**Actors:** users of the intended software.

**Means of work:** all means and tools for managing the data, e.g. forms, software systems.

**Work processes:** are identified on the basis of data handling processes (create, amend, delete).

**Object:** client's username (and data related to it).

**Outcome(s):** the correct data due to which the client is able to use e.g. email service, or these services are removed or restricted. Outcomes can be specified from different viewpoints, e.g. client, user or other systems.

**Network activities:** previous activities that set the preconditions for the activity; following activities that utilizes the outcomes and all other activities that affects on the target activity. For instance how that affects on our software that person changes organization within the university or person finishes working at the university.

All use case candidates were founded by the existing system and by analyzing the process descriptions. They were checked with create/destroy analysis of the type model (explained in paragraph "concept analysis"). The

UML use case scenarios were written on the basis of the process analysis. In addition, later the artifacts of the user interface design, i.e. the screen designs, proved to be very helpful, because by them the functionality of the system became evident more concretely. Here we show a stripped-down example of one our UML use case.

**Name:** Create a username

**Initiator:** Username creator

**Goal:** All the data is added into the system and the client is informed of the username and the first-time password

**Main success scenario:**

1. Select the client from the system.
2. Check that the client does not have an existing username.
3. Generate a username and save it.
4. Add other details related to user (e.g. server and user group(s)), and save.
6. Generate the first-time password.
7. Print data for the client.

**Extensions:**

1. The client's person data does exist in the person register. *Here we describe what should be done...*
- 2 a) Client already has a username, which state is close → go to use case “Open the closed username”.
- 2 b) Client already has a username...
3. And so on...

We also wrote some additional requirements and expected features, such as search key candidates, in these descriptions.

At the end of the phase we had a requirements specification described in a type model (information) and use case scenarios (functionality with all relevant features).

### 5. Deriving the software architecture

In the requirements specification workflow we created a type model, which included all relevant concepts. Here we created on the basis of the type model an interface responsibility diagram according to the Cheesman and Daniels procedure (example in figure 9).

As core types we identified “user”, “server” and “user group”, because we had decided to maintain in our system all data related to these concepts. Additionally, these concepts were determined to have an independent existence. After identified the core types, we created a manager interface for all of them, e.g. IUserManagement for the core type “user”.

All other concepts, such as “associated servers” and “associated user groups”, provided details of the core concepts. So, next we had to allocate for each such type the manager interface that managed the data related to it. This responsibilities assignment is shown by a solid diamond symbol (see figure 9). After that there was one thing left: we had to allocate the responsibility for inter-interface associations. In other words, for associations

which exist between types managed by different interfaces. For example, see association between “server” and “associated server” in figure 9. In this case the decision has been made so that the interface IUserManagement will store the reference between these two types. This is indicated with a thin arrow.

As a result of this work we had identified the business interfaces. For each interface we created one component specification so that we had first components for the business layer in our software architecture.

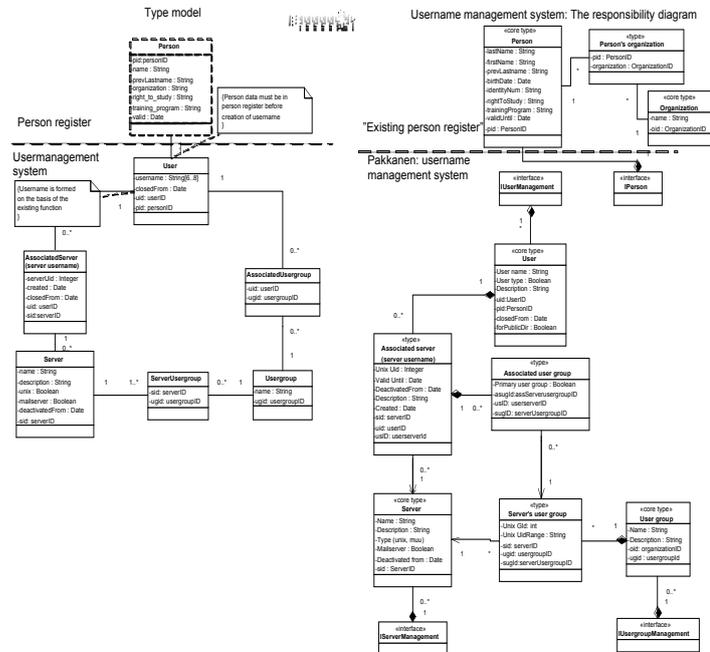


Figure 9. The responsibility diagram of the business interfaces (on the right) is derived from the type model.

From here we moved to identify the system interfaces and operations. We decided that each use case scenario represented one system interface. So, for instance, on the basis of the use case “Create a username” we created an interface called ICreateUser. By each step we considered is there some functionality for which the system would be responsible. By acting so, we recognized an initial set of operations of the ICreateUser interface. This is illustrated in figure 10.

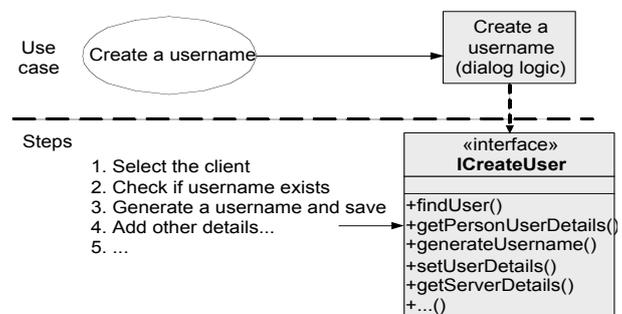


Figure 10. System interface is derived direct from the use case. The operations of the interface are discovered poring over the steps of the use case.

When we were satisfied with the initial interface specification (i.e. operations), we created a component specification for the interface. After that we put all these pieces together in accordance with the adopted architecture model (figure 11).

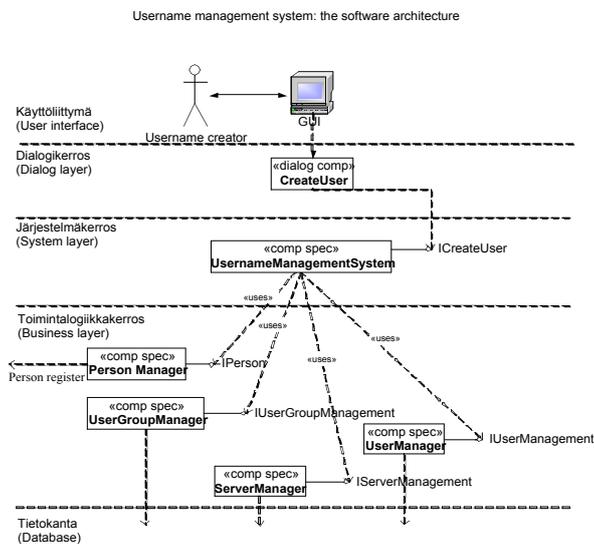


Figure 11. The initial software architecture. There are four components at the business layer, but only one at the system layer. In this method, the dialog components are not defined.

Later, we also provided the user interface technique for the operation finding. In accordance with the technique, the operations of a system are discovered from the screen designs (example in figure 12). We noticed that the technique cut corners of the system interface identifying process. For example, here we can see what data the search operation returns. In addition all individual functions (e.g. buttons) represents at least one operation. By means of the screens we can not only recognize the operation candidates but also specify their execution order and both input and output parameters.

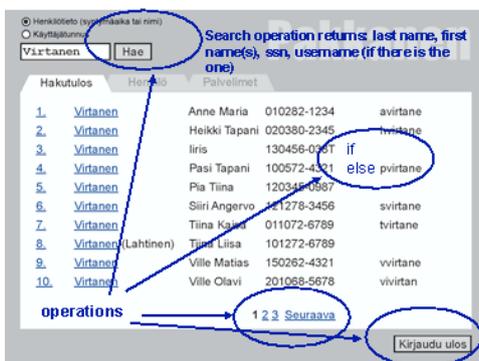


Figure 12. The user interface screen consists a lot of information for interface operations design.

In our study the application of this method ended up at this level. From here, we continued to specify the architecture and the components in depth details according to Cheesman and Daniels procedure.

## ACKNOWLEDGMENTS

Pauliina Ikävalko, Kirsi Karvinen, Kirsi Reponen, Saara Savolainen and Ritva Silvennoinen have greatly contributed to this paper. We thank our case study clients, Olavi Manninen and Sirkka Eskelinen, and the whole project team.

The paper is based on research funded by the National Technology Agency Tekes through the PlugIT project, grants no. 40246/02 and 90/03 in 2001-2004.

## REFERENCES

Cheesman, J., and Daniels, J. *UML Components. A Simple Process for Specifying Component-Based Software*. Addison-Wesley, 2001.

Fowler, M. *UML Distilled, A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, USA, 2000.

Herzum P and Sims O. *Business Component Factory*. Wiley Computer Publishing, New York, 2000.

Korpela M, Mursu A, Soriyan A, Eerola A, Häkkinen H & Toivanen M. Information systems research and development by Activity Analysis and Development: Dead horse or the next wave? In: *Kaplan B, Truex DP, Wastell D, Wood-Harper AT & DeGross JI, eds. Information Systems Research: Relevant Theory and Informed Practice*. Boston: Kluwer Academic Publishers, 2004, pp. 453-470.

Laakso, S.A. 2003. *Ensuring quality of the user interface with the GUIDE project model*. Available at <http://www.cs.helsinki.fi/u/salaakso/papers/GUIDE.pdf>

Nielsen J., Mack R.L. *Usability Inspection Methods*, John Wiley & Sons, Inc., 1994.

PlugIT project 2001-2004. Available at <http://www.uku.fi/tike/plugit/english/>.

Toivanen, M., Häkkinen, H., Eerola, A., Korpela, M., and Mursu, A. *Gathering, Structuring and Describing Information Needs in Home Care: A Method for Requirements Exploration in a "Gray Area"*, in *MEDINFO 2004: Building High Performance Health Care Organizations*, San Francisco, 7-11 September 2004, p. (forthcoming). IMIA, 2004.

The Usability Method Toolbox. Available at <http://jthom.best.vwh.net/usability/>.