A CONSTRUCTIONIST APPROACH TO THE DESIGN OF LEARNING ENVIRONMENTS

Learning Sciences 426 (cross-listed with Computer Science)
Spring 2005
Wednesday, 1:30-4:30 PM
Annenberg 303, North Learning Studio

Professor: Uri Wilensky
Address: Annenberg 311
Phone: 847-467-3818
E-mail: uri@northwestern.edu
TAs: Paulo Blikstein (467-3859 paulo@northwestern.edu) &
Josh Unterman (467-2816 junterman@ccl.northwestern.edu)

Course Web Site: http://ccl.northwestern.edu/courses/cd2005/
Course Instructors: cd-fac@ccl.northwestern.edu
Course Members (students and faculty): cd@ccl.northwestern.edu

Course Description

This course is a hands-on practicum in designing and building technology-enabled curricula and/or educational software. We will use many rich software toolkits designed to enable novice programmers to get their "hands dirty" doing iterative software design. In addition to the hands-on component, the course is also designed to introduce you to the Constructionist Learning design perspective. This perspective, first named by Seymour Papert and greatly influenced by the work of Jean Piaget, is very influential in the learning sciences today. The Constructionist approach starts with the assumption that teaching cannot successfully proceed by simply transferring knowledge to students' heads. Skillful teaching starts with the current state of knowledge of the student. In order for students to learn effectively, they need to construct the knowledge structures for themselves. In this class, we will engage in the construction of artifacts and, through such constructions, explore and evaluate the design of construction kits and tools to enable learners to construct motivating and powerful artifacts. In the spirit of Constructionism, students in this course will self-construct their own understanding of the educational software and of the literature through constructing artifacts (both physical and virtual) and engaging in reflective discussion of both the artifacts and the tools used to construct them.

Students will put all of this together in a substantial final project in which they design and implement a constructionist learning environment.

After completing this course, you should be able to:

- Design and implement educational software that is at least "Alpha" ready for use
- Design technology-enabled activities that take advantage of the computational medium
- Exercise good judgment in such design within the context, domain and deployment situation.
- Avoid common educational software design errors
- Assess programming/authoring/scripting technologies as to appropriateness for educational needs

- Evaluate and utilize educational claims of software authors and promoters
- Understand the Constructionist design perspective and use it to author and assess software tools and learning environments

This class will emphasize computer programming projects using Logo-like languages. Logo is a computer programming language designed explicitly for use by children and is in use in large numbers of schools, from elementary on up. No previous programming background is assumed, but you will be expected to devote substantial time to programming. This can and will be frustrating to many of you, at least initially. I do not mean to discourage you – I am confident all of you can master the programming aspect of the course, as have many students in the past, who had had no prior programming experience. It is my belief that even if you do not intend to be a software designer yourself, it is the reality of today – and more so, of tomorrow – that should inform your choice to become at least somewhat familiar with the promise of technology in education. You are strongly encouraged to get help from your fellow students through the class email list as well as from the TAs. The TAs will hold weekly office hours designed especially for technical and programming support. We will attempt to schedule these office hours flexibly, and per special requests.
In addition to projects, there will be weekly readings. Typically, one paper or two short papers per week. There is a considerable literature that we will not have time to read this term. I have provided a more extensive bibliography at the end of the syllabus. You may find some of these readings to be useful to you in completing the final project.

Software packages we will use

We will use quite a number of learning software packages in this course. The 3 packages we will use the most are all based on the computer language Logo.
They are:
Microworlds Logo – a multi-media version of basic Logo in common use in elementary schools worldwide. It also includes music, graphics, video and web tools.
NetLogo – a multi-agent version of Logo, this language is tuned for constructing models of complex dynamic systems. It is useful for creating models of ecological systems, chemical systems, economic trade, social behavior, ….
LEGO Mindstorms - a robot control language that enables learners to design behaviors for Lego robots We will construct Lego robots that have sensors and motors and can interact with objects in the world.

Besides these 3 basic packages, in the software review section of the class, we will also explore a number of other packages. Software we might look at includes: TableTop, Genscope, Biologica, Zoombinis, SimCalc, HubNet, ChemSense, Fathom, MediaMoo, Moose Crossing, CSILE, Hypergami, Stagecast Creator, Vehicles, RelLab, Interactive Physics, the Sims, Impromptu, Geometer's sketchpad, Matlab, Squeak, Boxer, Model-It, STELLA,…

Summary of Requirements

This course is designed to be somewhere between a class and a working group. I'm hoping that we'll work together to make sense of readings, and, for most of the class projects, you will be working in small groups.

So the requirements for everyone are:

1. Keep up with the readings and participate in class, both in person and virtually. You will be expected to post a comment on the week's reading each week by the day before class.
2. Complete and present several (mostly group) programming assignments using Logo, LEGO Mindstorms and NetLogo.
3. Review one educational software package and present your review in class.
4. For one week, to have the class get a little more breadth than the course has time for, we will ask each of you to pick and read something from the reading list that we didn't get to as a class and write a summary on the course discussion board.
5. Design and implement your final project
6. Give a presentation during the last week of the course.

In addition, due to the group project nature of the class, you are also asked to send email to cd-fac (as soon as you know) if you cannot make a particular class meeting. You are also responsible for communicating with your project-mates and letting them know in advance if there is any problem with your part of the project.

About the Final Project

The final project is to design and implement a constructionist learning environment. There are two* basic alternatives for this project:

1) Standalone Educational Software (scaffolding in software)

Design and implement some constructionist educational software. This option would involve writing a design specification for the software that describes what the software is for, who it serves, why it is needed, why it is best done in software, etc. Subsequent to receiving feedback on the design specification you will need to start working on a functional specification of the software itself and then embark on implementing it. You are free to use any authoring tools you like to implement the software as long as you make a good argument for their being well matched to the task. Suggested educational software genres are: a simulation game, a microworld, a collaborative role-play or MUD (a collaborative virtual space or Multi-User Dimension).

2) Software-embedded Curriculum (scaffolding in curricular materials)

Design and implement an educational activity that has a computationally embedded component. In this option, you are asked to use one of the three main software environments used in this course: Microworlds Logo, LEGO-Mindstorms or NetLogo. As above, you would begin with a design specification. Depending on the design, you may or may not require a functional specification – it could be a curriculum flow specification instead. You would then go on to construct the software and/or Lego constructions that form the kernel of the activity, flesh out the curricular materials that accompany the software and write up a paper that describes one person's (could be yourself) path through the activity.

The final project design specification is due by May 11.
The final project software specification (or curricular flow spec) is due by June 1.
The final project is due by June 3.
Final projects will be presented on June 6. You are welcome to invite friends and/or relatives to attend.
The Final projects may be conducted individually or in pairs.

* For some students, the final project could take a different direction, such as designing a (computational) research model of organizational change using NetLogo. If you're interested in this option, come and talk to me.

Grading

All assignments and projects will be graded as either complete or incomplete. If a project is judged incomplete, you will have an opportunity to complete it or redo it the following week. If you cannot complete the final project by June 3, you may take an incomplete for the course. No penalty will be assessed for late final projects – they can be handed in as late as the summer and your incomplete grade will be changed at that time, but you must make a coherent presentation on June 6. You will also be assessed on your class participation both in class and virtually.

Readings

A course reader is available from Andrea Zuckert in the LS office.
You also need to purchase a book at Norris:
Papert, S. (1980). Mindstorms. New York: Basic Books.

WEEKLY SCHEDULE

Readings are listed on the week they are due. They will be discussed during class that week. Reactions are to be sent to the CD email list <u>by Tuesday morning</u>.

<u>Week 1, March 30</u>
Please read the entire book Mindstorms as this week's assignment (that is, I will expect you to have read the book for Class 2, April 6; as always the reaction is due a day before the next class). Also read the 'CD getting started' information on the course web site and follow its instructions.
Start Group Quilt project

<u>Week 2, April 6</u>
Start Microworlds Logo programming assignment
Start Pairs Microworlds Logo project
Group Quilt project due

Discuss reading: Mindstorms

Week 3, April 13 (No class due to AERA)

Microworlds Logo programming assignment due
Harel, I., and Papert, S. (1990). Software Design as a Learning Environment. I*nteractive Learning Environments*, vol. 1, no. 1, pp. 1-32.
Edwards, L. (1995). “Microworlds as Representations.” *Proceedings of the 2nd International NATO Symposium on Advanced Technology and Education*.
Extra posting on reading: read all posts and send comment on each others’ posts by Friday

Week 4, April 20

Pairs Microworlds Logo project due
Start NetLogo or Robotics project
Discuss reading:
Papert, S. (1991). "Situating Constructionism." In Constructionism, edited by I. Harel and S. Papert. Norwood, NJ: Ablex Publishing
Selections from Jean Piaget

Week 5, April 27

Start Final Project Design Specification
Discuss reading:
Martin et al (2000). To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines. *In Robots for Kids: Exploring New Technologies for Learning Experiences*. (Edited by Allison Druin). Morgan Kaufman / Academic Press, San Francisco,
Wilensky, U. (2001) Modeling Nature’s Emergent Patterns with Multi-agent Languages. *Proceedings of EuroLogo 2001*. Linz, Austria.

Week 6, May 4

NetLogo or Robotics project due
Discuss reading:
Wilensky, U. & Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Perspective to Making Sense of the World. *Journal of Science Education and Technology*. Vol. 8 No. 1. pp. 3 – 18.

Week 7, May 11

Final Project Design Specification Due

Start Software review

Discuss reading:

Oren, T. (1990). “Designing a New Medium.” In *The Art of Human-Computer Interface*

Design (edited by B. Laurel). Reading, MA: Addison Wesley.

Week 8, May 18
Discuss reading:
Perkins, D. (1991). "Technology Meets Constructivism: Do They Make a Marriage?" *Educational Technology*, May 1991

Week 9, May 25

Software review due

Discuss reading: To be determined

Week 10, June 1
Final Project Software Specification due

Discuss reading: To be determined

Week 11, June 6 (Monday)
Final Project due on Friday, June 3
Final Project Presentations

Course packet readings:

DiSessa, A. A. (1997). Open Toolsets: New ends and new means in learning mathematics and science with computers. In E. Pehkonen (Ed.), *Proceedings of the 21st Conference of the International Group for the Psychology of Mathematics Education, Vol. 1,* Lahti, Finland, 47 – 62.

Edwards, L. (1995). "Microworlds as Representations." *Proceedings of the 2nd International NATO Symposium on Advanced Technology and Education*.

Eisenberg, M. (1991). "Programmable Applications: Interpreter Meets Interface*." MIT AI Lab Memo.*

Falbel, A. (1991). The Computer as a Convivial Tool. In I. Harel & S. Papert (Eds.). *Constructionism*. (p. 29 – 40). Norwood, New Jersey: Ablex Publishing

Goldman, R. (in press). Perspectivity Technologies: Computers, the Internet, and New Media for Learning. In *American Psychological Association Handbook*. New York: Wiley & Sons, Inc.

Hancock, C. (2001)."Children's Understanding of Processes in the Construction of Robot Behaviors." (Flogo Introduction)

Harel, I., and Papert, S. (1990). Software Design as a Learning Environment. I*nteractive Learning Environments*, vol. 1, no. 1, pp. 1-32.

Martin et al (2000). To Mindstorms and Beyond: Evolution of a Construction Kit for Magical Machines. *In Robots for Kids: Exploring New Technologies for Learning Experiences*. (Edited by Allison Druin). Morgan Kaufman / Academic Press, San Francisco, Oren, T. (1990). “Designing a New Medium.” In *The Art of Human-Computer Interface Design* (edited by B. Laurel). Reading, MA: Addison Wesley.

Papert, S. (1991). "Situating Constructionism." In Constructionism, edited by I. Harel and S. Papert. Norwood, NJ: Ablex Publishing.

Perkins, D. (1991). “Technology Meets Constructivism: Do They Make a Marriage?” *Educational Technology*, May 1991.

Piaget, J.(1952). Conservation of Continuous Quantities. *The child's conception of number*, (p. 3-17). London: Routledge and Kegan.

Piaget, J. (1929). The Child’s Conception of the World.

Piaget, J. (1952). The Origins of Intelligence in Children.

Resnick, M., Martin, F, Sargent, R & Silverman, B (1996). Programmable Bricks: Toys to Think with. *IBM Systems Journal.* Vol. 35, Nos. 3&4.

Smith, D., Cypher, A. & Tesler, L (2000). Novice Programming comes of Age. In *Your Wish is My Command.* (edited by Henry Lieberman).

Wilensky, U. & Resnick, M. (1999). Thinking in Levels: A Dynamic Systems Perspective to Making Sense of the World. *Journal of Science Education and Technology*. Vol. 8 No. 1. pp. 3 – 18.

Wilensky, U. (2001) Modeling Nature’s Emergent Patterns with Multi-agent Languages. *Proceedings of EuroLogo 2001*. Linz, Austria..

In addition, there is a second course packet of the software manuals:

Microworlds EX Manuals

NetLogo Manual - can be viewed at http://ccl.northwestern.edu/netlogo – click on the Users Guide

LEGO manuals

Martin, F. (1995) The Art of Lego Design. *The Robotics Practitioner: the journal for robot builders*, vol. 1 No. 2

Extra Readings (for reference only, not part of course packet)

Books:

Bolter, J. (1991). Writing : The Computer, Hypertext, and the History of Writing . Hillsdale, NJ: Lawrence Erlbaum.

Bowers, C. A. (1988). *The Cultural Dimensions of Educational Computing: Understanding the non-neutrality of Technology*. Teachers College Press.

Geisert, P. & Futrell, M. (1995). *Teachers, Computers and Curriculum* . Allyn & Bacon.

Grabe, M & Grabe, C. (1996). *Integrating Technology for Meaningful Learning.* Houghton Mifflin.

Harel. I & Papert, S. (Eds.) (1990). *Constructionism*. Ablex.

Harel, I. (1991) Children Designers: Interdisciplinary Constructions for Learning and Knowing Mathematics in a Computer-Rich School. Norwood, NJ: Ablex Publishing.
ISBN 0-89391-788-5

Illich, I. (1976). *Tools for Conviviality*. Harper & Row.

Kafai & Resnick (1993) *Constructionsim in Practice*. NJ: Lawrence Erlbaum.

Knapp, L. & Glenn, A. (1996). *Restructuring Schools with Technology*. Allyn & Bacon.

Maddux et al (1997). *Educational Computing.* Allyn & Bacon.

Male, N. (1997). *Technology for Inclusion.* Allyn & Bacon

Laurel, B. (Ed.) (1990). *The Art of Human Interface Design*. Addison Wesley.

McLuhan, M. (1964). *Understanding Media* .

Norman, D. A. (1988). *The Psychology of Everyday Things*. Basic Books.

Papert, S. (1992). *The Children's Machine: Rethinking Schools in the Age of the Computer*. Basic Books.

Papert, S. (1996). *The Connected Family*.

Perelman, L. (1992). *School is Out*.

Perkins, D. et al (Eds.) (1995). *Software Goes to School: Teaching for Understanding with new technologies*. Oxford University Press.

Postman, N. (1992). Technopoly. The surrender of culture to technology. New York: Knopf.

RelLab (1994). RelLab. Experimenting with relativity. Version 2.0. User Manual

Solomon, C. (1986). *Computer Environments for Children*. MIT Press

Taylor, R. (1980). *The Computer in the School: Tutor, Tool, Tutee*. Teachers College Press.

Turkle, S. (1995). Identity in the age of the internet. . New York: Simon & Schuster.

Articles:

Ackerman, E. (1991). From Decontextualized to Situated Knowledge: Revisiting Piaget's Water-Level Experiment. In I. Harel & S. Papert (Eds.). *Constructionism*. (p. 269 – 294). Norwood, New Jersey: Ablex Publishing.

Ackerman, E. (1996). Perspective-Taking and Object Construction: Two Keys to Learning. In Y. Kafai & M. Resnick (Eds.) *Constructionism in Practice* (p. 25 – 36). Mahwah: New Jersey: Lawrence Erlbaum Associates.

Anderson, J.R., Corbett, A., Koedinger, K. & Pelleetier, R. (1995). *Cognitive* Tutors: Lessons Learned. *Journal of the Learning Sciences* . 4, 2.

Bamberger, J. (1996). Turning Music Theory on its Ear: Do we hear what we see; Do we see what we say? *International Journal of Computers for Mathematical learning,* Vol. 1 N1.

Bareiss, R. & Beckwith, R. (199x). Advise the President: A Hypermedia System for Teaching Contemporary American History.

Borovoy, R., McDonald, M., Martin, F., & Resnick, M. (1996). Thinking Tags . *IBM Systems Journal*.

Bruckman, A. Programming for Fun: MUDs as a context for collaborative learning. MIT Media Lab.

Bruckman, A. E. (2000). Situated Support for Learning: Storm's Weekend with Rachael. *Journal of the Learning Sciences, 9 (3),* 329 – 372.

Collins, A. & Brown, J.S. (1988). The Computer as a Tool for Learning Through Reflection. In H. Mandl & A. Lesgold (Eds.). *Learning Issues for Intelligent Tutoring Systems* (pp. 1-18). New York: Springer Verlag.

Confrey, J. (1993). The role of technology in reconceptualizing functions and algebra. In J.R. Beceker & B.J. Pence (Eds.), Proceedings of the Fifteenth Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education Vol. 1, Pacific Grove, CA, October 17-20. San Jose, CA: San Jose State University, Center for Mathematics and Computer Science Education, p. 47-74.

diSessa, A., and Abelson, H. (1986). "Boxer: A Reconstructible Computational Medium. Communications of the ACM, vol. 29, no. 9, pp. 859-868.

diSessa, A. A., Abelson, H., & Ploger, D. (1991). An overview of Boxer. TheJournal of Mathematical Behavior, 10(1), 3-15.

Duffy, T. & Jonassen, D. (1991). Constructivism: New implications for instructional technology?, Educational Technology, May 1991, 7-12.

Edelson, D. Pea, R. & Gomez, L. (1994). Constructivism in the Collaboratory.

Fischer, G., & Lemke, A. C. (1987). Construction kits and design environments: Steps toward human problem-domain communication. Human-Computer Interaction, 3, 179-222.

Hancock, C., Kaput, J. & Goldsmith, L. (1992). Authentic inquiry with data: Critical Barriers to classroom implementation, Educational Psychologist, 27 (3), 337-364.

Hancock, C. (1993). The Medium and the Curriculum: Reflections on Transparent Tools and Tacit Mathematics. In A. A. diSessa, C. Hoyles, R. Noss (Eds.). *Computers and Exploratory Learning* (p. 221 - 240). Springer.

Harvey, B. (1991). "Symbolic Programming vs. the A.P. Curriculum." The Computing Teacher, vol. 18, no. 5 (Feb. 1991)

Horwitz, P. Contrasting styles in the design of science software. Bolt Beranek and Newman, Inc

Horwitz, P., Neumann, E. & Schwartz, J. (1994). The Genscope Project, *Connections*, Spring 1994, 10-11.

Horwitz, P. (1989). ThinkerTools: Implications for science teaching. In J.D. Ellis (Ed.), 1988 AETS Yearbook: Information technology and Science Education.

Horwitz, P., Taylor, E. & Barowy, W. (1994). Teaching special relativity with a computer, *Computers in Physics*, 8 (1), 92-97.

Hoyles, C. (1991). Computer-based learning environments for Mathematics. In A. Bishop, S. Mellin-Olson, and J. Van Dormolen (Eds.), *Mathematical Knowledge: Its growth Through Teaching*. Dordrecht: Kluwer, 147-172.

Kahn, K. (2000). Generalizing from Examples. In *Your Wish is My Command.* (edited by Henry Lieberman).

Kafai, Y. & Harel, I. (1991). Learning Through Design and Teaching: Exploring Social and Collaborative Aspects of Constructionism. In I. Harel & S. Papert (Eds.). *Constructionism*. (p. 85 – 110). Norwood, New Jersey: Ablex Publishing.

Kay, A., & Goldberg, A. (1977). Personal dynamic media. *IEEE Computer*, 31-41.

Koedinger, K. Using ANGLE: A cognitive tutor for geometry proof. Human-Computer Interaction Institute.

Koschmann, T. (1993). “Using Technology to Assist in Realizing Effective Learning and Instruction: A Principled Approach to the Use of Computers in Collaborative Learning.” *Journal of the Learning Sciences*, vol. 3, no. 3.

Kozma, R. B. (1991). Learning with media. Review of Educational Research, 61(2), 179-211.

Kuttner, R. The revival of a lost art, The Boston Globe.

Noss, R. & Hoyles, C. (1991). Logo and the Learning of Mathematics: Looking Back and Looking Forward. In Hoyles, C. & Noss, R. (Eds.) *Learning Mathematics and Logo*. London: MIT Press. p. 431-468.

Papert, S. (1990). A critique of technocentrism in thinking about the school of the future. Epistemology and Learning Group Memo No. 2. MIT Media Laboratory: Cambridge, MA.

Papert, S. (1993). Literacy and letteracy in the media ages, Wired, May/June 1993.

Papert, S., and Solomon, C. (1971). "Twenty Things to do with a Computer." Artificial Intelligence Memo 248, MIT AI Laboratory. Cambridge, MA.

Pea, R., Edelson. D. & Gomez, L.M. (1994). Distributed Collaborative Science Learning Using Scientific Visualization and Wideband telecommunications. Paper presented at the 160th. meeting of the American Association for the Advancement of Science.

Reiser, B. (1989). Pedagogical Strategies for Human and Computer Tutoring. Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, CA, March 30-April 2.

Repenning, A. (1994). Programming substrates to create interactive learning environments, Interactive Learning Environments, 4 (1), 45-74.

Resnick, M. (1994). “Behavior Construction Kits”. Communications of the ACM

Resnick, M., and Ocko, S. (1991). "LEGO/Logo: Learning Through and About Design." In *Constructionism* (ed. by I. Harel and S. Papert). Ablex Publishing.

Roschelle, J. (1992). Collaborative Inquiry: Reflections on Dewey and Technology for Situated Learning.

Scardamalia, M. & Bereiter, C. (1991). Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media, *The journal of the learning sciences*, 1 (1), 37-68.

Schank, R.C. & Jona, M.Y. (1991). Empowering the student: New perspectives on the design of teaching systems, The journal of the learning sciences, 1 (1), 7-35.

Schank, R. (1994). Goal-Based Scenarios: A Radical Look at Education. *Journal of the Learning Sciences,* Vol. 3 No. 4.

Schwartz, J. (199). Intellectual mirrors: A step in the direction of making schools knowledge-making places, Harvard Educational Review.

Soloway, E., Guzdial, M., & Hay, K. E. (1994). Learning Centered Design. Interactions, 1(2).

Thornton, R. (1993). Changing the Physics Teaching Laboratory: Using technology and new approaches to learning to create an experiential environment for learning physics concepts, *Proceedings of the Europhysics Conference on the Role of Experiment in Physics Education*. Ljubljana, Slovinia.

Turkle, S. & Papert, S. (1991). Epistemological pluralism and the revaluation of the concrete. In I. Harel & S. Papert (Eds.), Constructionism. Norwood, NJ: Ablex Publishing Corporation.

White, B. & Frederiksen, J. (1994). ThinkerTools II: A curriculum for teaching scientific inquiry and modeling.

White, B. Y. and Frederiksen, J. R. (2000). Technological Tools and Instructional Approaches for Making Scientific Inquiry Accessible to All. Innovations in Science and Mathematics Education: Advanced Designs for Technologies of Learning. M. J. Jacobson and R. B. Kozma. Mahwah, NJ, Lawrence Erlbaum Associates: 321-360.