

Video mediering i programmeringsundervisningen

- brug af videoer til at udfolde programmeringsprocessen

Jens Bennedsen

cand. scient, projektleder

it-vest

jbb@it-vest.dk

<http://www.daimi.au.dk/~jbb>



Jens Bennedsen er til daglig projektleder på it-vest, hvor han leder tvær-institutionelle uddannelsesudviklings projekter med fokus på fleksibel, netstøttet læring. Han har tidligere været ansat på Datalogisk Institut, Aarhus Universitet. Hans forskningsområde er objekt orienteret programmeringsundervisning og -læring

Denne artikel belyser hvorledes video mediering har givet en værdifuld ekstra dimension i undervisningen i introducerende programmeringsfærdigheder. I forbindelse med programmering er et af de vigtigste mål at lære de studerende en systematisk tilgang til udviklingen af programmer. Det vil sige at understøtte de studerendes læring af programmeringsprocessen. Traditionelle medier som lærerbøger er meget lidt velegnede til at illustrere denne proces; de fokuserer på produkterne (de færdige programmer) i stedet for at fokusere på produktionsprocessen. De videomedierede materialer er i denne sammenhæng optagelser af hvorledes underviseren løser praktiske programmeringsopgaver. Disse videoer kommunikerer vigtige pointer ved programmeringsprocessen såsom at der ikke er en lige vej til det færdige produkt, at man anvender forskellige værktøjer, at man ændrer sit program, at man tester sit program undervejs mm.

Jeg har identificeret fem forskellige kategorier af videomedierede undervisningsmaterialer og evalueret hvorledes de studerende har anvendt dem i et konkret kursusforløb. Evalueringen viser dels at de studerende anvender de fleste typer meget og finder dem meget anvendelige dels at videoer der blot dokumenterer forelæsninger ikke er ret benyttede.

Introduktion

Det er min opfattelse at det vigtigste faglige mål i en introducerende programmeringsundervisning er at lære de studerende en systematisk tilgang til udviklingen af programmer. Programmeringsprocessen er en meget vigtig del af denne systematiske tilgang, og jeg har erfaret at videoer i form af skærm optagelser af programmerings sessioner er en billig, effektiv og simpel måde at formidle processen. Dette kan opfattes som en udvidelse af ”the applied apprenticeship approach” som argumenteret i (Astrachan 1995, Linn & Clancey 1992).

En præsentation og udfoldning af programmeringsprocessen er vigtig for studerende der tager et introducerende kursus, men traditionelle statiske undervisningsmaterialer som lærerbøger, artikler, præsentationer på slides mm er utilstrækkelig til det formål. De er gode til fremvisning af et resultat af en proces (et produkt), men ikke til præsentationen af processen i sig selv. Udover at være utilstrækkelige til præsentationen af en programmeringsproces har an-

vendelsen af traditionelle materialer et andet problem: de er typisk fokuseret på præsentationen af en ideel løsning. Denne ideelle løsning er typisk resultatet af en ikke lineær udviklingsproces. Som andre (Soloway 1986, Spohrer & Soloway 1986, Spohrer & Soloway 1998b), anser jeg præsentationen af produktet uafhængigt af udviklingsprocessen som problematisk; det vil utvivlsomt efterlade de studerende med det fejlagtige indtryk at der er en lige, ”gylden” vej fra problem til løsning. Dette er meget langt fra sandheden. Begyndere, som ser deres underviser præsentere nydelige og simple løsninger forventer ofte at de kan udvikle løsninger på den samme direkte måde. Når de opdager at det ikke er tilfældet, føler de sig inkompetente og i det værste tilfælde mister de motivationen for at lære at programmere. Udover at undervise de studerende i de værktøjer og teknikker, der bruges i udviklingen af programmer (programmeringssproget, programmeringsteknikker som at opdele problemer i delproblemer, programmeringsomgivelsen etc.), må vi derfor også undervise dem i udviklingsprocessen: Konkret brug af disse værktøjer og teknikker på en systematisk, inkrementel og typisk ikke lineær måde til at skabe en god løsning på det foreliggende problem. En vigtig del af dette er at demonstrere

- at en løsning udvikles i mange små skridt i stedet for et stort
- at resultatet af hvert lille skridt skal testes, at tidligere beslutninger skal gøres om
- at løsningen skal ændres, at det er naturligt at lave fejl, at fejlmeddelelser fra programmeringsværktøjet kan være vildledende,
- at dokumentationen indeholder hjælp
- hvordan dokumentationen bruges
- uanset at det er en ikke lineær vej, så er der en systematisk måde at komme fra problem til løsning.

Vi kan ikke udelukkende basere os på at de studerende lærer dette selv, og her har videoer vist sig at være en god måde at udfolde denne proces på.

Jeg er desuden overbevist om at videoer også kan bruges i andre sammenhænge, hvor formidling af arbejdsprocesser er vigtig. Et eksempel kunne være hvordan man udarbejder en videnskabelig artikel.

Brug af videoer i undervisning

Gennem de sidste år er streaming video blevet mere populær og almindelig (Ma et al 1996, Smidth et al 1999). Kompressionsteknikker er blevet standardiseret og forbedret, båndbredden er væsentligt forøget i forhold til blot få år siden (også i private hjem), så det er nu muligt at anvende video uden de store problemer i undervisningen. Derudover har fremkomsten af PC video optage- og redigeringsværktøjer gjort det let for folk selv at lave deres egne videoer. De fleste eksempler på brug af videoer i undervisningssammenhæng er såkaldte webcasts – det vil sige optagelser af klasseundervisning og forelæsninger. Denne fremgangsmåde er brugt af mange universiteter inklusiv flere fremtrædende amerikanske universiteter som Berkeley og MIT (Berkley 2004, MIT 2004). Selvom sådanne videoer kan have værdi for studerende der af den ene eller anden grund ikke har mulighed for at deltage i undervisningen eller gerne vil have (dele af) forelæsningen gentaget, så tilføjer de ikke noget signifikant nyt til selve undervisningen.

Siddiqui & Zubaira (2000) argumenterer at “*the only change over time is the way we store information – writing on stones, leaves, and cloth was changed to using slates, blackboards and now electronic media*”. Jeg er meget uenig i dette udsagn. Brugen af ny teknologi (inklusive video) giver en bred vifte af nye muligheder for såvel undervisere som studerende. Med ny teknologi, her computere og værktøjer til video-optagelse, bliver det muligt at gemme information der beskriver og repræsenterer dynamisk opførsel – noget der er næsten umuligt ved brug af traditionelle medier som sten, tavler og bøger.

Udfoldning af programmeringsprocessen

Ideen om at udfolde programmeringsprocessen er ikke ny:

“Anyone with a reasonable intelligence and some grasp of basic logical and mathematical concepts can learn to program; what is required is a way to demystify the programming process and help students to understand it, analyse their work, and most importantly gain the confidence in themselves that will allow them to learn the skills they need to become proficient.”

Dette citat er femten år gammelt (Gantenbein 1989), men på trods af dette har programmeringsprocessen stadig ikke fundet vej til undervisningsbøgerne om introducerende programmering.

Bøger adresserer ikke problemet

På en nylig workshop (Kölling 2003), blev en undersøgelse af 39 af de bedst sælgende lærebøger om introducerende programmering præsenteret. Den overordnede konklusion var at alle tekstbøgerne var struktureret efter begreberne i programmeringssproget – *ikke* efter de programmeringsteknikker vi ønsker at lære vores studerende. Progressionen er fra detalje mod helet (bottom-up): fundamentale emner (variabler, typer, objekter og klasser, etc.), kontrolstrukturer (sekvens, selektion og iteration), metoder (parametre, rekursion), klasser og subklasser etc. Rækkefølgen kan adskille sig en smule, men den overordnede struktur er den samme. Sagt med andre ord: At lære programmeringssprogets regler har højere prioritet end at lære programmeringsteknikker og programmeringsprocessen. Denne almindelige fremgangsmåde vil hjælpe de studerende med at forstå programmeringssproget og strukturen af simple programmer, der bliver præsenteret i teksten, men det udfolder ikke programmeringsprocessen.

Når vi ved hvilket emne, der skal undervises i, kan man spørge hvorfor dette emne ikke har fundet vej til lærebøgerne? Det bedste svar jeg kan give er, at lærebogsmediet ikke er egnet til denne form for beskrivelser.

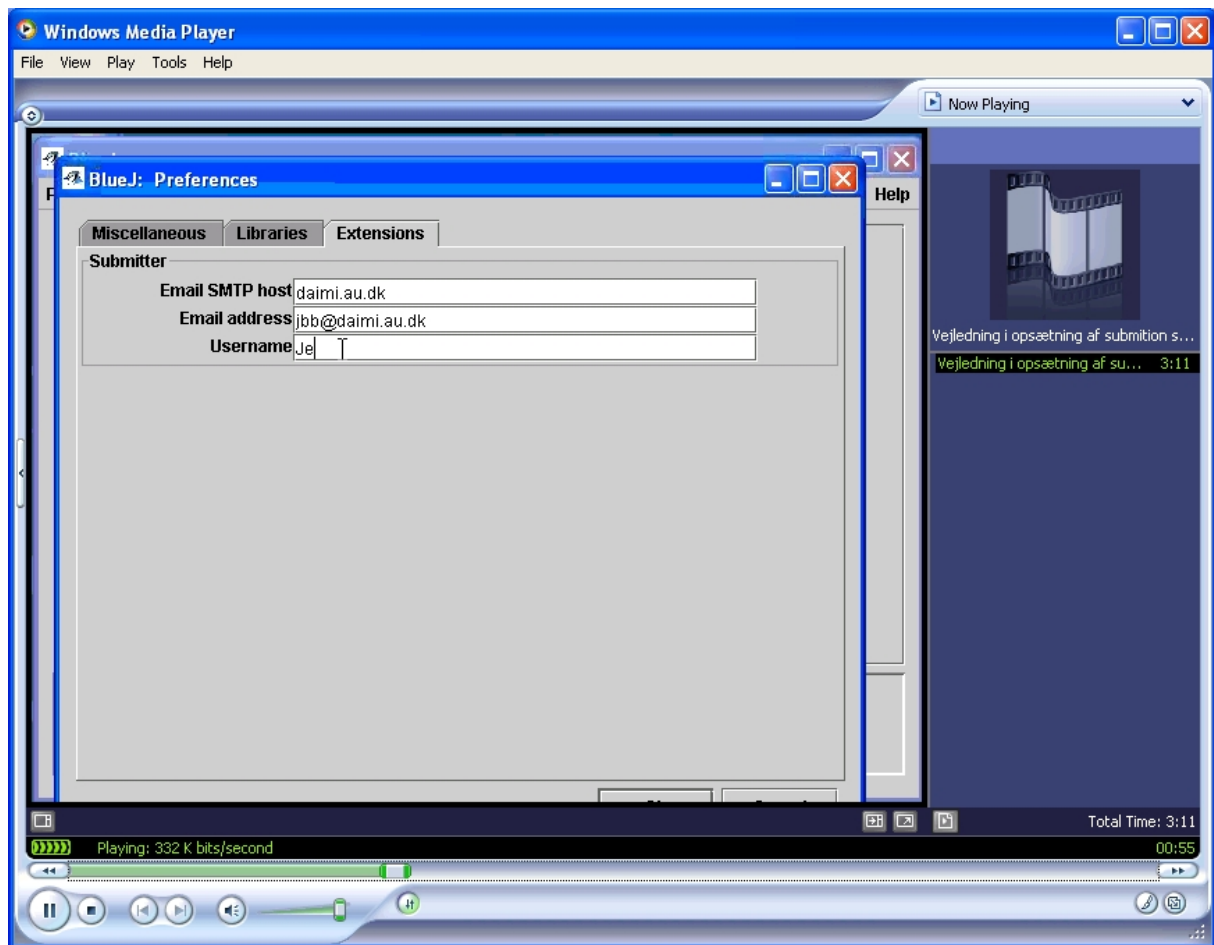
Elementer af programmeringsprocessen

Programmeringsprocessen er blandt andet karakteriseret ved følgende elementer: Brug af et konkret programmeringsværktøj, fejlhåndtering, brug af dokumentation og systematisk konstruktion af et program ud fra en abstrakt model af programmet. Disse elementer er ikke særligt velegnede til at blive præsenteret i et tekstbaseret materiale, men de er vigtige mål i undervisningen. For hvert af elementerne vil jeg her diskutere hvorledes det indgår i et introducerende kursus og hvordan videoer kan understøtte elementet.

Brug af et konkret programmeringsværktøj.

Programmeringsværktøjer er applikationer med forskellige funktionaliteter. Hovedformålet er at de kan anvendes til at konstruere og i de fleste tilfælde oversætte et program skrevet i et konkret programmeringssprog så det kan afvikles på en PC. Som alle andre applikationer er det hensigtsmæssigt med en introduktion til de forskellige faciliteter – hvad de kan bruges til og hvordan de bruges. Disse funktioner og faciliteter kan vises ved en video, hvor indholdet er optagelser af brugen af faciliteterne i værktøjet i et konkret programmeringsforløb. Ved en efterfølgende evaluering af et konkret kursus var de studerende begejstrede for denne formidlingsform i stedet for den traditionelle form med tekst og skærbilleder. Desuden er disse optagelser hurtige at lave og min egen oplevelse er, at de er væsentligt hurtigere at udarbejde end tekstbaserede beskrivelser. Et konkret eksempel var opsætningen af en afleveringsfunktion i programmeringsværktøjet. Her skulle den studerende angive en række tekniske ting. I

stedet for at tage en række skærbilleder af de forskellige vinduer, optag jeg en kort video af min egen opsætning. Indholdet af videoen var en optagelse af min skærm hvor jeg fortalte hvad jeg gjorde og hvorfor. Et eksempel kan ses i (Bennedsen 2004a). Yderligere information kan læses i (Dørup 2004)



Figur 1: Eksempel på video der illustrerer en bestemt funktionalitet i programmeringsværktøjet.

Fejlhåndtering

For at få de studerende til at føle sig mere trygge er det vigtigt at vise dem at enhver software udvikler begår fejl, og fejlhåndtering er en naturlig del af programmeringsprocessen. Det er desuden vigtigt at vise de studerende hvordan fejl håndteres. Specielt er det vigtigt at vise, at fejlmeddelelserne fra programmeringsværktøjet ikke altid indikerer den reelle fejl og at der er forskellige typer af fejl: syntaktiske fejl, semantiske fejl og logiske fejl. Videooptagelser af konkrete programmeringsforløb (igen i form af optagelser af hvad der foregår på skærmen og underviserens kommentarer dertil) viser de studerende hvorledes en erfaren programmør arbejder og giver dem derved tryghed (Fjuk et al 2004).

Dokumentation

Moderne programmeringssprog er ikke kun selve sproget, men i lige så høj grad hjælpekomponenter, som de studerende også bliver nødt til at kunne anvende. Dokumentationen til for eksempel programmeringssproget Java er tilgængelig online og de studerende skal kunne anvende dokumentationen for at konstruere programmer. Dette er en del af en programmeringsproces – også her er videoer om konkret problemløsning i programmering en værdifuld ressource, der illustrerer hvorledes dokumentationen anvendes.

Systematisk konstruktion af programmer

Studerende forsøger ofte at skabe en komplet løsning før de starter med at teste den. Det er ikke den opførsel vi ønsker at fremme hos vores studerende – vi ønsker at de studerende skaber en løsning på en inkrementel måde, hvor de tager små skridt med alternation mellem programmering og testning. Udvikling på denne måde gør det meget lettere at finde og rette fejl og derved lettes den samlede programmeringsproces. Emnet er ikke let at kommunikere i en traditionel lærebog; hvor starter man?, hvordan anvender man programmeringsværktøjet for inkrementelt at udvikle en løsning?, hvordan tester man sin løsning? mm. En lærebog er brugbar til at beskrive principper og teknikker for testning (black-box, white-box, testcases mm), men hvordan denne aktivitet integreres i den samlede udviklingsproces, bliver langt bedre demonstreret ved at anvende live programmering eller videoer. Med en videooptagelse af programmeringsprocessen understøtter man således procesaspektet af programmeringen. Et eksempel kan ses i (Bennedsen 2004b)

Når en studerende læser en lærebog får han let den opfattelse at programmører aldrig begår fejl. De udarbejder altid perfekt virkende løsninger i første forsøg, og de skal aldrig rette og forbedre deres program. Dette er naturligvis forkert. Alligevel er det den opfattelse, de studerende sidder tilbage med efter at have læst en lærebog. I (Fowler 1999) fremføres det, at en dygtig og erfaren programmør skal forvente at bruge ca. 50% af sin tid på at rette og forbedre sit program. Hvis dette er hvad en erfaren programmør skal forvente, vil en studerende på et introducerende kursus skulle forvente at bruge mere, og kan derfor ikke forvente at lave en perfekt løsning i første forsøg. Men studerende har en opfattelse af de skal kunne – og hvordan skal de også forvente anderledes? Vi har fundet det svært at motivere behovet for at forbedre sit program overfor de studerende. Målet med at forbedre sit program er at gøre det lettere at vedligeholde og forstå. Studerende ved typisk ikke hvornår det er nødvendigt at forbedre deres program, de synes at løsningen er tilfredsstillende når programmet kan oversættes og ”overleve” nogle simple tests. Med ved at vise forbedringerne ”live” giver det en meget bedre forståelse for teknikker og værktøjer til forbedring og viser nødvendigheden af forbedringen. For yderligere at forbedre motivationen har vi i visse tilfælde startet med et program fra de studerende og herefter vist hvorledes teknikker til forbedring af programmet kan gøre det simple og mere forståeligt.

Brug af forskellige typer af videoer

I dette afsnit vil jeg beskrive hvorledes video medierede materialer har været brugt i et introducerende programmeringskursus, der har været afviklet som et ”blended learning” forløb. For en uddybende beskrivelse se (Bennedsen & Caspersen 2003, Fennefoss & Bennedsen 2003)

Video kategorier

I kurset har vi optaget følgende fem forskellige typer af videoer:

- 1. Introduktion til opgaver:* Mange studerende slås med hvordan man for begyndt på en opgave: hvad er problemet, hvordan skal jeg starte, hvor skal jeg præcist gøre? Mange af disse spørgsmål kan adresseres i en video hvor også fragmenter af en løsning kan præsenteres.
- 2. Løsninger på opgaver:* Fremvisning af processen til at nå til en løsning på en opgave. Her er fokus på både løsningen (produktet) og processen.
- 3. Dokumentation af webmedierede undervisningsaktiviteter:* Ved at optage live programmeringen som den foregår, har de studerende mulighed for at gense (dele af) processen på et senere tidspunkt.

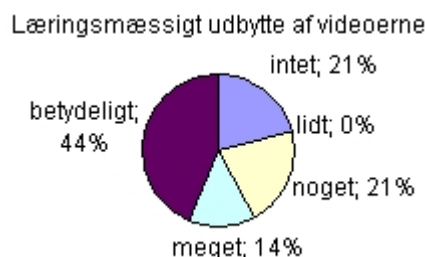
4. *Alternative undervisningsmaterialer*: For de centrale begreber i kurset har vi lavet små programmeringsproblemer, der illustrerer anvendelsen af begrebet. Dette giver en diversitet i kursusmaterialet, så forskellige læringsstile kan understøttes.

5. *Værktøjshjælp*: Vi har produceret videoer til forskellige funktioner i de værktøjer vi anvender i kurset. Ligesom Alford (2003) har vi fundet, at det er meget lettere at lave videoer, der beskriver hvorledes man *gør* tingene end at lave skrevne beskrivelser.

Feedback fra de studerende

I 2003 har en kollega og jeg afholdt et introducerende programmeringskursus med hhv. 35 og 20 studerende, hvor vi anvendte de omtalte videotyper. Jeg har evalueret brugen af videoer i kurset, både via et omfattende spørgeskema og en række interviews af studerende. Spørgeskemaet viser at mere end 2/3 af de studerende har set mere end 50 % af samtlige videoer. Alle videoer er gemt på en almindelig webserver, og gennem den første halvdel af kurset var der totalt 9800 hits på kursushjemmesiden (dette inkluderer webcrawlers,). Af disse hits var 1650 hits på videoerne. Mange af de studerende downloader videoerne en gang for alle i stedet for gentagne gange af hente dem fra vores server, så anvendelsen kan være større. Distributionen af hits på de forskellige kategorier var: Introduktion til opgaver 28 %, løsninger på opgaver 19 %, dokumentation af webmedierede undervisningsaktiviteter 9 %, alternative undervisningsmaterialer 21 % og værktøjshjælp 23 %. Det interessante er, at kun 9 % af hittene var på optagelser af webmedierede undervisningsaktiviteter. Dette indikerer at muligheden for at gense undervisningslektioner, den mest almindelige brug af videoer på nettet, er den mindst brugbare af de fem kategorier.

De studerende har vurderet det læringsmæssige udbytte af videoerne, resultatet kan ses i figur 1



Figur 1. Læringsmæssigeudbyttet af videoerne

58% har indikeret et højt eller meget højt udbytte. I et interview efter et af kurserne har en studerende karakteriseret brugen af videoer som følger:

Jeg vil påstå at læringspotentialer er bedre ved denne undervisningsform end ved traditionel undervisning; i det virtuelle klasseværelse kan jeg eliminere alle former for støj og afbrydelser. Kombineret med muligheden til at gense aktiviteten er udbyttet optimalt.

Video Produktion

Fremstillingen af skærmoptagelser vha. video er billig og kan gøres næsten hvor som helst. Til optagelse og udsending af webmedierede undervisningsaktiviteter, har vi brugt Microsoft Media Encoder, som er gratis

(<http://www.microsoft.com/windows/windowsmedia/download>). Efterproduktionen har været begrænset til at klippe begyndelsen eller slutningen af en optagelse væk, noget vi kunne gøre med Windows Media Fileeditor – også et gratis produkt, der følger med ved download af Windows Media Encoder. Specielt videoerne der dokumenterede webmedierede undervisningsaktiviteter har været omkostningsfrie – vi optog skærm og lyd under aktiviteten ved

hjælp af Windows Media Encoders indbyggede screen capture funktion og gemte det på en webserver efterfølgende.

For at gøre det muligt at finde forskellige dele, er der lavet et simpelt emne→tid indeks for alle videoerne. Disse indexer er blevet gemt i en database således at de studerende kan søge efter materialet på et senere tidspunkt. Tilføjelsen af emne→tid indekset har gjort at undervisningsaktiviteterne får et udvidet brug fra noget, der kun har værdi når deltagerne er tilstede, til en del af deres repertoire af undervisningsmaterialer.

Videoerne med det alternative undervisningsmateriale er lidt anderledes. Her har jeg foretaget en smule forarbejde hovedsageligt med at finde et godt eksempel til at illustrere emnet. Det er min erfaring, at en meget detaljeret drejebog er overflødig, det kan gøre videoerne mindre autentiske og til og med mere kedelige. Jeg har produceret ca. 30 af denne type af videoer og det er min erfaring at jeg bruger ca. en time i forberedelse og 20 minutter i efterbehandling af en 30 minutters video. (se : <http://www.daimi.au.dk/~ioop/video>)

Relateret arbejde

Brugen af videoer i undervisning er ikke ny (Smidth et al 1999). Videoer er for eksempel brugt ekstensivt i (Gries & Gries 2001), men deres brug er noget anderledes end vores anvendelse:

- Alle videoer er meget korte og fokuserer på at vise et enkelt aspekt ved programmerings-sproget eller programmeringsprocessen.
- Videoerne er “perfekte” – de viser ikke at det er normalt at lave fejl (og hvordan de kan rettes)
- Videoerne viser ikke et integreret brug af forskellige værktøjer som tegneværktøjer, programmeringsværktøj og dokumentation.

Ifølge kategoriseringen af videoer beskrevet ovenfor, kan videoerne i (Gries & Gries 2001) beskrives som alternative undervisningsmaterialer.

Andre bruger en meget rigere form for multimedia, hvor der bliver lavet små simuleringer, tegnefilm, lagt baggrunds lyd mm.. Et eksempel indenfor programmeringsverdenen er læringsobjekterne diskuteret i (Boyle 2003). Disse læringsobjekter fokuserer på at give en multimedia fremstilling af basale begreber indenfor programmering som for eksempel iteration. Igen er det ikke programmeringsprocessen der er i fokus men illustration af forskellige begreber. De samme forskelle der er nævnt ovenfor gør sig derfor også gældende her og disse læringsobjekter er meget dyre at udvikle.

Konklusion

Udfoldelse af programmeringsprocessen er et vigtigt mål i et introducerende programmerings kursus, der ikke kan formidles vha. traditionelle undervisningsmaterialer som lærebøger, undervisningsnoter, tavler, overhead præsentationer etc. Hovedargumentet for dette er, at disse medietyper ikke egner sig til denne form for formidling.

På baggrund af vores erfaring på IT-vest foreslår jeg brugen af videoer i form af skærmoptagelser, hvor underviseren fortæller hvad og hvorfor han gør det han gør, som en simpel, effektiv og billig måde at foretage denne udfoldning. video mediering Fra evalueringen af de to kurser ved jeg, at de studerende bruger og værdsætter videoerne, nogle studerende finder dem endda overlegne i forhold til traditionel undervisning. Produktionen af video medierede materialer har vist sig at være let og billig i modsætning til andre mere multimediale læringsobjekter.

Fremkomsten af ny teknologi til digital videofremstilling har gjort det muligt let at udfolde elementer i en programmeringsproces, der tidligere har været implicite. De studerende hilser det nye materiale velkomment og jeg oplever at det har en stor indflydelse på deres praktiske

programmeringsfærdigheder. Dette afspejles blandt andet i beståelsesprocenter på kurset – de to gange kurset har været afholdt er der tilsammen startet 54 og der er 42 der har bestået eksamen. De studerende er master studerende og langt hovedparten af dem er studerende uden nogen form for programmeringsmæssig baggrund.

Litteratur

- Alford, K. *Video FAQs – Instruction-On-Demand*. Proceedings of Frontiers in Education, Boulder Colorado, 2003.
- Astrachan, O. & D. Reed. *AAA and CSI: The Applied Apprenticeship Approach to CSI*. Proceedings of the twenty-sixth SIGCSE Technical Symposium on Computer Science Education, Nashville, Tennessee, 1995. pp. 1-5.
- Bennedsen, Jens og M. Caspersen. *Rationale for the Design of a Web-based Programming Course for Adults*. Proceedings of ICOOL 2003, International Conference on Open and Online Learning, Mauritius, 2003.
- Bennedsen, Jens. *Afleveringssystemet i BlueJ* tilgængelig på <http://www.daimi.au.dk/~ioop/video/submit-vejledning.wmv> . Sidst tilgået 17marts 2004.
- Bennedsen, Jens. *Eksempel til kapitel 2 i Barnes og Kölling*. <http://www.daimi.au.dk/~ioop/video/kapitel2-2.wmv>. Sidst tilgået 17marts 2004.
- Berkeley, <http://webcast.berkeley.edu/courses/>, 29. februar 2004
- Boyle, T. *Design principles for authoring dynamic, reusable learning objects*. Australian Journal of Educational Technology, 19 (1), 2003, pp. 46-58.
- Dørup, J. *Windows Media Encoder og Producer: en praktisk tilgang til brugen i universitetsundervisningen*, UNEV 2004, <http://???>
- Fennefoss, T og J. Bennedsen. *Fleksibel webundervisning vokser med opgaverne*, UNEV 2003, http://www.unev.dk/files/turid_fennefoss_jens_bennedsen.pdf, 29. februar 2004.
- Fjuk, A. et al. *Learning Object-Oriented through ICT-mediated Apprenticeship*. Submitted til ICALT 2004, 4th International Conference on Advanced Learning Technologies, Joensuu, Finland
- Fowler, M. *Refactoring – Improving the Design of Existing Code*. Addison-Wesley, 1999. ISBN 0-201-48567-2
- Gantenbein, R. E. *Programming as Process: A 'Novell' Approach to Teaching Programming*. Proceedings of the twentieth SIGCSE Technical Symposium on Computer Science Education, Louisville, Kentucky, 1989, pp. 22-26.
- Gries, D. & P. Gries. *ProgramLive*. John Wiley & Sons, 2001.
- Kölling, M. *The Curse of Hello World*, Invited lecture at Workshop on Learning and Teaching Object-orientation – Scandinavian Perspectives, Oslo, October 2003.
- Linn, M. C. & M. J. Clancy *The Case for Case Studies of Programming Problems*. Communications of the ACM, 35 (3), 1992, pp. 121-132.
- Ma, W. et al. *Video-based Hypermedia for Education-On-Demand*. Proceedings of the fourth ACM International Conference on Multimedia, 1996, pp. 449-450.
- MIT www.swiss.ai.mit.edu/classes/6.001/abelson-sussman-lectures/. 29. februar 2004
- Siddiqui, K. J. & J. A. Zubairi. *Distance Learning using Web-based Multimedia Environment*. Academia/Industry Working Conference on Research Challenges (AIWORC'00), 2000 pp. 325, IEEE.
- Smidth, T, A. Ruocco & B. Jansen. *Digital Video in Education*. Proceedings of the thirtieth SIGCSE Technical Symposium on Computer Science Education, New Orleans, Louisiana, 1999, pp. 122-126.
- Soloway, E. *Learning to Program = Learning to Construct Mechanisms and Explanations*. Communications of the ACM, 29 (9), 1986, pp. 850-858.

Spohrer, J. & E. Soloway. *Novice Mistakes: Are the Folk Wisdoms Correct?*. Communications of the ACM, 29 (7), 1986, pp. 624-632.

Spohrer, J. & E. Soloway. *Analyzing the High-Frequency Bugs in Novice Programs*, I Iyengar, S. & E. Soloway, (Eds.). *Empirical Studies of Programmers*. Ablex, New York, 1986.