

Design and Professional Languages

Peter Bøgh Andersen
Kim Halskov Madsen

DAIMI PB – 244
March 1988

Design and Professional Languages*

Kim Halskov Madsen and Peter Bøgh Andersen
Information Science Department
Aarhus University
Denmark

(email: halskov@daimi.dk)

Abstract

This paper is about the design of computer systems and the professional language of its users. In the first part of the paper, we present an investigation of the professional language of three employees at a library. The focus of our attention is the intensive use of metaphors. The investigation supports the idea of using metaphors in design since use of metaphors is one way of relating to the previous experiences of the employees. In the second part of the paper, we present a linguistic approach to design of a computer system. Our approach is related to ideas of K. Nygaard. He has suggested that a careful examination of the professional language of the users should be followed by enriching the professional language through concepts related to informatics. We suggest an approach, not of planned language change, but an approach where the professional language is taken as the starting point for design. We present a concrete way of making a linguistic analysis of the professional language and we show how the linguistic choices may be transposed into a number of design choices. As an illustration of our approach, we use the material from the first part of the paper that documents the physical space metaphor.

* To appear in S.F. Larsen & K. Plunkett (eds.): *Computers, Mind and Metaphors*, the proceeding of The Symposium on Computers, Cognition and Epistemology, Institute of Psychology, University of Aarhus, 1987.

Contents

1. Introduction	1
2. An investigation of a professional language	2
2.1 Recording	2
2.2 Metaphors	2
2.3 The physical space metaphor	3
2.4 The conversation partner metaphor	6
2.5 The organism metaphor	7
2.6 The media metaphor	7
2.7 Discussion of the observations	8
3. The design approach	9
3.1 Corpus	11
3.2 Analysis of corpus	13
3.3 The command language	18
3.4 Discussion of the command language	21
3.5 Summary of the design approach	35

1. Introduction

This paper is about the design of computer systems and the professional language of its users¹. Computer systems are controlled by signs, be they pictorial or verbal. This means that a system designer makes strong suggestions to the users concerning how the system should be referred to and interpreted. Since we use language to create dispositions for action, it follows that attention to the semiotic aspects of design is important. The paper is in two parts:

In the first part of the paper, we present an investigation of the professional language of three employees in a library. The focus of our attention is the intensive use of metaphors - for instance the computer system seen as a physical space. Metaphors are used by the employees to understand substantial parts of the structure as well as the functionality of the system. The metaphors do not originate from the specific profession but from the humans everyday experience. The investigation supports the idea of using metaphors in design since use of metaphors is one way of relating to the previous experiences of the employees.

In the second part of the paper, we present a design approach where the professional language is taken as the starting point for the design of a computer system. Our approach is related to ideas of K. Nygaard². He has suggested that a careful examination of the professional language of the users should be followed by enriching the professional language through concepts related to informatics. We suggest an approach, not of planned language change, but an approach where the professional language is taken as the starting point for design. On the one hand, we want to use the professional language as our point of departure, thereby making the computer system more powerful and easier to use. On the other hand, since the use of the command language will differ from the use of the professional language, we consciously introduce changes in the language. We present a concrete way of making a linguistic analysis of the professional language and we show how the linguistic choices may be transposed into a number of design choices. As an illustration of our approach, we use the material from the first part of the paper that documents the physical space metaphor.

The second part of the paper (section 3) may be read independently of the first .

¹The paper is inspired by the work of working group 2 of the SYDPOL programme. SYDPOL is a Scandinavian research programme financed by NORDFORSK. The aim of the programme is to support the development of methods, languages and tools for user participation in systems development. We wish to thank our colleagues in the working group for many stimulating discussion.

²K. Nygaard: Profession Oriented Languages, Presented as keynote speech at Medical Informatics Congress Europe 1984, Brussels September, 1984.

2. An investigation of a professional language

In this section we start out by presenting the method of the investigation and the metaphor concept. The main part of this section documents four metaphors of the professional language.

2.1 Recording

The paper is based on tape recordings of three conversations we had for approximately one to two hours with employees at a Danish library; two librarians, and one clerical worker. The three employees were working in the circulation department, at the circulation desk, and the catalogue department. During each conversation we and one of the employees were sitting in front of a terminal. The employee was asked to show us some typical tasks involving use of the present computer system, and as she went along, to tell us what she did. To stimulate the conversation, we asked questions like "what does this mean?" or "what was it called before computerization?"

The tape recordings were examined in the following way. We heard the tapes and took notes in terms of keywords from phrases or sentences we found relevant. In this way we got an overview of the language use. Later on we heard the tape once more and wrote down the important sentences word-for-word.

Our original aim was to record the language use related to the use of the computer system in general. But soon, we realized that metaphors played an important role when the employees talk about their work, which made us focus on this specific aspect of language use.

2.2 Metaphors

Using metaphors means understanding one thing in terms of another, but in a way that is seemingly wrong. For instance, you may talk about an electronic brain and hereby see or understand the computer as a human brain, which is seemingly wrong because a computer is not made up by living cells. Within linguistics a metaphor may be defined as:

A metaphor is an utterance where a concept X is assigned to the main category B of another concept Y. B must be a major linguistic category like *animate*, *inanimate*, *human*, *animal*, *plant* *physical object* etc. In its literal sense, X must belong to another category A that is disjunctive from B³.

For instance, the utterance "Bruce is a lion" is metaphorical because Bruce (a human) is assigned to the main category of lion (an animal).

The difference between a metaphor and a comparison (Bruce looks like a lion) is that a metaphor provokes a minor break down of the language whereby an

³P. Bøgh Andersen: Semiotics and Informatics: Computer as a media. In Ingwersen et al.: Information Technology and Information use, Taylor Graham, London 1986 p. 64.

important distinction (for instance living - dead) is eliminated. But a comparison remains within the structure of the language.

What you consider as a metaphor depends on what is meant by "major linguistic categories". In classical rhetoric, the category changes are concerned with the main structure of the language. In personifications, for instance, attributes normally assigned to human beings are assigned to abstract concepts, e.g. "Thou blind fool, love, what dost thou to mine eyes"⁴. And in animation, attributes normally assigned to living organisms are assigned to dead things, e.g. "Now, ere the sun advance his burning eye. The day to cheer and night's dark dew to dry"⁵.

Metaphors are not only used within poetry. They pervade our entire language⁶. Computer systems are often understood in a way similar to the way that Shakespeare gives a tangible presentation of the abstract concept of love and its effect on your eyes. An employee at the library says, for instance: "I am in the circulation control system". The intangible computer system is understood in terms of concepts normally used about a physical space.

Computer systems are often understood metaphorically because we lack any direct sensory experience of them. We can't see what is going on behind the screen. Hence we try to see them as something we experience in our daily life. Moreover, computer systems are a fairly new phenomena, therefore language has to change to make it possible to talk about them. The history of language shows that metaphors play an important role in language change⁷.

We proceed and present observations from the tape recordings. In each of the following four subsections we document the use of one of the four different metaphors: The physical space metaphor, The conversation partner metaphor, The organism metaphor, and The media metaphor.

2.3 The physical space metaphor

The identification of the physical space metaphor is based on the following criterion: When logical or functional aspects of the computer system are referred to in terms of concepts normally used about a physical space, then the physical space metaphor is used. As an example we have *I can go in and make*

⁴W. Shakespeare: Sonnet 137.

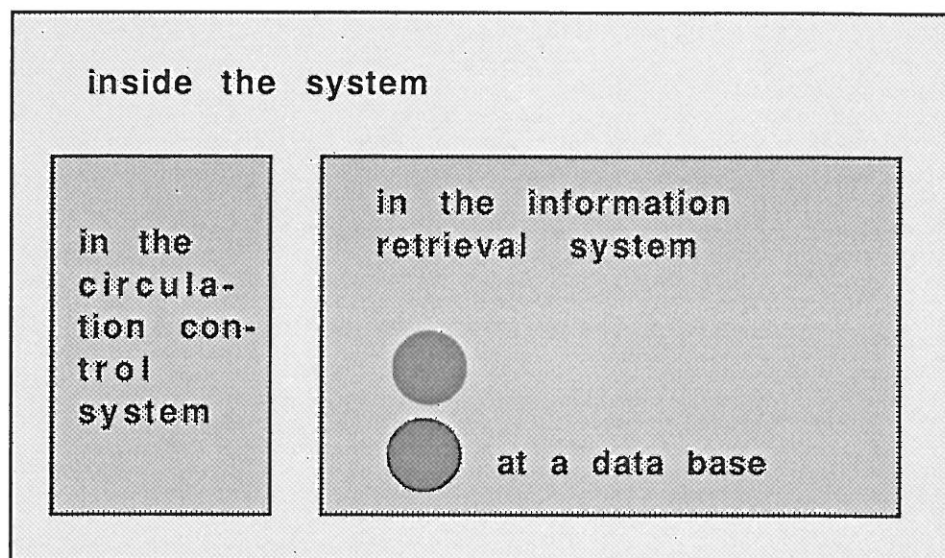
⁵W. Shakespeare: Romeo and Juliet, Act II, Scene III.

⁶G. Lakoff and M. Johnson: Metaphors We Live By, The University of Chicago Press, Chicago 1980.

⁷A Kock: Sprogets forandring, in Danish (The change of language) , H. Hagerups Forlag 1913.

reservations. The employee does not literally move her body to another physical location, but she brings the computer into a state where she can make reservations. The physical space is like a building in which you can move around. We may draw the following map,

outside the system



You can go in,

1. they must have made another *access image*⁸
2. *the access codes* to the system are made up of two parts
3. then I could *go in* and make back-up copies, and place some magnetic tapes on the system.

When you have gone in you may call forth the system you want to use,

4. And I can go in and *call forth* the circulation control system

You have two alternatives, the circulation control system and the information retrieval system. In the information retrieval system you can move between several databases,

⁸ Examples number 1 to 20 are *word-to-word translations* of the corresponding Danish sentences, because these form the corpus of the linguistic analysis based on the Danish language use, (section 5). The rest of the translations are idiomatic, i.e. translations only preserving the meaning.

5. then I have *to go back* to the ALIS-base

and you can move objects from one base to another,

6. that is we *move* a search between the bases

The metaphor is intensively used. Below is a number of other examples.

Getting in:

7. on levels above three, you automatically *go into* the circulation control system

8. but look, in fact, it is like it let me *come in* where I was before

9. it is also possible for the technical college people *to come in* directly on some lines

10. now I have *to get in* again

Moving around:

11. when you want to get *over from* the circulation control system *to* the information retrieval system

12. If you have been doing something in the circulation control system and then you want *to get over*, shift over to do a search, then it is faster in that manner

13. well, I can also *shift to* another base

Moving objects:

14. = 4. and I can also (...) go in and *call forth* the circulation control system

15. = 6. that is, we *move a search* between the bases

16. (why do you use that phrase?) well, that is because I *move a search* from the DTB-base to for example the DVJB-base (...) as if you *carry* the order around between different card files

17. this screen image *is fetched* from our circulation control system

Being located:

18. since the borrower numbers must *lie in* the circulation control system anyway

19. well, the problem is now that I really don't know what base I
am placed in

The above examples were all from the circulation department but you also find the metaphor at the circulation desk,

20. I want to get *into* number seven, and then I go down in that line there. Then I run this *in*.. Look, now the system is open, and I am *in* it (...) then I can go *into* the information retrieval system placed down there, and I go *in* and want to see if Hanne has got these two

2.4 The conversation partner metaphor

The identification of the conversation partner metaphor is based on the following criterion: The sentence contains speech act verbs normally expressing linguistic acts of human beings, and the computer is either the sender or the receiver. In other words, it is a kind of personification. In most cases the employee is the receiver of the conversation and the computer is the sender:

21. well, then *it asks for* the number of my borrower's card
22. and then *it asks for* that number
23. and then I get *an answer* like that
24. here I become uncertain because *it answers* differently from what I *expect it to do*
25. *it answered* in a way that I'm not used to see
26. what it did now was that it dropped the connection, (.....), without further *comments*
27. if I now write (.....) here, then *it asks* me to write (.....)

The employee may also be the sender of the conversation:

28. then I can *say* F (.....) I choose the UB2-base and the I *say* "find" search-set number 1, and then I *say* "show it"

In a couple of instances she interprets the prompts of the computer and her own responses as a kind of internal monologue:

29. do I want to have it sent? Yes, I do. Do I want a message? Yes, I do. Or do I not?

It seems as if the employee is in charge. She may command the computer, *show it*, whereas the computer only may put requests *then it asks me to write*.

2.5 The organism metaphor

The identification of the organism metaphor is based on the following criterion: The sentence contains verbs like *know*, *make*, *do* and the computer is the subject of the sentence. The organism metaphor has similarities with the conversation partner metaphor. In both cases it is as if the computer is a human being, but in the case of the organism metaphor the computer has a will of its own and an ability to act. In other words it is a kind of animation. One of the employees is aware of the animation:

30. we have a kind of personified perception of the system

Instances of the organism metaphor are:

31. it means somehow that it *knows* itself what it shall do

32. I'm not sure whether it *has shifted*

33. the system *has done* what I expected it to do

34. what it *did* now was that it dropped the connection

35. (what is wrong with number 8, why is that an error?)
apparently it *doesn't want* to make requisitions

2.6 The media metaphor

The identification of the media metaphor is based on the following criterion. The sentence contains speech act verbs, and the sender or the receiver is not the computer but a person, e.g. "he" in the sense of the "designer of the system". The system is seen as a media through which people communicate:

36. all these remarks, the lines, he *writes*, they probably mean something

37. (about an error message) that one *came from* here

38. that one came from here, from our part of the system, it is one that *is put in* from out here

39. then *they notify* that there are three (copies of a book), but we only have two out here

They might be the cataloguers. The use of the word "turn over the leaves" fits well with the illusion that the system is a written media with pages:

40. and then I can turn over the leaves here. Now I *leaf through* it -what can I say, there must be some kind of chronology

2.7 Discussion of the observations

The computer system is understood in terms of at least four different metaphors: The physical space metaphor, The conversation partner metaphor, The organism metaphor, and The media metaphor. The metaphors do not originate from the professional language but from humans' every day experience. The metaphors are rapidly changing, but a general pattern seems to appear: the functional structure of the system is understood in terms of the physical space metaphor, the interaction as the communication partner metaphor, and when something unexpected happens the organism metaphor. The media metaphor doesn't seem to be part of a clear pattern.

The metaphors originate from humans' every day experience and are used to make the intangible computer system tangible. This fits well with observations made by Anatol Holt⁹, who points out the dissimilarities between computer systems and other artifacts from our everyday life:

"Taking the standpoint of primitive man, we soon notice a significant difference between working in an electronic environment and otherwise working in the office, the factory, the store, the highway: in these common settings we work with artifacts, that have bodies as ourselves. We know these artifacts without the intervention of intellect. We recognize them as mobile or fixed, hard or soft, heavy or light, cheap or expensive, usually at a glance. The electronic world is different. We cannot bodily enter it. There we create, manage, use and maintain work related structures by "remote control" - by an array of means which are at once infuriatingly diverse and infuriatingly uniform"

Moreover, Holt points out the importance of the physical organization of the workplace : each area has its dedicated function: the storage place, the assembly stations, the sales office etc. Physical proximity becomes a prerequisite for doing the task. For instance, the tool, the material and the person have to be near each other.

The employees at the library use metaphors as a way of handling the intangible computer system. The metaphors come from areas where they are well experienced: the physical space, other human beings and conversations. They interpret the system in such a way that they can use their previous experiences.

The above discussion supports the idea of using metaphors in design. As designers we always favour an interpretation of the system. We might just as well provide one that explicitly links to the users experiences as reflected in their language. But metaphors are not good in themselves. A metaphor reflects a certain perspective on reality. Any metaphor is a carrier of specific values, in the sense that something is good or bad, important or unimportant, etc. The organism metaphor reflects that the system is in charge or has the initiative. The organism metaphor has similarities with the artificial intelligence metaphor, which makes us think of computers as similar to human beings. This kind of

⁹A. Holt: Primitive Man in the Electronic Work Environment, Presented at the conference on electronic work, Milan, March 1986.

metaphor gives us an erroneous understanding of what computers can do¹⁰.

Another example of inappropriate use of metaphors is the following. If you mail an undeliverable message by the electronic e-mail system you get the following message back: *returned by postmaster at* (for instance) *Seismo*. The postal service metaphor makes you think that a human actually returned the mail. At Stanford University undeliverable electronic mail is returned together with the message: *returned by the mailer demon*. In this way, it is emphasized that the mail is not returned by a human being.

The use of only a single metaphor leads to a narrow view. The tool metaphor has been used in the design of computer support for graphic workers¹¹. Although successful in many respects, the tool metaphor has created a blindness towards other aspects, for instance communication, of graphic work - graphic work is seen as individual work¹².

In the following section, we take current language use as starting point for design, although we are well aware of the fact that the present perspective on the work is not necessarily the most appropriate perspective. For instance, at the Postal Giro in Stockholm the professional language reflects a "paper flow" perspective¹³, but the work could just as well be seen as accounting. The two different ways of understanding a workplace may give rise to quite different computer systems.

3. The design approach design

In this section, we explore the idea of designing interface and functionality with the point of departure of an authentic professional language. We want to design a small command language based on the sentences in 2.3 that use physical space as a metaphor. Thus, we use one of the users' metaphors as it is expressed in their language.

On the one hand, we want to use the professional language of the librarians as our point of departure, thereby making the command language more powerful and easy to use. On the other hand, we consciously introduce changes in the professional language. In Artificial Intelligence, human language is also used

¹⁰H. Dreyfus and S. Dreyfus: *Mind over Machine*, The free Press, New York 1986.

¹¹P. Ehn and M. Kyng: *A Tool Perspective on Design of Interactive Computer Support for Skilled Workers*, DAIMI PB-190, Aarhus University, Denmark, 1985.

¹²See *ibid.* section 4: "Limits of the Tool perspective". And for a classification of perspectives see chapter 9 of Andersen et al.: *Professional Systems Development* (in Danish, *Professionel Systemudvikling*), Teknisk Forlag 1986.

¹³B. Holmqvist och G. Kälgren: *Postgiro som språkmiljö I-II*, MINS 1986.

as a basis for design, and it is instructive to compare the AI approach with ours. An important difference is that AI aim at similarity between the command languages and human languages whereas we take for granted that the command language will differ from the language of the librarians, because the use situations are widely different. Whereas the AI tradition wants to formalize as large parts of natural language as possible, we set it up a choice: how much of the professional language should have a formal interpretation?

This section contains rather detailed discussions about the Danish language. In fact, its main theme is: how can a part of Danish be used as a point of departure for design? This creates problems of translation. Normally, we append an English translation to the Danish example, trying to keep as closely to the original wording as possible. However, in a few cases, where Danish has expressive devices not found in English, we quote the Danish terms, and explain their meaning.

The design approach consists of four steps. In the first step we have collected parts of the authentic professional language. The collection process itself is accounted for in section 2 - in section 3.1 we simply repeat the selected corpus, i.e. our linguistic data. In the second step (section 3.2) we do a linguistic analysis of the corpus that shows which constituents are contained in the sentences. In the third step (section 3.3) we make a simple command language that can be used to perform the commands described by the sentences of the corpus. In step 4 (section 3.4 - the main subsection of this section) we take up the command language for discussion. We set up a number of design choices and describe how we have chosen in our proposal for a command language.

Before we proceed, we point out some of the limitations of the approach as applied in this section. By doing so, we at the same time make suggestions for improvements of the approach or point at directions for future research efforts.

1: Although our corpus is part of the professional language of the library staff the corpus doesn't reflect the profession specific part of the professional language. The last part of this paper grew out of an interest in metaphors and not until after the collection and initial analysis of our linguistic material did we decide to use our material as a basis for design. In a more carefully planned project it would be obvious to start out from the more library-specific language usage, for instance language usage in relation to information retrieval. Moreover, in a real life design project, it would be necessary to consider a more extensive part of the professional language.

2: We have recorded the language *outside* the regular work situation (c.f. section 2) and not the language *in* the work situation well aware of that the two languages are different.

3: The command language we design in this paper is extremely simple and covers only a very limited part of a library system. Our command language is by no means an essential part of this paper. The command language plays the

role of an example in the mediation of the design choices set up in section 3.4.

4: In the first part of the paper we have argued for the relevance of more than a single metaphor when designing a complete system. But in this part of the paper we only design a limited part of the library system and we claim that a single metaphor is feasible.

5: The design choices we set up have grown out of the linguistic domain and in most cases we have succeeded in transposing the choice into the computer domain, but in a few cases we set up linguistic choices without being able to formulate a corresponding choice within the computer domain. This doesn't necessarily indicate that the choice is irrelevant, only that our approach isn't fully developed. These kinds of choices, as well as the other choices, need to be reconsidered and elaborated in a future project.

6: We set up the design choices and hereby we open up the possibility for overlooked design alternatives or we make explicit choices perhaps otherwise only made unconsciously. We do not find it appropriate to make strong and general recommendation for choosing a specific design alternative. In the concrete design project the actual choice should be made in collaboration with the future users. Our approach may in this situation be supported by other design approaches, for instance prototyping.

Well aware of these limitations we now return to our design approach.

3.1 Corpus

Our corpus, i.e. our data, consists of the following sentences:

1. har de måttet lave et andet *adgangs* billede
(they must have made another *access* image)
2. *adgangs* koderne til systemet er sat sammen af to ting
(*the access* codes to the system are made up of two parts)
3. så kunne jeg *gå ind* og lave sikkerhedskopier, og lægge
nogen magnetbånd *ind på* systemet
(then I could *go in* and make back-up copies, and place
some magnetic tapes *on* the system)
4. og jeg kan også *gå ind* og kalde udlånssystemet *frem*
(and I can also *go in* and *call forth* the circulation control
system)
5. så *skal* jeg *tilbage* til ALIS-basen
(then I have to *get back* to the ALIS-base)

6. det er vi *flytter* en søgning *mellem* baserne
(that is we *move* a search *between* the bases)
7. på niveauerne over tre, der *går* man automatisk *ind i* udlånssystemet
(on levels above three, you automatically *go into* the circulation control system)
8. men se den den lod mig altså lissom faktisk *komme ind* der hvor jeg var før
(but look, in fact, it is like it let me *come in* where I was before)
9. det er også muligt for højskolens folk at *komme ind* direkte på nogen linjer, ikke
(it is also possible for the technical college people to *come in* directly on some lines)
10. nu skal jeg lige *ind* igen
(now I have to just get *in* again)
11. når man skal *over fra* udlånssystemet *til* søgesystemet
(when you want to get *over from* the circulation control system *to* the information retrieval system)
12. hvis man har *siddet* og lavet noget *i* udlånssystemet og så man lige skal over, *skifte over til* at lave en søgning, så er det hurtigere på den måde der
(if you have been doing something *in* the circulation control system and then you want to *get over, shift over* to do a search, then it is faster in that way)
13. altså, jeg kan også *skifte til* nogen andre baser, ikke
(well, I can also *shift to* another base)
16. [hvorfor bruger man det udtryk] Jamen, det er fordi jeg *flytter en søgning fra* DTB-basen *til* for eksempel DVJB-basen....at man lissom *bærer bestillingen rundt* mellem nogen forskellige kartoteker
(well, that is because I *move a search from* the DTB-base *to* for example the DVJB-base...as if you *carry the order around between* different card files)
17. det billede der bliver *hentet fra* vores udlånssystem
(this screen image *is fetched from* our circulation control system)

18. da lånernumrene alligevel skal *ligge i* udlånssystemet,
(since the borrower numbers must *lie in* the circulation control system anyway)
19. se det der er problemet nu, det er at jeg rent faktisk ikke ved
hvad base vi *befinder os i*, vel
(well, the problem is now that I really don't know what base we are *placed in*)
20. jeg skal *ind i* syveren, og så går jeg *ned der i* den linje,
ikke. Så *kører* jeg den *ind*. Se nu er systemet åbent, og jeg
er *inde i* det. (...) så kan jeg *gå ind i* søgesystemet som står
dernede, og ellers så *går* jeg *ind* og vil gerne se om Hanne
har de her to
(I want to *get into* number seven, and then I *go down in* that line there. Then I run it in. Look, now the system is open, and I *am in it* (...) then I can *go into* the information retrieval system placed down there, and I *go in* and want to see if Hanne has got these two)

Notice how selective we have been: The examples in 3.1 were only a fraction of the tape recordings, and we have now chosen only the parts that directly concern the physical space metaphor. We do not use the whole professional language as a basis for design.

3.2 Analysis of Corpus

The next step is to analyze our corpus. We do a grammatical analysis of selected parts of the corpus that shows which constituents are contained in the sentences. On the basis of this, we classify the sentences into three main types that are syntactically similar.

The categories we employ are rather traditional. They are defined by a combination of position and meaning¹⁴. We have simplified the Danish sentence structure so that it corresponds more closely to our corpus. In order to give a better overview of the semantic structure of the sentences, we have converted all sentences to declarative sentences, and some infinite constructions have been converted to full sentences. The effect is that constituents with similar semantic function tend to be placed in the same column in the analysis table. We have the following constituents:

¹⁴Paul Diderichsen: Elementær Dansk Grammatik. Gyldendal 1962. Diderichsen's main idea is that the sequence of constituents(slots) is constant. What varies is the way in which constituents are filled. Thus his grammar is a slot-and-filler grammar. Normally, not all slots will be filled. A slot contains fillers of a certain type, e.g. a particular word class. For example, VBF contains tensed verbs.

1. SUBJECT, in action sentences the animate being that causes the action:

jeg(I), man(you), højskolens folk(the technical college people), vi(we), lånernumrene(the borrowers' numbers)

2. VBF, the finite(tensed) verb. Subject and finite verb constitute the backbone of the sentence:

kunne(could), kan(can), skal(has to, wants), går(goes), kommer(comes), flytter(moves), bærer(carries), kører(runs), er(is), har(has), befinder os(are)

3. ADV, adverbs that express the attitude or perspective of the speaker:

også(also), lige(just), alligevel(anyway)

4. VBI, the infinite (non-tensed) verb. If the finite verb is an auxiliary, VBI contains the main verb:

gå(go), skifte(shift), lægge(place), kalde(call), siddet(sit), ligge(lie), hente(fetch)

5. OBJ, the object. In action sentences the object is the entity affected by the action. Here it is always a noun phrase¹⁵:

nogen magnetbånd(some magnetic tapes), udlånssystemet(the circulation control system), en søgning(a search), bestillingen(the order), det billede(that screen image), den(it), noget(something)

6. TIME/MANNER, an adverbial that describes the manner or time of the movement:

automatisk(automatically), igen(again)

7. DIR_{rel}, an adverbial that describes the movement or position of an object relatively to the previous position of the object and/or the position of the speaker:

ind(in), tilbage(back), ind der hvor jeg var før(in where I were before), over(over), frem(forward, forth), ned(down), rundt(around), inde(in)

8. DIR_{abs}, a prepositional phrase that describes the movement or position of an object absolutely in relation to another object:

til ALIS-basen(to the ALIS base), i udlånssystemet(in the circulation control system), på nogen linjer(on some lines), fra udlånssystemet til søgesystemet(from the circulation control system to the information retrieval system), til at lave en søgning(to do a search), til nogen andre baser(to some other bases), i syveren(in number seven), i den linje(in that line), i

¹⁵ A noun phrase is a constituent whose kernel is a noun., e.g. *The little house on the hill.*

søgesystemet(in the information retrieval system), på systemet(on the system), mellem baserne(between the bases), mellem forskellige kartoteker(between different card files), fra vores udlånssystem(from our circulation control system), i det(in it), i hvad base(in what base)

By means of these categories, our corpus can be analyzed in this way:

I. Actions without object

SUBJECT	VBF	ADV	VBI	TIME MANN	DIR _{rel}	DIR _{abs}
jeg I	kunne could		gå go		ind in	
jeg I	kan can	også also	gå go		ind in	
jeg I	skal have to get				tilbage back	til ALIS- basen to the ALIS base
man you	går go			auto matisk automati- cally	ind into	i udlåns- systemet the circulation control system
jeg I	kommer come				ind der hvor jeg var før in where I was before	
højskolens på folk the technical college people	kommer come				ind in	direkte nogen linjer directly on some lines
jeg I	skal have to get	lige		igen again		ind in

I. Actions without object (continued)

SUBJECT	VBF	ADV	VBI	TIME MANN	DIR _{rel}	DIR _{abs}
man you	skal want to get				over over	fra udlåns systemet til søgesystemet from the circulation system to the information retrieval system
man you	skal want to	lige	skifte shift		over over	til at lave en søgning to do a search
jeg I	kan can	også also	skifte shift			til nogen andre baser to some other bases
jeg I	kan can	også also	gå go		ind in	
jeg I	skal want to get				ind	i syveren into number seven
jeg I	går go				ned der down	i den linje in that line there
jeg I	kan can		gå go		ind	i søgesystemet into the information retrieval system
jeg I	går go				ind in	
det billede this screen image	bliver is		hentet fetched			fra vores udlånssystem from our circulation control system

II. Actions with an object

SUB- JECT	VBF	ADV	VBI	OBJ	DIR _{rel}	DIR _{abs}
jeg I	kunne could		lægge place	nogle magnet- bånd some mag- netic tapes	ind	på systemet on the system
jeg I	kan can	også also	kalde call forth	udlåns- systemet the circulation control system		frem
vi we	flytter move			en søgning a search		mellem baserne between the bases
man you	bærer carry	lissom		bestillingen the order	rundt around	mellem nogle forskellige kartoteker between different card files
jeg I	kører run			den it	ind in	
man you	har have		siddet og lavet been doing	noget something		i udlåns- systemet in the circulation control system

III. Relations without an object

SUBJECT	VBF	ADV	VBI	DIR _{rel}	DIR _{abs}
jeg I	er am			inde	i det in it
vi we	befinder os are placed				i hvad base in what base
lånernum- rene the borrowers' numbers	skal must	alligevel anyway	ligge lie		i udlånssystemet in the circulation control system

The sentence analysis makes it possible for us to record the paradigms that underlie our corpus¹⁶. A paradigm represents the options the language user has when speaking. Part of the meaning of a sign comes from the other signs that could have been chosen instead of the actual one: meaning is choice. The paradigm describes how its members are mutually delimited by contrasts and similarities. We want to become conscious about the meaning changes we introduce in our command language, and we do this by recording how we change the linguistic paradigms.

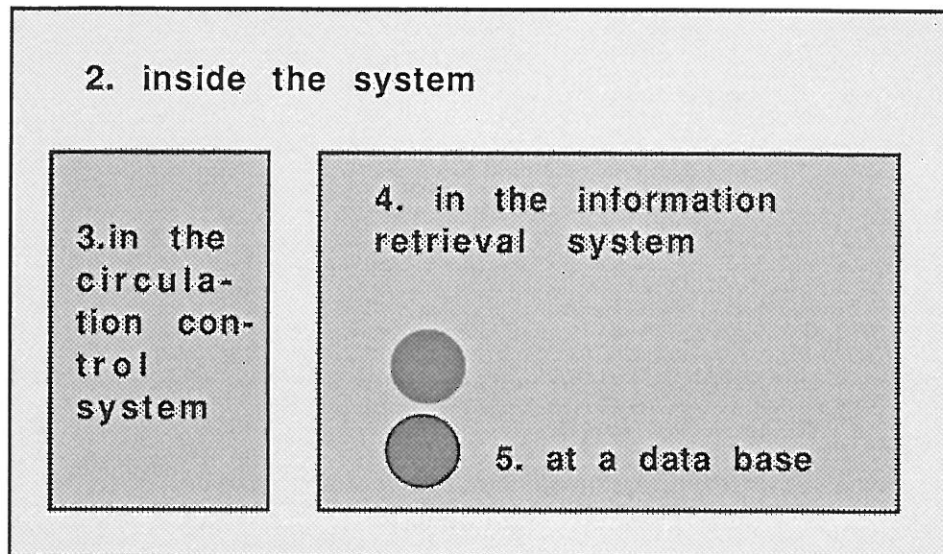
3.3 The command language

We now make a small and simple command language that can be used to perform the operations described by the above sentences. Furthermore, we will try to design the command language in such a way that it supports the physical space metaphor on which the sentences are based. In section 3.4 we use the analysis from section 3.2 to discuss the design.

We conceive the computer systems as consisting of physical spaces or areas in which we can move around. We have defined five areas, and in the following we give a description of the corresponding commands. The five areas are: 1. outside the system, 2. inside the system, 3. in the circulation control system, 4. in the information retrieval system, and 5. at a data base:

¹⁶ A paradigm consists of phrases that can replace each other in a sentence. They can fill the same "slot". To the constituent/slot VBF corresponds the paradigm *kunne, kan, skal, går, kommer, flytter, bærer, kører, er, har*.

1. outside the system



In the following, we describe the commands belonging to each area. We focus on commands that enable the user to move around in the system. The command language is menu based.

COMMAND LANGUAGE, DANISH VERSION

1. Uden for systemet

GÅ IND : Du befinder dig herefter i systemet (område 2)

GÅ IND I UDLÅNSSYSTEMET : Du befinder dig herefter i udlånssystemet (3)

GÅ IND I SØGESYSTEMET : Du befinder dig herefter i søgesystemet (4)

2. I systemet

GÅ UD : Du befinder dig her efter uden for systemet (1)

GÅ IND I UDLÅNSSYSTEMET : Du befinder dig herefter i udlånssystemet (3)

GÅ IND I SØGESYSTEMET : Du befinder dig herefter i søgesystemet (4)

3. I udlånssystemet

GÅ UD : Du befinder dig herefter uden for systemet (1)

GÅ OVER I SØGESYSTEMET : Du befinder dig herefter i søgesystemet (4)

GÅ TILBAGE TIL SØGESYSTEMET : Du befinder dig herefter i søgesystemet (4) med den søgeprofil du havde da du sidst var i søgesystemet

4. I søgesystemet

GÅ UD : Du befinder dig herefter uden for systemet (1)

GÅ OVER I UDLÅNSSYSTEMET : Du befinder dig herefter i udlånssystemet (3)

GÅ HEN TIL BASE1: Du befinder dig herefter ved base1 (5)

5. Ved en base

GÅ UD : Du befinder dig herefter uden for systemet (1)

GÅ OVER I UDLÅNSSYSTEMET : Du befinder dig herefter i udlånssystemet (3)

FLYT REFERENCE OVER TIL UDLÅNSSYSTEMET : Du befinder dig herefter i udlånssystemet (3) med den reference du sidst fandt

GÅ HEN TIL BASE2: Du befinder dig herefter ved base2 (5)

FLYT SØGEPROFIL HEN TIL BASE2 : Du befinder dig herefter ved base2 (5) med den samme søgeprofil som før

GÅ TILBAGE : Du befinder herefter ved den base du var sidst

FIND : Lav en søgeprofil og foretag en søgning

SØG : Foretag en søgning med den forhåndenværende søgeprofil

COMMAND LANGUAGE, ENGLISH TRANSLATION

1. Outside the system

GO IN: You are now in the system (area 2)

GO INTO THE CIRCULATION CONTROL SYSTEM: You are now in the circulation control system(3)

GO INTO THE INFORMATION RETRIEVAL SYSTEM: You are now in the information retrieval system(4)

2. In the system

GO OUT: You are now outside the system(1)

GO INTO THE CIRCULATION CONTROL SYSTEM: You are now in the circulation control system(3)

GO INTO THE INFORMATION RETRIEVAL SYSTEM: You are now in the information retrieval system(4)

3. In the circulation control system

GO OUT: You are now outside the system(1)

GO OVER IN THE INFORMATION RETRIEVAL SYSTEM: You are now in the information retrieval system(4)

GO BACK TO THE INFORMATION RETRIEVAL SYSTEM: You are now in the information retrieval system with the search profile you had last time you were there

4. In the information retrieval system

GO OUT: You are now outside the system(1)

GO OVER IN THE CIRCULATION CONTROL SYSTEM(3)

GO TO BASE1: You are now at base1(5)¹⁷

5. At a base

GO OUT: you are now outside the system(1)

GO OVER IN THE CIRCULATION CONTROL SYSTEM: You are now in the circulation control system(3)

MOVE REFERENCE OVER TO THE CIRCULATION CONTROL SYSTEM: You are now in the circulation control system(3) with the last reference you found

GO TO BASE 2: You are now are now at base2(5)¹⁸

MOVE SEARCH PROFILE TO BASE 2: You are now at base 2(5) with the same search profile as before

GO BACK: You are now at the base you were last time

In addition, the system will contain search commands as

FIND....: Create at search profile and do a search

SEARCH: Do a search with the current search profile

We will not in detail treat these commands here.

3.4 Discussion of the command language

We have now designed the first version of a command language by changing the system¹⁹ underlying our corpus. We want to become conscious of these changes in order to criticize and change our design proposal. We do this by setting up design choices and describing how we have chosen in the present design. Some design choices are only described by setting up a paradigm and commenting upon the changes we have made. In the remaining cases, we describe two opposite strategies and we indicate how we have chosen.

We have grouped the choices in five main classes. The first two classes concern language functions that express our interpretation of the world and our interaction with other people, and include classical language functions as transitivity, mode, and modality. The rest concerns lexical paradigms, syntax,

¹⁷The Danish command contains an adverb *hen* which we cannot translate into English

¹⁸The Danish command contains an adverb *hen* which we cannot translate into English

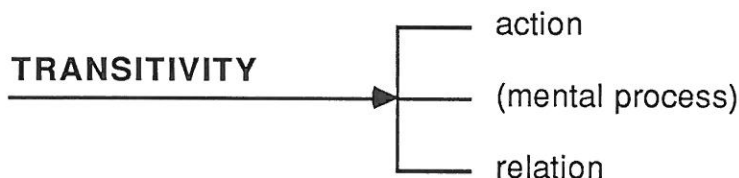
¹⁹A description of the *language system* behind a language use is a description of the *choices* that are realized in the language use.

and reference. In the classification, we have used ideas from the works of M.A.K.Halliday²⁰.

Language functions that express our interpretation of the world

The first choice we treat is transitivity:

(1)



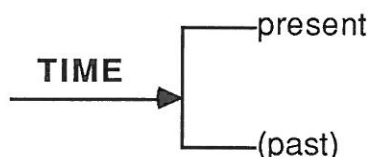
Transitivity is a name for a set of linguistic choices that concern the ability of a language to discriminate and describe different types of processes. Halliday 1976 distinguishes between the following sentence types: action, mental process and relation.

In our corpus, eight main verbs describe an action: *komme(come)*, *flytte(move)*, *bære(carry)*, *køre(run,drive)*, *gå(go)*, *skifte(shift)*, *lægge(place)*, *kalde(call)*, and three a relation: *sidde(sit)*, *ligge(lie)*, *befinde sig(be placed)*.

Obviously, actions can be used as a blueprint for commands, but what can relations be used for? One possibility is to use them as questions, e.g. *Hvor sidder jeg(Where do I sit)?* or *Hvor ligger lånumrene(Where do the borrower numbers lie)?* Another possibility is to use them as feed-back after a command has been issued: *Vi befinder os nu i udlånssystemet(We are now in the circulation control system)*.

COMMAND LANGUAGE: all commands are actions. Relations are used for describing the state of the system after the command is executed: the result of "GO OUT" is described by "You are now outside the system". We have discarded sentences of the mental process type since they belong to the organism metaphor. (c.f. section 2.4).

(2)



Most languages have a tense system with a present tense ,for example, *talks*

²⁰ Here and in the following, we adopt Halliday's descriptions unchanged. We are aware that it is quite problematical to transfer a description of English to Danish, but it is outside the scope of the present paper to develop a systemic description of Danish. A parenthesis means that the choice is not exploited in our corpus.

and a past tense, *talked*. The tense system has a time interpretation, but can also express other semantic dimensions, e.g. the difference between the actual and the potential.

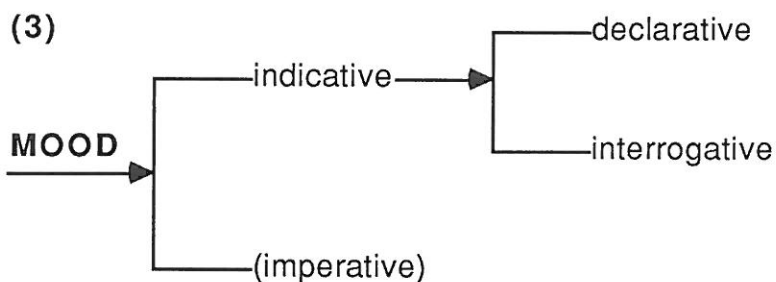
Most verbs in our corpus are in the present tense, but there are two examples of *kunne(could)* in the past tense. The meaning of these past tenses is not temporal but potential: *jeg kunne lægge nogen magnetbånd ind på systemet(I could place some magnetic tapes on the system)*.

In the command language, past or perfect tense could be used in questions about commands that have just been executed: *Hvor var jeg før? (Where were I before)* or *Er jeg kommet ind i udlånssystemet(Have I come into the circulation control system)?*

THE COMMAND LANGUAGE: tense is not used in the commands, since we have chosen imperative mood, and imperatives cannot be tensed. This is worth noticing: a choice in one place of the linguistic system can force us to other choices other places in the system: we have chosen imperative mood instead of e.g. declarative, and this prevents us from exploiting the tense system.

Feed back is declarative, and in one case, we use a tense contrast between present and past time(*the search profile you had last time you were there(3)*).

Language functions that express our interaction with other people



The declarative function is expressed by the word order subject + finite verb: *He throws the ball*. Interrogative yes/no function is expressed by the reverse word order, finite verb + subject: *Does he throw the ball*. Imperatives can be expressed by imperative verb and missing subject: *Throw the ball*.

In the corpus, most sentences are declarative. The user obviously talks to us *about* the work - she tells us about what *she or we do*.

Normally, command languages are imperative. This means that the subject is left out and that the user addresses the system and tells it what it must do. But, if more persons are working together, it may be important to be able to distinguish between different subjects in the command language: I am doing this, and you are doing this. Even if it is hard to notice, we do have a mood choice:

(4) IMPERATIVE OR DECLARATIVE?

Request to the computer about what it should do
(imperative mood with empty subject)

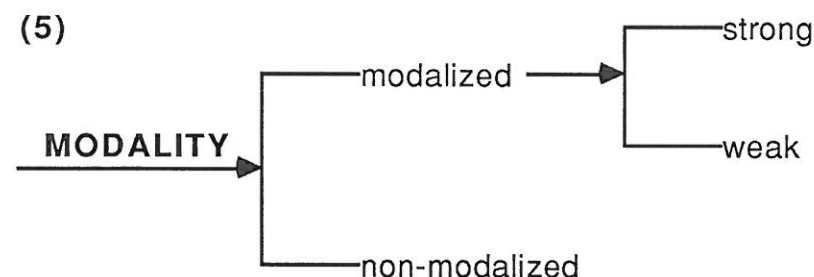
vs

Description of who does what with the computer now
(declarative mood with explicit subject)

A similar choice is known from recent developments in programming languages, where traditional imperative languages such as Pascal have been supplemented with declarative languages like Prolog. The idea is that the user should only need to state the problem, not to specify the exact algorithmic steps that solve it. The same argument could be applied to our command language.

The interrogative mood may be used to ask about the state of the system: *Er jeg ved en base?* (*Am I at a base?*).

COMMAND LANGUAGE: in the command language, we have chosen imperative mood with empty subject. We have chosen not to deviate from the normal imperative mode, since we want to create the illusion that the user moves a thing or person around in space by commanding them to move. At the same time, we have chosen to adhere to the conception of library work as single person work, and therefore we do not need to specify who does it.



The finite verb in modalized sentences is a modal verb. The modality system offers complex choices, but we have only shown the choice between strongly committed and weakly committed modality. The first one is expressed by *should* or *must*: *I must go*, the second one by *could* or *might*: *I might go*.

Many sentences in the corpus are modalized. The main modality distinction is between what it is possible to do (weak commitment, expressed by the Danish verb *kunne*(*could, might*)) and what I intend to do (strong commitment, expressed by the verb *skulle*(*should, must*²¹)).

In the command language, weak modality may be exploited in facilities that

²¹The translations are only approximate, the Danish and English modality systems being rather different.

make it possible to check if an operation is possible or feasible without committing oneself to do it, e.g. because it might have undesirable irrevocable consequences.

The commands in menus reflect a weak modality, as opposed to prompt-response sequences where the user is forced to do certain things (strong modality): ENTER PASSWORD.

COMMAND LANGUAGE: we have the rudimentary form of modality: placing commands in a menu means "possibility". The feed-backs are always non-modalized: they are factual information. We do not exploit the strong/weak opposition in the corpus.

Lexical paradigms

Here we will look into the choices of single words within the same semantic field. In principle, three things may happen: meaning distinctions may remain, disappear or new ones may be introduced.

SYSTEMATIC AMBIGUITY OF COMPUTER BASED SIGNS

A command language always has two interpretations: it has a formal meaning based upon the data processing it controls, and it has a real interpretation based upon the user metaphor we support. These two need not coincide, and can be weighted differently. The choices described below all depend on how we weight the two kinds of interpretations.

The professional language can be used for generating ideas of data processing that would otherwise not have come to mind. To give an example, consider the adverbs: *ind(in)*, *frem(forward)*, *tilbage(back)*, *over(over)*, *rundt(around)*, *ned(down)*, *inde(within)*. Can we imagine data processing facilities that will give *Gå ind i udlånssystemet* (*Go into the circulation control system*) a formal meaning that is different from *Gå ud i udlånssystemet* (*Go out in the circulation control system*)? Yes we can:

1. A space can have the same name as one of its subspaces. The name of the room may be X, we are placed in one of X's subspaces called Y, and Y has itself a subspace called X. If we are placed in Y, *Gå ud i X* will bring us out into the largest space, whereas *Gå ind i X* will bring us into the smallest room.

2. We could make the system issue error messages telling us that we have a wrong conception of our position in the system. For example, we are placed in space B that is a subspace of space A. We write *Gå ind i A* (*Go into A*), and the system writes back: *You have a wrong conception of your position. Room A is outside room B.* The same argument can be made with *over/under* and *frem/tilbage*.

If we place the emphasis upon the formal interpretation of the command language, we will only include the adverbs if the system in fact has this functionality.

However, we could perfectly well choose to give high priority to supporting the

spatial metaphor of the user, and include the adverbs without a corresponding functionality. Thus, we have the following general choice:

(6) WEIGHTING THE AMBIGUITY

Place the emphasis upon the formal interpretation
vs
place the emphasis upon the real interpretation

COMMAND LANGUAGE: from the adverbs of the corpus, we have only included *ind*, *over*, *tilbage*. We have added two new ones, *hen* and *ud*. In our choice, we have placed most emphasis upon the metaphor, that is, the real interpretation. In most commands, the adverbs can be deleted without any functional difference, since the commands can be differentiated by other means than the adverbs. Only in one case, two commands are distinguished functionally by means of the adverbs: GÅ UD/GÅ TILBAGE(GO OUT/GO BACK). This choice is not unreasonable in menu based interaction where command names are written by the system, but would be rather awkward in command based interaction, where full commands are input by the user. In our design, the formal and real interpretation do not coincide.

DELETING PARADIGM MEMBERS.

In many cases, metaphorical transfer of words into the computer domain will mean that meaning differences disappear. Let us start out with the verbs as an example:

The corpus includes the following verbs: *komme*(come), *flytte*(move), *bære*(carry), *hente*(fetch), *køre*(run,drive), *gå*(go), *lægge*(place), *sidde*(sit), *ligge*(lie), *befinde sig*(be placed). These verbs belong to a lexical paradigm that can be called "transport and location". The library staff uses this paradigm as a part of the metaphor for their system.

What should happen to this paradigm when we include it in our command language? The first thing to ask is, whether the existing metaphorical usage has not already removed some meaning distinctions: *Man går ind i systemet* (You go into the system) may have become synonymous with *Man kommer ind i systemet* (You come into the system)? *Jeg flytter søgningen over i DTB-basen* (I move the search into the DTB base) may have become synonymous with *Jeg bærer søgningen over i DTB-basen* (I carry the search into the DTB-base)?

In non-metaphorical usage, *gå* and *komme* differ in the orientation of the movement in relation to a point of reference, often the place of speaking. Both verbs signify a movement from an initial to a final position. They differ in that with *gå*, the initial position is identical to the point of reference ("here"), whereas with *komme*, the point of reference is the final position. *Han går* can mean "He removes himself from the place of speaking".

In non-metaphorical usage, *flytte* and *bære* both signify that somebody transports an object. They differ in that *bære* requires the subject to be immediately involved in the transport, whereas this is not required by *flytte*. For example, if a person transports a piano in a truck, *flytte* can be used, but not *bære*..

Our last example is the prepositions. The corpus includes the following prepositions: *fra*(from), *til*(to), *i*(in), *på*(on), *mellem*(between). If we place the emphasis upon the formal interpretation, we are tempted to reduce *i* and *på* to one meaning. It does not seem possible to devise a formal meaning that will differentiate *Læg Post3 i ALIS-basen*(Place Record3 in the ALIS base), and *Læg Post3 på ALIS-basen*(Place Record3 on the ALIS base) .

Given that we have chosen to place the emphasis upon the formal interpretation, there are two opposite strategies for converting a lexical paradigm from the professional language to a command language:

(7) PRECEDENCE OF FUNCTIONALITY AND METAPHOR

Assign a formal interpretation to as many meaning distinctions of the professional language as possible, and design the functionality on this basis. ("Here/not-here", "immediate/mediate subject" must be assigned a formal interpretation)

vs

First design the functionality, then reduce the lexical paradigm to exact those members that have a formal interpretation in the functionality. (Reduce the paradigm of transport and position to two members, *gå* and *flyt*, and exploit the distinction that with *gå* the subject is moved, whereas with *flyt* the object is moved).

COMMAND LANGUAGE: for the the most part we have started out from the functionality and reduced the paradigms accordingly. We have removed *komme* from the transportation paradigm -*here* is always the initial position. And we have not exploited difference between *flytte* and *bære* . We have reduced the preposition paradigm to only two prepositions: *i* and *til*. The rest are removed. *i* and *til* have no formal interpretation, their meaning is only defined in relation to the user metaphor.

Metaphorical usage always causes meaning distinctions from the literal domain to disappear in the metaphorical domain. Here are some other examples.

In the spatial world, there is a speed difference between *go* and *run*, but this difference has disappeared in the world of data processing where they both mean start a program's execution. *Run* is not faster than *go*, they are mere variants.

When we describe the writing process, we distinguish between *writing* (something new) and *adding* (material to something we have already written). In a particular editor, the concept of *writing* has disappeared, being reduced to a borderline case of *adding*, namely adding material to an empty text.

In an administrative hospital system, patients cannot be *transferred* from one department to another: they have to be *discharged* from the first department and *admitted* to the second department. The three member paradigm of *admit*, *transfer*, and *discharge* has been reduced to a two member paradigm.

Finally, the distinction between copy and original has lost its meaning in the

world of data files, whereas it is rather important outside: one gets into trouble when trying to pay with a copy of a hundred dollar bill.

The designers of Xerox Star define a type of commands they call universal commands²². This command type can be viewed as a conscious strategy for removing meaning distinctions that are considered unnecessary. Universal commands are commands that can operate on many different objects. For example, the CUT, COPY, and PASTE of the Macintosh can be used to edit text, diagrams, movies, spreadsheets, etc.

Removing paradigm members means that the remaining ones acquire a more general meaning. The expression *double click/open* is used in two situations in the Macintosh Finder: for opening a folder and for opening an application. To open a folder can also be described as *getting access to data*, whereas opening an application can be described as *starting a new program execution*. However, the Macintosh design tries to persuade the user that this is really the same thing. The idea of starting a program execution by *opening* can be seen as a way of pressing program execution into the office or desktop metaphor.

In some cases, the strategy to obliterate meaning distinctions can go too far. One example from the Finder is this: when a folder is dragged from one diskette to another, the folder does not disappear from the source, but is copied into the target diskette. However, if a document is dragged from one folder to another, both residing on the same diskette, it is really moved, that is, it disappears from the first one. In the use situation, there is a need to distinguish between moving and copying, but the Finder expresses both concepts in the same manner: by *dragging*.

When we remove paradigm members by making universal commands we face a dilemma: On the one hand we may, as indicated above, create confusion, but on the other hand we have to weight against the confusion caused by having a large number of similar commands.

ADDING PARADIGM MEMBERS

Consider again the lexical paradigm of "direction and place": *ind(in)*, *frem(forward)*, *tilbage(back)*, *ned(down)*, *over(over)*, *rundt(around)*, *inde(within)*. In descriptions of the Danish language, these paradigms contain far more members. On the one hand, the words are structured in antonym pairs: *ind/ud*, *frem/tilbage*, *over/under*, and on the other hand, Danish has a locative inflection of some of the words: *ind/inde*, *ud/ude*, *frem/fremme*, *over/ovre*. etc.

When designing a command language, it is tempting to aim at orderliness and systematization by completing the paradigms: if something is called *back* there must also be a thing called *forward*. Our concept of what is systematic may originate from our general linguistic knowledge. However, we must remember

²² D.C.Smith et Al.: Designing the Star User Interface. BYTE , April, 1982, p.210-223.

that we have to do with a special language register²³, language in connection to usage of a library system, and this register needs not contain the same members as prescribed by the grammar of the national language. On the other hand, we also have to be aware of that the corpus not necessarily give rise to coherent and appropriate functionality. In the extreme, for instance, our corpus includes *gå ind (go in)* but it doesn't include *gå ud (go out)*. We can set up the following choice:

(8) COMPLETENESS

We make the command language more systematic and complete than the corpus it is based upon. That is, we add *ud* and *under*.

vs

We try to capture the special language usage in the corpus, since we believe it to be significant. We do not add *inde* and *under*.

COMMAND LANGUAGE: Compared to the corpus, we have added two members, *ud* and *hen*, but not *frem(forward)* the antonym of *tilbage(back)*. We have tried to retain the meaning of the words. For example, *over* is used about movements between two not too distant positions on the same level, whereas *hen* is used about movements between two proximate positions. In one case, we have changed the meaning of a word: *gå tilbage(go back)* has got the additional meaning that the place one goes back to is unchanged from the time it was visited last. This meaning is not part of the word outside the command language.

When we look at command languages, we will sometime note that new distinctions pertaining to the technical structure of the system have been introduced. In CP/M operating system, copying is standardly done by means of the PIP-command. However, since PIP is slow for copying the content of an entire diskette, many manufactures supply special COPY-commands for this purpose. Thus, the concept of copying has been split up into two subconcepts, PIP and COPY.

Incidentally, this change is the opposite of the *run/go* change mentioned above. In the *run/go* example, a speed distinction in the literal domain is lost in the metaphorical domain. In the PIP/COPY example, a speed distinction has been introduced in the computer domain.

Syntax

Our design exercise gives rise to three type of syntactic considerations:

WHICH TYPES OF SYNTACTIC CONSTRUCTS SHOULD BE PERMITTED?

In our corpus, the adverbs of direction have three variants. They can contain two constituents $DIR_{rel} + DIR_{abs}$: *tilbage + til ALIS-basen(back + to the ALIS-base)*, but there are two one-constituents variants: *ind (into)(DIR_{rel})* and *i*

²³ A register is a language variety that belongs to a specific situation type.

udlånssystemet (in the circulation control system)(DIR_{abs}). The corpus will often contain a much richer syntactic variation than is reasonable to introduce in the command language. The choice can be described thus:

(9) SYNTACTIC PATTERNS

We transfer the syntactic variations of the professional language
as faithfully as possible

vs

We standardize the language

In most command languages there is a tendency to favour simple syntactic constructs. For instance, the value of a variable is on the display screen by AUTHOR: Smith and not by AUTHOR is Smith.

COMMAND LANGUAGE: two constructs are permitted: DIR_{rel} + DIR_{abs} and DIR_{rel}, whereas DIR_{abs} alone is prohibited. The reason is that we wish to stress the feeling of space by systematically using DIR_{rel}. Thus, we have removed a syntactic construction in our design..

SHOULD SELECTIONAL RESTRICTIONS BE EMPLOYED?

In natural languages, there are many restrictions on the material that may fill a slot in a sentence. Chomsky's famous example is *Colorless green ideas sleep furiously*²⁴. Ideas cannot be green, they cannot sleep, and nobody can sleep furiously. Selection restrictions are often semantic and express our background assumptions about a normal world. Chomsky's example presupposes that the world consists of disjunct classes of concrete and abstract entities that do not share properties. If we change the selectional restrictions, we change the language's background assumptions about a normal world. Metaphors work by violating the selectional restrictions and momentarily breaking down our conception of normality. In fact, Chomsky's sentence is a very poetic animation(c.f. section 2.2).

Our corpus also contains selectional restrictions. For example, *til* can only be combined with *gå* but not with *lægge*.

If we copy the professional language in this respect, we may violate the design ideal of universal commands mentioned above. Since the point is that a command (for instance *open*) should manipulate many different types of objects (*folders* and *applications*), it means syntactically that universal commands should have few or no selectional restrictions. We face the following choice:

²⁴The term selectional restrictions was coined by N.Chomsky, see e.g. Aspects of the theory of syntax, MIT Press, 1965.

(10) SELECTIONAL RESTRICTIONS

All constituents must combine freely. No selectional restrictions.

vs

The selectional restrictions should mirror those of the professional language.

COMMAND LANGUAGE: we have placed selectional restrictions on the words..*Gå* can combine with alle DIR_{rel} whereas *flyt* (move) only combines with *over* and *hen*. The reason is that functionally it does not make sense to move something *ud*.

The bases only combine with *til*, whereas the subsystems combine with *til* and *i*. The reason is that we want to stress that the bases are not spaces you can be in, but objects you can do something to. In order to further stress the difference between spaces and objects, we ought to have permitted only *i* to be used with subsystems. Thus, we have exploited selectional restrictions to create a background assumption about how our physical space world looks like.

WHAT CONSTRUCTS MAY A WORD ENTER INTO?

In the corpus, a noun may play more than one grammatical role. For example, the circulation control system can be object in *call forth the circulation control system* and adverbial of direction in *go into the circulation control system*. The formal meaning seems to be the same, but the metaphorical interpretation is different. In the first case, the circulation control system is seen as an object the user can move to herself, whereas in the second case, the user herself is an object that is moved to the circulation control system, which now is a location. The grammatical role structure is a device for displaying the same fact in different perspectives²⁵, and, like the selectional restrictions, it expresses our background assumptions about the normal world. In our design, we have the following choice:

(11) GRAMMATICAL ROLE STRUCTURE

We only permit a word to play exactly one role. For example, we can decide that the circulation control system is always a space, never an object.

vs

We permit a word to play more than one role, depending on the task, and on which perspective seems natural.

²⁵ The theory of case is primarily connected to the work of Ch. J. Fillmore. Two important papers are *The Case for Case* (i: E.Bach & R.T.Harms: Universals in Linguistics, 1968) and *The Case for Case Reopened*(i: P.Cole & G.M.Sadock: Grammatical Relations (Syntax and semantics vol.8), 1977). In Danish, Hanne Ruus has used case theory on authentic text: *Sproglig Betydningsanalyse*, NyS 10, 1979.

COMMAND LANGUAGE: we have chosen the first alternative. All subsystems and bases are adverbs of direction, and all references and search profiles are objects. We want to create an unambiguous space. A space is always a space, an object always an object.

Reference

There are differences in the manner in which the system is classified by the corpus and the manner in which a designer will classify it. The semantic fields of the library staff and the computer scientist are different. Let us once more look at the adverbs of direction and place. From the perspective of computer science, these adverbs denote entities of very different type and size:

programs (*udlånssystemet, the circulation control system*) **vs.** data (*ALIS-basen*),
large modules (*udlånssystemet*) **vs** small modules (*syveren, number seven*),
logical invisible entities (*udlånssystemet*) **vs** visible signs on the screen (*den linje, that line*).

Still, they all play the same role in the sentence: they are locations, and they can be combined with similar verbs and prepositions: *Jeg skal ind i syveren* (I have to get into number seven), *Jeg går ned der i den linje* (I go down in that line), *Jeg kan gå ind i søgesystemet* (I can go into the information retrieval system), *Jeg skal tilbage til ALIS-basen* (I have to get back to the ALIS base).

In a similar way, our informants do not seem to separate physical and logical aspects of the system: compare *lægge nogen magnetbånd ind på systemet* (physical) og *lånenumrene skal...ligge i udlånssystemet* (logical). When a designer designs a command language, he must consciously take a stand as to whether the classification of his own language is so important that it ought to be adopted by the users. The corpus motivates at least two serious decisions. The first one concerns the naming of the logical and physical aspects of the system:

(12) NAMING PHYSICAL AND LOGICAL ASPECTS

We use the same words for physical and logical entities. We use the same verb when we place a data object in a file, and when we place a physical tape in a drive. For example, we say *Place Record5 in the information retrieval system* and *Place Tape5 in drive 2*.

vs

We use different words for the logical and physical entities, since we want to stress that the physical and logical architecture of the system needs not be identical. Thus, we say *Place Record5 in the information retrieval system* but *Mount Tape5 on Drive2*.

The other choice concerns the naming of the computer system and the part of the world referred to by the system:

(13) NAMING THE SYSTEM AND ITS REFERENCE

We want to use the same metaphor for the computer system and its reference. For example, we use the term *document* about the paper document as well as its electronic representation. The system should be conceived of as a model of the real document file.

vs

We use different expressions about the computer system and its reference. For example, we use the word *record* about the electronic representation and *document* about the paper document.

THE COMMAND LANGUAGE: we have not designed the part of the system where the problem shows up, but we give an example from another system.

One of the authors works in the project "Professional languages in change" that aims at investigating language change in a department at the Postal Giro Office in Stockholm, Sweden. Previously, the department worked with paper. The employees took care of the customers' paying-in forms, and prepared them for optical reading and computer processing. The real accounting was done outside the department in a data processing department. Then the forms returned to the department, and the workers paired the receipt forms with statements of account, put them in envelopes, and mailed them back to the customers. In 1986, work stations were introduced in the department. The forms are read into the system with an optical reader. From the workstations, additional data are input and faulty data are corrected. Then the electronic representations are sent to the data processing department via a local network. Simultaneously, the paper forms are dispatched in plastic boxes. To a large extent, the previous paper flow has been used as a metaphor for the new system.

In the system, forms are called *document* and boxes are called *lådor* (boxes), no matter if they are paper or electronic. The system designer distinguishes between them by means of the adjectives *fysisk*(physical) and *logisk*(logical). Status reports talk about *fysiska dokument* and *logiska dokument*, and it is explicitly stated whether a box *har sänts fysiskt*(has been sent physically). The command for transmitting data through the local network is called *Sänd via data-com*. However, the work language of the workers seems not to have assimilated this distinction. The workers seem to employ two different verbs, *sända* and *skicka*, instead of the adjectives *fysisk/logisk*. When a physical plastic box is sent, it is called *skicka en låda*, while transmitting an electronic box is called *sända en låda*²⁶, probably copied from the command. A preliminary analysis of the manuals (which are written by the workers

²⁶Berit Holmqvist, personal communication

themselves) seems to indicate that the physical/logical distinction is not important here either. The distinction appears from the context or is irrelevant.

Both of the above choices depend upon whether we want to signify that the computer-based system is homogenous, or whether we want to give the opposite message, that it is heterogenous and consists of parts of different types. We can generalize in this manner:

(14) HOMOGENEITY

The metaphor defines the general perspective in which we see the system, and the user should be able to interpret and handle each part, program or data, large or small, paper or electronic, visible or invisible, within this metaphor. We may want to switch to other metaphors, but they too must be complete. The system is homogeneous.

vs

The metaphor should only cover certain parts of the system, e.g. the structure of large modules. When we work with other parts we should shift to other metaphors. The system is heterogeneous.

The discussion becomes of especial importance when a computer system is to be used by two different professions or in two different departments. At the library, the bibliographical database is based on one metaphor in the catalogue department (forms with lines), but on a different one in the circulation department (entry cards with fields). We face a dilemma. On the one hand we can argue for the use of different metaphors in the different departments because the tasks are different. On the other hand we can argue for the use of a single metaphor, or a number of coherent metaphors, to facilitate communication and to make it possible to take over the work in another department.

A related, but different situation, you find in case of the Macintosh. On the Macintosh the specific applications are typically based on *one* metaphor, a desktop with folders and documents, whereas the technical reference manual "Inside the Macintosh" is based on other metaphors, road map, forks etc. We have two different metaphors, one for the user profession and one for the computer profession. In this case you may argue for the use of two different metaphors since the two metaphors are for two quite different purposes.

COMMAND LANGUAGE: up till now, we have only employed one metaphor, viz. the physical space metaphor. We work with a homogeneous system. However, search facilities must be added to the system. This can be done in two ways: we can stay within the space metaphor, and add commands like SEARCH and FIND which belong naturally to the space metaphor. Alternatively, we can bring the communication metaphor, also found in our data, into play, and add interactions of the type question/answer. Then the system becomes heterogeneous. It is not difficult to think of natural ways to connect the two metaphors. For example, we can create an illusion of a space inhabited by clerks at the data bases, who can be asked questions and give answers. However, other aspects of the system cannot be expressed within either of the two metaphors, e.g. the power aspect. A library system embodies certain power relations between its users. Our system contains seven levels of rights, and each password belongs to a level. We are aware of the problem, but do not take it up for discussion in this paper.

3.5 Summary of the design approach

Professional language is a good point of departure for design, because a detailed analysis of small parts of the professional language can give ideas for both interface and functionality. Moreover, it becomes clear in what respect the system is a continuation of existing verbal habits and to what degree it deviates from them. Using metaphors from domains where the users have good experience can enhance their possibility for controlling and mastering the system.

Taking the point of departure in the professional language opens up a large number of design choices. Many of these choices can be summarized as a decision about the role we allow the professional language to play in systems development:

We use the professional language and its metaphors as an interpretation and disguise of a functionality that has already been defined by other means.

vs

We use the professional language and its metaphors as a basis for design of interface as well as functionality. The professional language expresses important experiences of the profession, and all its expressive devices are in principle interesting, and can give new ideas for the functionality.

In our design, we have chosen the first alternative. To a large degree, the metaphor is a disguise of a functionality that was designed beforehand. Many of the possibilities for a new functionality we have noticed in the corpus have not been exploited.

The following activities could be borrowed from linguistics and used in design:

- collecting data: to avoid filtering too much linguistic material away beforehand (cf. 2.1)
- delimiting a corpus: to clarify the empirical ground on which the design rests (cf. 5.1)
- analysis of corpus: syntactic patterns, paradigms, functional analysis: to get a systematic overview of the design choices (cf. 5.2)
- discussing the recordings, the analysis and the design with the users: to sharpen their awareness of their language and communicative habits and thereby enhancing their possibilities for taking a stand in the design process and the changes that can/should occur with the introduction of the system.

In this paper, we have exemplified all activities except the last one. Due to lack of time and resources, discussion of recording, analysis, and design has not been part of the project. If the project were to be continued, this would be the next step.