

Udvikling og implementering af webbaserede læringsobjekter vha. Macromedia Captivate

Mikkel Godsk

It-koordinator & ekstern lektor, cand.it
E-læringsenheden & Institut for Informations- og Medievidenskab
Aarhus Universitet
godsk@e-learning.au.dk
<http://www.e-learning.au.dk/>



Mikkel Godsk er it-koordinator ved E-læringsenheden, Aarhus Universitet og arbejder med såvel tekniske som pædagogiske aspekter af udvikling, undervisning og forskning inden for feltet. Endvidere er Mikkel ekstern lektor på Institut for Informations- og Medievidenskab, hvor han underviser i formidling på internettet. Mikkel har i kraft af sit arbejde førstehåndserfaring med anvendelsen og udvikling af e-læringsplatforme og læringsobjekter til blended learning.

Introduktion

E-læringsplatforme stiller typisk nogle værktøjer til rådighed til udarbejdelse af webbaserede læringsobjekter, men desværre kan de færreste systemer imødekomme alle de individuelle ønsker og behov, underviseren, de studerende eller kursets pensum og form måtte have.

I stedet for at tage udgangspunkt i det enkelte kursus eller underviser er udviklingen og implementeringen af læringsobjekterne typisk styret af e-læringsplatformens muligheder og især begrænsninger. For at undgå denne infleksibilitet kan det være nødvendigt at kigge sig om efter supplerende værktøjer, som teknisk set nemt kan integreres i eksisterende e-læringsplatforme, content management-systemer (CMS) eller på standardwebservere.

Programmet Macromedia Captivate 1.0 (herefter Captivate – se Macromedia Captivate 2005 for yderligere information) kunne være et bud på et supplerende værktøj, som i dets relative enkelhed giver den mindre it-kyndige underviser muligheden for *selv* nemt at udvikle og implementere egne, interaktive læringsobjekter til internettet efter egne præferencer. Vha. Captivate kan man ved få klik importere hele PowerPoint-præsentationer, indlejre små interaktive øvelser og tests samt måle de studerendes færdigheder.

Formålet med denne artikel er at berøre det aspekt af e-læringsimplementering, som vedrører indførelse af interaktive, webbaserede læringsobjekter i undervisningen samt at komme med et forslag til, hvordan e-læring i denne form relativt nemt kan udvikles og implementeres af underviseren selv vha. eksempelvis Captivate og en "standard" e-læringsplatform. Jeg håber naturligvis også, at artiklen kan inspirere og give undervisere "blod på tanden" i udviklingen af egne webbaserede læringsobjekter.

Implementering af webbaserede læringsobjekter

Når man taler om implementering af e-læring, er det vigtigt at præcisere, hvad man lægger i begrebet pga. de meget forskellige implikationer hhv. implementeringen af en e-læringsplatform og implementeringen af et læringsobjekt kan have. Implementeringen af en universitetsgennemgående e-læringsplatform eller et LMS forudsætter en række generelle hensyn til organisationens kendetegn, ønsker og behov. Derimod bør udviklingen og implementeringen af læringsobjekter typisk foregå på kursusniveau med hensyntagen til den specifikke under-

viser, de studerende og kursets pensum. Ydermere bør underviseren være central i forhold til udviklingen af det specifikke læringsobjekt, da netop han/hun ligger inde med den faglige ekspertise samt den pædagogiske erfaring i formidlingen af det pågældende pensum.

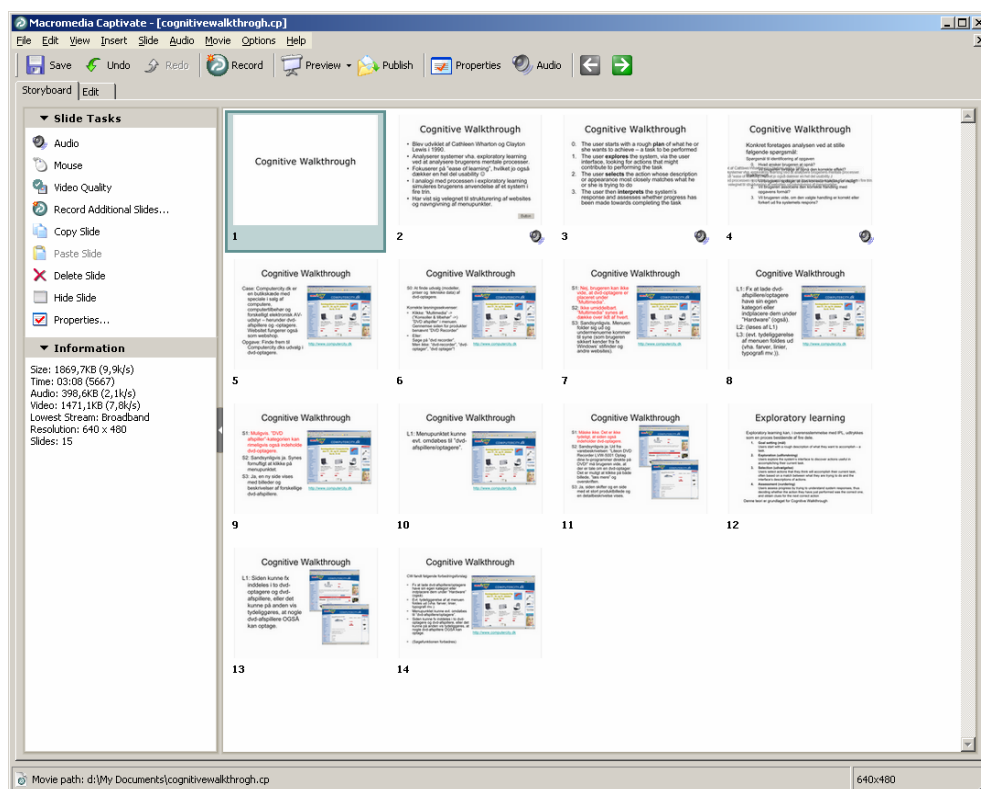
De fleste e-læringsplatforme har en eller anden form for værktøj til udarbejdelse af læringsobjekter, fx et øvelsesmodul, en html-editor, et testværktøj, en læringssti/guidede ture el.lign., som det umiddelbart er muligt for de fleste undervisere at tilgå og anvende. Desværre er disse funktioner ofte utilstrækkelige i forhold til at imødekomme individuelle ønsker og behov, så underviseren må i stedet se sig om efter andre mere avancerede teknologier som fx Flash, Java eller serverside-programmering. Disse teknologier er imidlertid enten vanskelige at integrere i den eksisterende e-læringsplatform eller kræver megen teknisk ekspertise og tidsmæssige ressourcer.

En mulig løsning er, at universitetet stiller et udviklingsteam til rådighed med en e-læringspædagogisk ekspert og en multimedieudvikler, foruden underviseren selv, til selve udviklingen, men en sådan konstellation kan være forholdsvis dyr og infleksibel. En alternativ løsning kunne være, at der tilbydes et udviklingsværktøj til underviseren, som både er brugervenligt, lettilgængeligt og let integrerbart (fx i en eksisterende e-læringsplatform), evt. suppleret med passende support og pædagogisk vejledning i e-læring. Captivate er et sådant udviklingsværktøj.

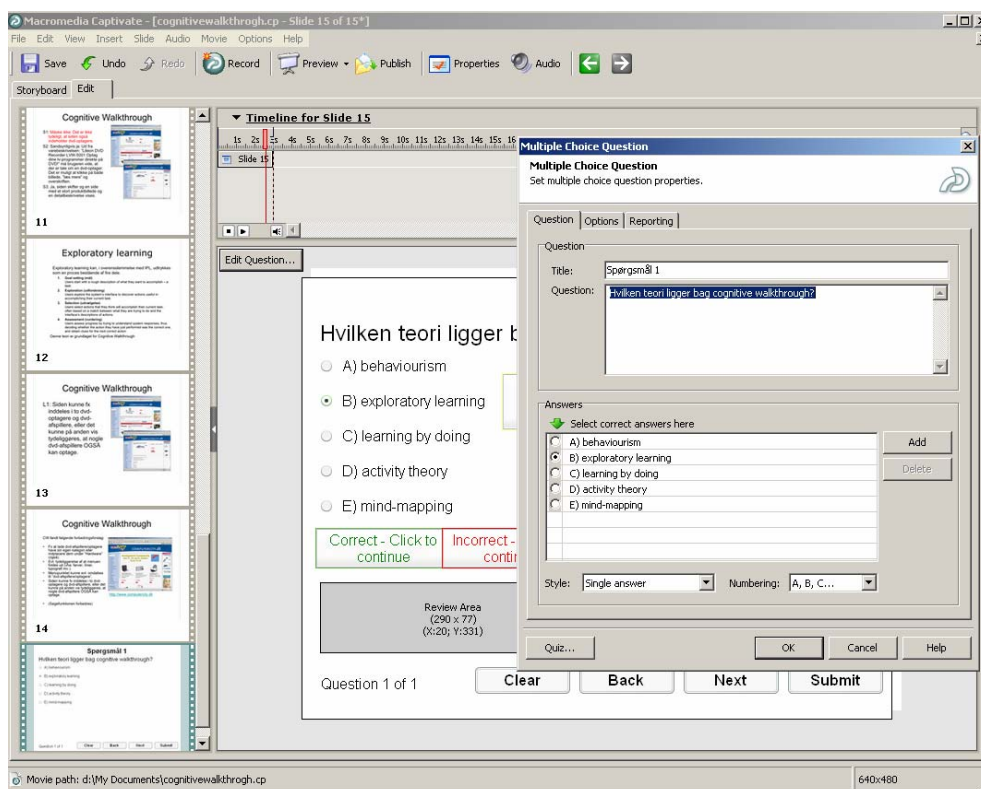
Captivates muligheder og begrænsninger

Captivate er, populært kaldet, et screencapture- eller screenflash-program, hvis ”hovedfunktionalitet” er at kunne optage en video af computerskærmen, evt. suppleret med speak, til fx demonstration af anvendelsen af et program eller et website (Captivate genererer et Flash-objekt – også kaldet en *animation*). Programmet er en videreudvikling af RoboDemo og skiller sig ud fra mere simple screencapture-programmer ved at have forskellige filimporteringsmuligheder samt mulighed for at udvikle og indlejre interaktive elementer som fx tests/quizzes med pointtælling, aktive links, animerede tekstbokse, highlight-bokse og betingede handlinger, samt at time handlinger og synkronisere lyd. En af de meget nyttige funktioner er muligheden for at importere hele PowerPoint-præsentationer, hvorved udarbejdelsen af læringsobjekter ofte kan tage udgangspunkt i allerede eksisterende og gennemarbejdet materiale. Det er ikke på nuværende tidspunkt muligt at få den fulde animation eller interaktivitet overført fra en PowerPoint-præsentation, men Captivate giver en række andre muligheder for at tilføje interaktivitet, animering af tekst, indlejring af eksterne medier mm. Endeligt indeholder Captivate en omfattende publiceringsfunktion, som bl.a. giver mulighed for at komprimere billeder og lyd samt vælge imellem forskellige outputformater.

Captivate er opbygget med et storyboard, hvori præsentationens forløb af ”slides” (dvs. skærbilleder) kan redigeres, samt en ”edit”-del for hvert slide med en tidslinie (”timeline”) til timing af forskellige objekter – fx lyd, quizzes, visning af tekstbokse, pauser mv. Hvert slide har nogle egenskaber, der afhænger af dets type, og som kan justeres individuelt. Det er bl.a. muligt at justere slidets varighed, transition (overgang fra det ene slide til det andet), billedkomprimering og eventuelle musebevægelser (ved screencapture), og ligeledes kan forskellige egenskaber justeres for indsatte objekter.



Figur 1: PowerPoint-præsentation importeret til Captivate. Hvert slide fremstår som et selvstændigt objekt i Storyboard, hvor det kan tilpasses og gøres interaktivt.



Figur 2: Indsættelse af multiple choice-quiz i Captivate. Vha. menuen Insert > Question Slide kan man vælge mellem seks forskellige quiztyper ("Multiple choice", "True/False", "Fill in the blank", "Short answer", "Matching" og "Likert"). Denne figur illustrerer også, hvordan egenskaber kan justeres for indsatte objekter.

Sammenlagt giver Captivates funktioner mulighed for at udarbejde læringsobjekter med, det man kunne kalde, en *konsultativ* interaktivitet, dvs. hvor brugeren navigerer igennem et eksisterende udbud af information (Jensen 1998, s. 202 + s. 232). Det er imidlertid ikke direkte muligt at lade brugeren indgå i en dialog eller videre interaktivitet med læringsobjektet, så mere avancerede interaktivitetsformer, som fx registrerende interaktivitet, hvor objektet løbende tilpasser sig brugeren, er ikke umiddelbart muligt¹. Ligeledes er det heller ikke muligt at lave avancerede simuleringer som fx virtuelle fysikeksperimenter, 3d-simuleringer og andre eksplorative former, uden at kigge sig om efter andre eksterne værktøjer med integrerbare filformater.

Captivates force er derimod de umiddelbare muligheder, man har for at udvikle simple, men dog konsultativt interaktive, læringsobjekter. Ved langt de fleste almindelige behov for fremstilling af online præsentationer af et teoretisk materiale med speak og evt. suppleret med småtests er Captivate fuldt ud tilstrækkeligt.

Pædagogik og e-læringsobjekter

Inden man går i gang med udviklingen af læringsobjekter, er det relevant at overveje sin undervisningsfilosofi, og hvordan man egentlig vil inddrage læringsobjekterne i undervisningen. Vil man anvende læringsobjekterne som informationsressourcer, skal de præsentere pensum og indgå i de studerendes undervisningsforberedelse, eller skal de fungere uden supplerende materiale og konfrontationsundervisning? Forskellige undersøgelser anbefaler forskellige måder at inddrage læringsobjekter i undervisningen afhængigt af bl.a. pensum, så der findes ikke noget entydigt svar herpå. En tendens i anbefalingerne synes dog at indikere en sammenhæng med den bagvedliggende læringsteoretiske tilgang og evt. det aktuelle pensum, og fx viser nyere forskning i problemorienterede og selvstyrede læringsaktiviteter baseret på den socialkonstruktivistiske læringsteori (jfr. bl.a. Vygotskys (1978) og Leontjevs (1978) teorier og Engeströms (1987) ekspliciteringer), at læringsobjekter bør opbygges af læringsressourcer, som anvendes i sammenhæng med andre kilder, frem for at indeholde komplette pensumdele (Dalsgaard 2005b; Koper s. 12 ff.).

Definitioner på læringsobjekter

Læringsobjekter er et centralt begreb inden for e-læringsfeltet og betegner gerne et program, videoklip, dokument, en fil, eller andet materiale, der indeholder noget information, og som vha. dets anvendelse kan omsættes til viden. Der er imidlertid flere, og ikke altid overensstemmende og præcise, definitioner på begrebet. En af de meget anvendte og udbredte definitioner er The IEEE Learning Technology Standards Committees definition: "a learning object is defined as any entity, digital or non-digital, that may be used for learning, education or training" (LTSC 2002, s. 6). LTSC's definition er muligvis *for* bred, da den principielt kan omfatte alle materialer/enheder i hele verden, som kan indgå i læring – og hvad kan ikke det? I modsætning til denne brede definition går Weller et al.s definition: "a learning object is a digital piece of learning material that addresses a clearly identifiable topic or learning outcome and has the potential to be reused in different contexts" (Weller et al. 2003, s. 2) skridtet videre og inkluderer et *emne* og et *resultat*. Denne definition understøtter imidlertid ikke de socialkonstruktivistiske læringsaktiviteter, som baserer sig på studerendes problemorienterede og selvstyrende arbejde (jfr. Hannafin & Land 1997, s. 168+170), da socialkonstruktivismen ser et læringsobjekt som et underordnet element i forhold til den problemorienterede læringsaktivitet og læringsformålet. Det vil sige, at et læringsobjekt potentielt kan indgå i forskellige læringsaktiviteter, og derfor ikke i sig selv kan varetage et indlejret, prædefineret læringsudbytte (Koper 2001, s. 3; Dalsgaard 2005a, s. 2).

¹ Det er ikke muligt at udvikle avancerede simuleringer o. lign i Captivate, men da programmet understøtter import af Flash-objekter, kan simuleringerne eventuelt udvikles i Macromedia Flash og efterfølgende importeres.

For ikke at udelukke bl.a. den socialkonstruktivistiske læringsfilosofi ved en ”problematiske” definition af et læringsobjekt, vil jeg i denne artikel holde mig til LTSC’s definition og supplere med nogle karakteristika særligt for digitale og webbaserede læringsobjekter. Dette giver følgende, operationelle og brede definition på et webbaseret læringsobjekt:

Et webbaseret læringsobjekt er en digital entitet, der kan være sig et program, et tekstdokument, et billede, et filmklip mm., som i kraft af sit indhold kan indgå i eller understøtte en lærings- eller undervisningsaktivitet, helt eller delvist via internettet.

Ud fra denne definition kan et læringsobjekt på internettet fx være et videoklip om, hvordan man tager en blodprøve, en artikel om en metode til analyse af websites, et organisationsdiagram over FN, en online PowerPoint-præsentation eller et avanceret Flash-objekt – som fx er udviklet i Captivate. Disse læringsobjekter kan så frit indtænkes, skaleres og formmæssigt ”åbnes” til den ønskede læringsteoretiske tilgang, pensum og undervisningssituation (hvad enten det er blended learning eller et rent webbaseret kursusforløb).

Hvorvidt læringsobjekterne så skal udvikles som mindre informationsressourcer uden indlejret læringsudbytte, eller objekterne skal fungere stand-alone og indeholde deciderede læringsformål, må være op til bl.a. underviserens læringsteoretiske tilgang og undervisningssituationen. Praktiske erfaringer har vist, at det kan være fordelagtigt at have læringsobjekter med forskellig form til at supplere hinanden af hensyn til den studerendes motivation (jfr. Weller et al. 2003, s. 4 ff.). En fornuftig strategi kunne fx være at formidle det overordnede, teoretiske pensum vha. et såkaldt *narrativt* læringsobjekt, der introducerer til emnet og derfra henviser til andre ”ressource-læringsobjekter”. Disse ressource-læringsobjekter kunne fx være artikler om emnet, eksempler, simuleringer, øvelser, referenceoversigter eller kollaborative opgaver (Weller et al. 2003, s. 4), og det overordnede, narrative læringsobjekt kunne være erstattet af underviserens introduktion til emnet og ressource-læringsobjekterne i en forelæsning.

Eksempel: udvikling og implementering af et læringsobjekt

Hvis man allerede ligger inde med en gennemarbejdet PowerPoint-præsentation af sit materiale, kan denne være et godt udgangspunkt for udarbejdelsen af sit læringsobjekt. Hvis ikke, bør man i stedet overveje at opbygge sit materiale direkte i Captivate for at sikre den bedst mulige billedkvalitet, da Captivate ellers importerer PowerPoint-præsentationer som bitmap-billeder, samt for at bevare mulighederne for redigering af indsatte objekter, som fx tekst, billeder og animationer. Da jeg allerede har en gennemarbejdet PowerPoint-præsentation med objektets indhold, vil jeg anvende den i dette eksempel, men inden jeg importerer den i Captivate, vil jeg tilpasse præsentationen til de særlige tekniske og pædagogiske karakteristika.

Teknisk set skal der tages højde for en række indstillinger vedrørende skærmopløsning, skærmproportioner og båndbredde, ligesom der pædagogisk set er nogle udfordringer i forhold til den særlige formidlingssituation. Måske er der for meget tekst på slidene, for at de er læsbare på skærmen, præsentationen er måske normalt suppleret med eksterne instrumenter (som fx tavle, overhead eller pegepind), som pludselig mangler, eller måske er timingen styret af selve undervisningssituationen. Disse ting er enten umulige at indleje/tage højde for i sit læringsobjekt eller skal inkorporeres på en anden vis. Dertil er der en række overvejelser vedr. undervisningsfilosofien – om fx læringsobjektet skal udarbejdes ud fra en problemorienteret, socialkonstruktivistisk tilgang og kun indgå som en ressource, eller det skal anvendes alene og inkludere et komplet pensum.

I det aktuelle eksempel vil jeg udvikle et læringsobjekt om metoden Cognitive Walkthrough (herefter CW), som kan indgå i Human-Computer Interaction-undervisning. Metoden skal mere konkret indgå i en undervisningsgang i interaktionsdesign, hvor CW er én blandt flere

metoder, som den studerende kan anvende til at løse en opgave i at lave en brugercentreret formidlingsstrategi til et website. Den studerende er altså forinden blevet præsenteret for en problemstilling, som overordnet handler om, hvordan de interaktive aspekter af formidlingen på et website kan målrettes en bruger.

Trin 1: Indsamling og forberedelse af materiale

Fra en tidligere forelæsning har jeg allerede en omfattende PowerPoint-præsentation, som bl.a. andet inkluderer materiale om metodens bagvedliggende teori og forståelseskontekst, faktuelle oplysninger om metoden, selve metoden og 1-2 eksempler på anvendelse af metoden i praksis. Dette materiale vil danne udgangspunktet for læringsobjektet.

Før jeg importerer PowerPoint-præsentationen i Captivate, vil jeg foretage nogle rettelser i PowerPoint, så jeg bl.a. bedre kan styre animationen i Captivate. Først sletter jeg de slides, som ikke direkte er relateret til CW, da sammenhængen med pensum forklares i konfrontationsundervisningen og problemstillingen. Dernæst tynder jeg ud i teksten på de enkelte slides, da der ikke bør være mere end ca. 8-10 linier på hvert slide af hensyn til læsbarheden. Hvis teksten ikke må slettes, kan den fx distribueres på flere slides, placeres i en pop-up-besked eller på et særskilt infoslides, som man kan linke til. Man skal endvidere være opmærksom på, at der ofte er et større behov for at inkorporere uddybende og supplerende informationer i stand-alone læringsobjekter end ved konfrontationsundervisning med PowerPoint, da den studerende ikke umiddelbart kan stille opklarende spørgsmål til lærerens slides.

Herefter skal animationen tilpasses eller ”indfanges”, da den ikke umiddelbart kan overføres til Captivate. Dette kan fx gøres vha. det gratis hjælpeprogram ”Capture Show” (Pillai 2004), som egentlig er en makro/add-in til PowerPoint. Programmet kan generere en billedserie eller en ny PowerPoint-præsentation, hvori alle ”vis”-animationer² er distribueret over en serie af hhv. png-filer eller slides³.

Til sidst importeres PowerPoint-præsentationen i Captivate vha. funktionen ”Import Microsoft PowerPoint slides”, og er så klar til videre bearbejdning.

Trin 2: Planlægning af læringsstrategi

Læringsobjektet bør udarbejdes med hensyn til undervisningssituationen, dets indhold og relation til pensum samt den valgte undervisningsfilosofi. Da jeg i denne sammenhæng har anlagt en problemorienteret strategi, hvor den studerende *kan* anvende Cognitive Walk-through-metoden blandt andre metoder til at løse opgaven, vil jeg opbygge læringsobjektet som et ”åbent” objekt. I forbindelse med dette objekt vil det sige, at det indeholder en gennemgang af metoden og evt. et eksempel på dens brug, men ikke en videre perspektivering i forhold til de andre metoder, afhængighed af andre læringsobjekter eller et prædefineret læringsformål. Objektet skal med andre ord være afgrænset til at præsentere metoden og evt. dens anvendelse på websites.

Trin 3: Opbygning af storyboard og slides

Captivates storyboard tager sit udgangspunkt i den tilpassede PowerPoint-præsentation, som jeg netop har importeret. Rækkefølgen af de importerede slides ser fornuftig ud, så den vil jeg ikke ændre. Jeg vil derimod tilføje et aktivt link til screendumpet af websitet i præsentationens eksempel, en quiz, så den studerende kan afprøve sin opnåede viden, samt et infoslides med supplerende informationer. Såvel det aktive link som infoslides kan udarbejdes vha. en Click Box, da en Click Box både kan linke til websider og til andre, interne slides. Man kan fx tilføje supplerende informationer som ekstra slides sidst i sit læringsobjekt og så linke til

² ”Vis”-animationer er betegnelsen for den animationsform, hvor tekst, billeder, tabeller osv. kommer til syne ved fx klik.

³ Se bilag 1 for vejledning i installering og anvendelse af Capture Show-makroen.

og fra disse slides vha. Click Boxe. Selvtesten vil jeg udarbejde som to simple multiple choice-tests vha. Question Slide-funktionen; en test hvor der skal angives metodens korrekte teoretiske fundament, og en anden test hvor der spørges til metodens anvendelse. Som feedback på selvtesten tilføjer jeg et slide med visning af testresultatet suppleret, vha. Quiz Manageren, med en kommentar om den studerendes standpunkt⁴.

Struktureringen af læringsobjektet er nu klart, så inden jeg går videre med det næste trin, vil jeg med udgangspunkt i mit allerede eksisterende forelæsningsmanuskript foretage nogle initiale rettelser af slidenes timing. En hurtig gennemgang viser, at nogle af slidene kræver længere eller kortere tid end standardindstillingens tre sekunder, hvorfor det rettes for hver enkelt slide i Slide Properties.

Trin 4: Speak-over

Når storyboard mv. er færdigt, er det tid til speak-over. Da jeg ikke er helt sikker på, hvad jeg præcist vil sige til de enkelte slides, tager jeg et enkelt eller nogle få slides ad gangen indtil jeg er tilfreds. Den efterfølgende, og ofte tidskrævende, timing med de viste slides justeres vha. Edit Audio Timing-værktøjet.

Trin 5: Implementering

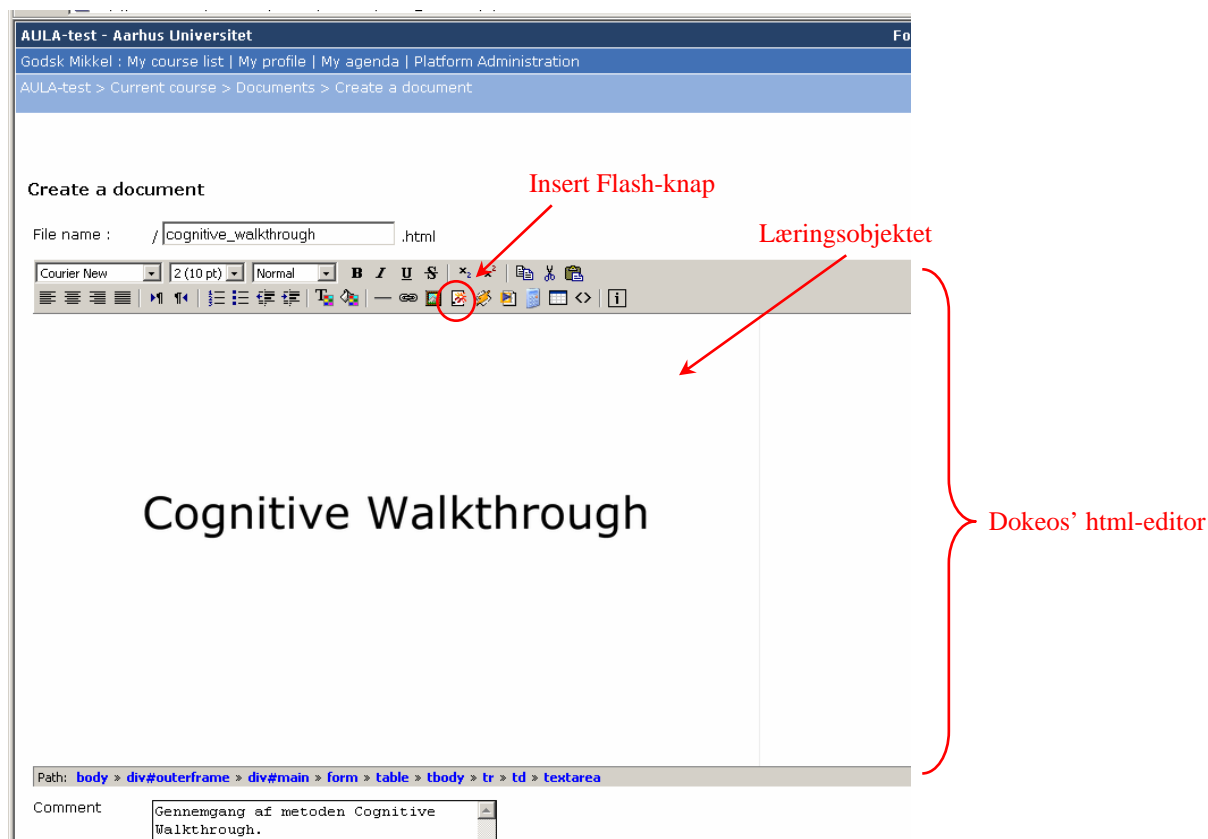
Implementeringen af læringsobjektet gøres ret enkelt og kræver ikke en særlig webserver, specielle webservices eller en særlig konfiguration, da Flash-objekter som dette afvikles ved automatisk at blive downloadet og afviklet i brugerens browser (dvs. client-side). Afhængigt af om læringsobjektet skal indgå i et større *læringsrepository* eller *læringsframework* (jfr. Dalsgaard 2005a) side om side med andre læringsobjekter og med fx statistik på anvendelse og sporing af dets brugere, eller det blot skal være tilgængeligt online, kræves der forskellige tekniske implementeringer. Den enkleste implementering, hvor læringsobjektet blot er tilgængeligt online, kræver kun en standardwebserver eller et CMS, mens en mere avanceret løsning kan være at indlejre objektet i en e-læringsplatform.

Jeg vil implementere det aktuelle læringsobjekt vha. e-læringsplatformen Dokeos, som er Aarhus Universitets e-læringsplatform. Ligesom i mange andre e-læringsplatforme har Dokeos 1.6.1 en html-editor, som anvendes til at lave dokumenter til online anvendelse, og tilmed har denne version af Dokeos direkte mulighed for integrering af Flash-objekter.

Inden jeg kommer så langt, skal den påkrævede båndbredde justeres og Flash-objektet genereres. Vha. menuen Bandwidth Analysis i Captivate får man et godt overblik over den påkrævede båndbredde, som i dette tilfælde holder sig under 128 kB/s for de fleste slides vedkommende. Enkelte slides kræver dog væsentligt højere båndbredde, hvorfor jeg komprimerer de pågældende slides under Quality i Slide Properties. Genereringen af Flash-objektet gøres vha. menuen Publish. I Publish-menuen kan man foretage en række indstillinger om objektets billede- og lydkomprimering, tilgængelighed, forskellige afspilningspræferencer (det er bl.a. muligt at beskytte afspilning vha. et password) samt output-formatet. Da jeg vil anvende objektet på internettet, publicerer jeg som "Flash (SWF)".

Selve implementeringen af læringsobjektet i Dokeos foregår ved, at jeg i dokument-værktøjet vælger "create a document" og efterfølgende indsætter swf-filen via Insert Flash-ikonet i html-editoren. Når dokumentet med Flash-objektet er gemt, er det tilgængeligt online for kurssets deltagere. Hvis ikke man har adgang til en e-læringsplatform, kan man i stedet fx anvende en standardwebserver med fx FTP-adgang.

⁴ Der findes adskillige andre muligheder for rapportering og arkivering af testresultatet, men man skal være opmærksom på, at nogle af rapporteringsformerne kan være i konflikt med bl.a. Internet Explorers Service Pack 2. Overvej derfor, om det er tilstrækkeligt med en "intern" rapportering (hvor resultatet kun vises i selve læringsobjektet).



Figur 3: Flash-læringsobjekt fra Captivate integreret i Dokeos. Vha. html-editoren i Dokeos 1.6.1, kan man umiddelbart uploade og indsætte Flash-objekter, som fx læringsobjekter genereret i Captivate.

I det aktuelle eksempel, kommer læringsobjektet til at ligge på e-læringsplatformen side om side med andre Captivate-læringsobjekter, artikler og eksempler på anvendelsen af andre HCI-metoder. Derudover vil opgavens problemformulering, en række links til eksterne ressourcer samt uddrag af PowerPoint-præsentationen fra den pågældende forelæsning være tilgængeligt (fra og med selve forelæsningsdagen). I selve forelæsningen vil der afsættes tid til, at de studerende kan gå i gang med opgaven i et computerlokale, og underviseren vil være til stede i tilfælde af spørgsmål til såvel pensum og opgaven som anvendelsen af e-læringsplatformen og læringsobjekterne.

Et par tips

- **Anvend en læsevenlig skærmopløsning**

Overvej om præsentationen skal afvikles i full-screen, og hvilken opløsning dine brugere anvender. Pt. (dvs. september 2005) er den mest almindelige opløsning 1024x768 (TheCounter 2005), så et godt råd er at designe til en lavere opløsning, fx 640x480, så alle brugere vil kunne se hele præsentationen uden at den nedskaleres til ulæselighed.

- **Undgå mere end 8-10 tekstlinier på et PowerPoint-slide**, da teksten let bliver ulæselig på en skærm ved nedskalering og ved andre skærmopløsninger end først tiltænkt. Det kan være nødvendigt at dele sin tekst op på flere slides, anvende pop-up-tekst eller infoslides.

- **Indtal spoken i mindre bidder**

Hvis ikke man er helt sikker på, hvad der skal siges og speakens varighed, er det nemmere at finjustere og synkronisere mindre lydstopper.

- **Overvej påkrævet båndbredde**

Captivate giver i menuen View > Bandwidth Analysis et overblik over den påkrævede båndbredde for afviklingen af læringsobjektet. I kolonnen "KB per sec" vises den påkrævede båndbredde for hvert enkelt slide. Objektet vil sandsynligvis godt kunne afvikles af bruger med lavere båndbredde, men i så fald med mulige forringelser og ventetider. I de fleste tilfælde kan den påkrævede båndbredde minimeres ved at sætte Quality til "standard" i Slide Properties.

- **Anvend Capture Show til overførsel af PowerPoints vis-animationer**

Da animationer ellers går tabt ved konverteringen af PowerPoint-præsentationer til Captivate, kan makroen Capture Show splitte vis-animationen op på enkeltslides og derved bevare illusionen. Det er dog bedre at udvikle sit objekt direkte i Captivate og derved bevare fuld redigerbarhed.

- **Hav alt materialet klar inden sammenfletningen i Captivate**, da Captivate kun har stærkt begrænsede manipuleringsfunktioner af indsatte elementer, herunder bl.a. PowerPoint-præsentationer og billeder.

- **Overvej din undervisningsfilosofi**

Skal de studerende præsenteres for en eksplicit problemformulering før (eller først i) læringsobjektet, skal læringsobjektet fungere selvstændigt med indbygget læringsformål, eller skal det fungere som en supplerende læringsressource?

... og et trick

- **Preview i browser vha. F12** er yderst nyttig, da den indbyggede preview-funktion ikke afslører mulige problemer med fx browserintegration.

Tekniske perspektiver

Captivate er et eksempel på et program, som kan producere læringsobjekter, og hvor underviseren stadig har en forholdsvis stor fleksibilitet i udformningen af materialet. Programmer som fx HotPotatoes (Half-Baked Software 2005), Qarbon ViewletBuilder (Qarbon 2005), Reload Editor (Reload 2005), Easyquizz (Epistema 2005), diverse andre quizværktøjer og forskellige, indlejrede værktøjer i e-læringsplatforme m.fl. er godt nok lettilgængelige, men mindre fleksible og derfor knap så brugbare for den underviser, som selv vil have fuld kontrol over formen af sit undervisningsmateriale. Omvendt kan kun de færreste undervisere udvikle undervisningsmateriale i de programmer, som ellers ville give "ubegrænset" fleksibilitet, som fx Lingo i Director, ActionScript i Flash, avancerede JavaScript- eller Java-applikationer eller har adgang til en server, som kører CGI, PHP, ASP el.lign. Et realistisk alternativ er derfor de brugervenlige, grafiske programmer, som stadig giver øget kontrol og frihed som fx Captivate.

Andre brugervenlige programmer til udarbejdelse af læringsobjekter er bl.a. Windows Media Encoder, som kan anvendes til at streame forelæsninger til nettet, Microsoft Producer som bl.a. kan producere online PowerPoint-præsentationer i videoformat (jfr. Dørup 2004 for yderligere information), RapidBuilder til online simuleringer, tutorials og demonstrationer (XStream Software 2005), TutorPro til udvikling af forskelligartet e-læring (TutorPro 2005) og mange andre.

Mht. de tekniske aspekter af implementering af læringsobjekter har Flash-teknologiens gode integration i mange browsere og platforme minimeret de tekniske barrierer for udviklingen. Langt de fleste brugere kan umiddelbart udvikle Flash-læringsobjekter, så fokus er nu flyttet fra at få teknikken til at virke (som det ofte er tilfældet med fx ActiveX- eller Java-læringsobjekter), og om båndbredden er stor nok, til selve *indholdet* af objektet. Brugervenlige forfatterværktøjer til generering af Flash-objekter som fx Captivate giver undervisere mulighed for selv at udvikle læringsobjekter med hensyn til bl.a. kursets og egne præferencer.

Pædagogiske refleksioner

Et helt andet aspekt er naturligvis, hvordan læringsobjektet skal indgå i undervisningen. Som tidligere nævnt findes der forskellige pædagogiske filosofier om læring, som fordrer forskellig udformning og anvendelse af læringsobjekter. Hvad der er den "rigtige" strategi findes der intet entydigt svar på og afhænger da også af målgruppen, pensumet, de fysiske omstændigheder og forskellige kulturelle faktorer, så jeg vil blot konstatere, at forskellige læringsforhold fordrer forskellige læringsstrategier. Set fra en operationel synsvinkel og egne erfaringer har det vist sig nyttigt at anlægge en socialkonstruktivistisk ramme om fx blended learning. Den socialkonstruktivistiske ramme gør, vha. tanken om åbne læringsobjekter uden prædefineret læringsudbytte, det muligt for underviseren at pille enkeltdele ud af sit materiale til digitalisering, hvilket gør transformeringen fra den "klassiske" konfrontationsundervisning til blended learning mere overkommelig. Ligeledes kan ekspliciteringen af problemstillingen gøre det lettere at motivere de studerende til at anvende læringsobjekterne, da objekterne er let-tilgængelige læringsressourcer, som kan hjælpe til problemløsningen. I praksis kan undervisningen fx have en fragmenteret form, hvor indholdet er delt op i forskellige ressourcelæringsobjekter for hver metode og teori, sammenkædet af et narrativt læringsobjekt eller en forelæsning. Objekterne kan så være mere eller mindre lineære (og det kan være nødvendigt af forskellige praktiske årsager) – fx i form af bearbejdede PowerPoint-præsentationer suppleret med interaktive elementer i Captivate.

Mht. læringsobjektbegrebet ser jeg det som en fordel at have en bred forståelse heraf. Der er mange forskellige informationstyper, som kan give mening i en undervisningssammenhæng, så det ville være en skam at påkræve, at et læringsobjekt skulle have et indbygget læringsformål. Fx vil gennemgangen af Cognitive Walkthrough fint kunne suppleres med læringsobjekter indeholdende en linksamling med eksempler på "problematiske" websites, andre gennemgange af metoden, en artikel om exploratory learning (CW's bagvedliggende teori) samt andre artikler om emnet.

I forhold til implementering af e-læring og underviserens rolle i udarbejdelsen af elektronisk undervisningsmateriale, har der vist sig nogle nye muligheder med et program som fx Captivate. Hvis ikke den eksisterende e-læringsplatform har indbygget de fornødne værktøjer til at imødekomme underviserens ønske til udformningen af webbaserede læringsobjekter, er det måske slet ikke så underligt, at nogle undervisere føler, at de mister lidt af kontrollen over deres kursus' form og måske derfor stejler. Captivate kan være et bud på et simpelt program, som kan lade underviseren bevare en vis fleksibilitet og kontrol over formen på sin undervisning ved indførelsen af en e-læringsplatform, så den nye e-læringsplatformstilpassede undervisningsform ikke bare bliver dikteret af det aktuelle system. Implementeringen af e-læring bliver derved delvis decentraliseret og tilpasset det enkelte kursus frem for universitetet som helhed. Ydermere kan fleksibiliteten og underviserens aktive deltagelse i udviklingen måske også give en mere smertefri overgang ved skiftet fra eksisterende konfrontationsundervisning til blended learning.

Tak til...

Maria Hvid Stenalt, Jens Dørup og Christian Dalsgaard for løbende kommentarer og konstruktive forslag til artiklen.

Referencer

- Dalsgaard, Christian: Learning frameworks as an alternative to repositories. ICL2005, 2005a
- Dalsgaard, Christian: From Learning Objects to Learning Activities: References as an Alternative to Pedagogical Metadata. AACE E-Learn, 2005.
- Dørup, Jens: Windows Media Encoder og Producer: værktøjer til udvikling af netbaseret videomateriale. Tidsskrift for Universiteternes efter- og videreuddannelse. 1. årgang. nr. 3. Online: <http://www.unev.dk/files/temapublikation_3.pdf>, 2004.
- Engeström, Y.: Learning by Expanding: an Activity Theoretical Approach to Developmental Research. Helsinki: Orienta-Konsultit Oy, 1987.
- Epistema: Easyquizz. Online: <<http://www.epistema.com/en/>>, 2005.
- Half-Baked Software: Hot Potatoes. <<http://web.uvic.ca/hrd/halfbaked/>>, 2005.
- Hannafin, Michael J. & Land, Susan M: The foundations and assumptions of technology-enhanced student-centered learning environments. Instructional Science 25, 167-202, 1997.
- Jensen, Jens F.: Interaktivitet & Interaktive Medier. In *Multimedier, Hypermedier, Interaktive Medier*. Aalborg Universitetsforlag. Aalborg, 1998.
- Koper, Rob: Modeling units of a study from a pedagogical perspective. The pedagogical meta-model behind EML. First draft, version 2. Open University of the Netherlands. Online: <<http://eml.ou.nl/introduction/docs/ped-metamodel.pdf>>, 2001.
- Leontjev, A. N.: *Activity, Consciousness, and Personality*. Englewood Cliffs, N. J. Prentice-Hall, 1978.
- LTSC: Draft Standard for Learning Object Metadata. Online: <http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf>, 2002.
- Macromedia Captivate: Online: <<http://www.macromedia.com/software/captivate/>>, 2005.
- Pillai, Shyam: Capture Show. Online: <<http://skp.mvps.org/cshow.htm>>, 2004.
- Qarbon ViewletBuilder: Online: <<http://www.qarbon.com/>>, 2005.
- Reload Editor: Online: <<http://www.reload.ac.uk/editor.html>>, 2005.
- TheCounter: Online: <<http://www.thecounter.com/stats/2005/September/res.php>>, 2005.
- TutorPro: Online: <<http://www.tutorpro.com/products/Author.asp>>, 2005.
- Vygotsky, L. S.: *Mind in Society*. Harvard University Press, 1978.
- Weller, M. J., Pegler, C. A. and Mason, R. D.: Working with learning objects - some pedagogical suggestions. Sheffield University, 8th-10th September 2003. Online: <<http://iet.open.ac.uk/pp/m.j.weller/pub/altc.doc>>, 2003.
- XStream Software: RapidBuilder. Online: <<http://www.xstreamsoftware.com/>>, 2005.

Bilag 1: Installering og anvendelse af Capture Show

Download og installering af Capture Show

Denne vejledning forudsætter Windows XP samt en nyere version af PowerPoint.

1. Download Capture Show fra <http://skp.mvps.org/cshow.htm>.
2. Unzip det downloadede filarkiv ved at højreklikke på cshow.zip og vælge Extract All.
3. Åbn PowerPoint.
4. Indstil macrosikkerheden til "medium" i menuen Tools > Macro > Security.
5. Tilføj Capture Show ved at vælge Tools > Add-ins > Add new og vælge Capture.ppa og klikke OK.

Anvendelse af Capture Show

Denne vejledning tager udgangspunkt i en færdig PowerPoint-præsentation indeholdende vis-animationer.

1. Åbn præsentationen i PowerPoint.
2. Tilpas præsentationen til skærmopløsningen for at undgå sorte kanter. Dette gøres ved at vælge fx 800x600 eller 640x480 i feltet Performance i menuen Slide Show > Set Up Show.
3. Vælg Tools > Capture Show.
4. Vælg fx Automatic og As slides in a new presentation og klik Begin (hvorefter præsentationen startes og "indfanges").
5. Resultatet bliver en ny PowerPoint-præsentation, hvor animationerne er delt ud på en række slides. Gem denne præsentation, da den efterfølgende skal importeres i Captivate.