
• Bruckman Amy (1993), Programming for Fun: MUDs as a Context for Collaborative Learning, MIT Media Lab, working paper.


• Evard, Rémy (1993), “Collaborative Networked Communication: MUDs as Systems Tools, LISA - November 1-5, Monterey, CA


**WWW**


**Hypertext & Hypermedia (totally uncomplete)**


• NIELSEN, Jakob [1990], Hypertext and Hypermedia, Boston: Academic Press.


• WOODHEAD [1990], Nigel, Hypertext and Hypermedia, Theory and Applications, Wilmslow: Sigma Press.

**Textual Virtual Realities, Virtual Classrooms, CSCW**


• JONES M. & WIPOND K, (1991) Intelligent environments for curric-


Learning Theory & Instruction Theory


Advanced Learning Environments

Appendix 5 Bibliography

Cited (well, some)


Appendix 4 Educational Hypermedia

Related Projects

1. Hyper-G: A Universal Hypermedia System (cf. Kappe, Maurer & Sherbakov 93). For a demo session with a simple client: telnet to finfo.tu-graz.ac.at and login as info. It can be accessed in limited ways by W3 clients (http ??).

2. JITOL

3. CSILE: (cf. Scardamalia)

(Appendixes are getting worse, dont they?)
ences. These differences result from an increase in the difficulty of the challenges proposed by the coach. More complex challenges compel the learner to handle a larger number of dimensions and hence increase the working memory load. At the end of the second sub-level, the learner receives challenges that already belong to the next level. This shows the learner the necessity to have more powerful control structures to solve the proposed challenge (As in Case theory sub-level i.4 is equivalent to sub-level i+1.0). The “reunification” of the objects used at some level in a new more powerful object frees the memory resources necessary to solve the problem.

(to be continued, most of the Case stuff should be cut....)
classical four stages:

- Perception of objects and motor activities
- Relations between motor activities
- Manipulation of dimensions (quantifiable variables)
- Second order dimensions (ratios)

How do we explain the formation of new units and the transition between stages? According to Case, each new sub-stage within a stage is characterized by the subordination of a new basic unit to the executive control structure: the first sub-stage has two basic units, the second has three and the third has four. The complexity of subordination reached at the final sub-stage (in stage n) is such that it corresponds to a basic unit at the next stage (stage n+1). When the executive control structure of stage n+1 subordinates two of these basic units passed up from below, it will enter its own first sub-stage... and so on. The last sub-stage of stage n can thus be considered as sub-stage zero of stage n+1. In other words, the four-unit control structure of stage n can be translated into a one-unit control structure at stage n+1. It is this formal process which Case calls “hierarchical integration”.

The increase in “Short Term Storage Space” (STSS) permits the transition from one sub-stage to the next. This increase is achieved within the “Total Processing Space” (TPS) which also contains the “Operating Space” (OS) utilized to control the active schema. STSS increases with age during development as a result of the maturation of the nervous system. It also increases during the learning of schemata as the result of an increase in the efficiency of the control structure: as the learner masters a task, the compilation of her knowledge frees up short term memory to hold new objectives.

There is an obvious mapping between the structure defined by Case and our intermediate framework. The control structures at each level of the pyramid integrate the control structures located at the lower level. The sequence of microworlds within the pyramid is structured as Case’s view of development: quantitative variations define the improvement possible within some level (or microworld or stage) while the qualitative variations define the transition between two levels. The concept of stage transition is translated into the language shift mechanism. This transition is necessary when the learner tries to solve problems that have too high memory load constraints. After the language shift, the learner has at her disposal new control structures that enable her to solve the problems with a reduced cognitive load.

The shift from one level to another, i.e. to shift from one language to another corresponds to some qualitative jump in learning. Within each level, we defined four sub-levels that are discriminated by quantitative differ-
Our intermediate framework also introduces the designer to the theories of Vygotsky. The *apprenticeship* idea is reified in the pyramid model by sharing control between the coach and the learner: when the learner is able to perform at some level L, the tutor must guide her activities at level L+1. This level L+1 corresponds to the concept of zone of proximal development (Vygotsky, 1978). At each language shift, the learner will assume a more important control of his solution process and the coach’s guidance will be reduced. Moreover, Wertsch (1985) proposed a linguistic analysis of the internalization process that relates it to the language shift. He observed (in mother-child interactions) that the move from the inter-individual to the intra-individual plane was preceded by a language shift inside the inter-individual level: mothers replace a descriptive language by a strategy-oriented language (i.e. a language that refers to objects according to their role in the problem solving strategy).

The third but central theoretical background that fits with our framework is the neo-Piagetian theory of Robbie Case (1985). We focused on this theory because of its rather operational form. The key idea in Case’s theory of intellectual activity and development is what he calls the “executive control structure”. He believes that problem solving across domains can be viewed as the execution of a “mental plan” defined as a program of schemata. There are two types of schemata: “figurative schemata” represent states and “operative schemata” represent transformations. The mental plan is divided into three main sub-components.

- A representation of the “problem situation”: this is the set of conditions relevant to the plan. The complexity of the representation will depend directly on the complexity of the problem.
- The goals to be achieved defined as a set of new states, or “solution situation”.
- The “problem solving process” to be used, stated as a set of operations that transform the problem situation into the solution situation.

These components are further analyzed. Elements of the problem situation are mapped to elements in the solution situation, and both are mapped to transformations in the strategy set. The result is a well-defined formal structure associating specific tasks with problem solving processes in a rigorous way.

Case formulates his general theory with reference to developmental stages in specific domains. One of the characteristics of his theory is that it relates *quantitative changes* within a stage to *qualitative changes* between stages: for example, an increase in the active unit capacity of working memory occurs within a stage, but helps to explain the transition to the next stage. Case distinguishes activity within a stage (i.e. a “sub-stage”) by first defining what he calls “basic units of thought”. He then notes that during development (and probably also during skill acquisition) we have the
poses sequences of elementary actions. The description language is the set
of symbols (strings, graphics,...) used by the computer to show the learner
some description of her behavior. This description reifies some abstract
features of the learner’s behavior in order to make them explicitly available
for metacognitive activities (Collins and Brown, 1988).

The command and description languages are different at each level
of the pyramid, but each level integrates its lower neighbor. This integra-
tion is encompassed in the relationship between the languages used at suc-
cessive levels: if a description language at level L is used as a new com-
mand language at level L+1, then the learner is compelled to use explicitly
the concepts that have been reified at level L. This is what we called the
language shift mechanism (Dillenbourg, 1992): when she receives a new
command language, the learner must explicitly use the concepts that were
implicit in her behavior. The meaning of the new commands has been in-
duced at the previous level by associating the learner’s behavior with some
representation. This representation is now the new command.

The process by which properties that are implicit at some level of
knowledge can be abstracted and explicitly reached at the higher level has
been studied under the label of reflected abstraction (Piaget, 1971). The
language shift mechanism has two uses. Firstly, it translates this psy-
chological concept in a terminology more relevant for ILE designers. Second-
ly, it describes a pedagogical strategy (mainly inductive) to trigger reflect-
ed abstraction. By applying the framework to ILE design, we not only
ground the structure of learning environments in a model of cognitive de-
development. But such models of development can be tested through the dif-
cult process of implementation. We found that this intermediate frame-
work can be used to “interface” several theoretical backgrounds. Most psy-
chological theories address actually only a specific facet of learning while
an ILE designer must consider learning in its globality and complexity.
Therefore, an intermediate framework should integrate multiple theoretical
bodies of knowledge, each relevant for some aspect of reality. An edu-
cational computing system must account for the importance of discovery,
for the role of practice and for the effect of coaching, because all of them
occur at some stage of learning in the real world. The framework we pro-
pose can be read from different theoretical perspectives.

From Campbell and Bickhard’s (1986) viewpoint, the language shift
mechanism can be viewed as a process of inducing interaction patterns.
An elementary interaction associates some sequence of user’s actions and
the computer’s description of this sequence. Inferring the meaning of the
description language can indeed be described as the result of inducing the
relationship between the actions performed and their representation (Dil-
lenbourg, 1992). This corresponds to a view of knowledge as something
that stands in the interaction between the subject and her environment. It
creates a bridge between our model and current research on situated learn-
ing (Brown, 1990), a “hot” issue in AI and Education.
Appendix 3 Research on Advanced Learning Environments

Ok, this some of our own work, the important idea in here is the modern advanced learning environments stress much less the “Intelligent Tutoring Aspect, but rather design of a global learning environment taking into account learning (and some instructional) theories, making use of all useful technologies available. Whereas it is not realistic to see the kind of experimental programs we play with in practice, some ideas from our research can be taken and transferred in any kind of learning environment.

One of the main research line of TECFA are ‘intelligent learning environments’ (ILE). An ILE refers to a category of educational software in which the learner is ‘put’ into a problem solving situation. A learning environment is quite different from traditional courseware based on a sequence of questions, answers and feedback. The best known example of a learning environment is a flight simulator: the learner does not answer questions about how to pilot an aircraft, he learns how to behave like a “real” pilot in a rich flying context. Experience with learning environments (like LOGO) showed that those systems gain efficiency if the learner is not left on his own but receives some assistance. This assistance may be provided by a human tutor or by some system components. In our flight simulator example, the future pilot would gain from discussing his actions with an experienced pilot. The implementation of these agents is based on artificial intelligence techniques in advanced experimental learning environments. In summary, we use the word ‘intelligent learning environment’ for learning environments which include (1) a problem solving situation and (2) one or more agents that assist the learner in his task and monitor his learning.

Designing an intelligent learning environment (ILE) involves implementing some theory of learning and teaching. However, most available theories do not have the level of operationality required for implementation work. Designing an ILE is real research work. We are developing an intermediate framework that builds a bridge between theories and implementations by translating psychological knowledge into terminology more relevant to computer scientists. It specifies the cognitive architecture of systems like MEMOLAB. Let’s examine two key concepts: the pyramid metaphor and the language shift mechanism.

The “pyramid” metaphor represents the concepts and skills to be acquired by the learner, ranked bottom-up according to their level of “hierarchical integration”. Learning consists in moving up in the pyramid. Each level of the pyramid is defined by two languages: the command language and the description language. The command language vocabulary is the set of elementary actions that the learner is allowed to do at some stage of interaction. The command language syntax defines how the learner com-
4. synthesizers
5. analogies
6. cognitive strategy activators
7. a learner control format

The first component is the critical as far as elaboration theory is concerned. The elaborative sequence is defined as a simple to complex sequence in which the information epitomizes (rather than summarize or abstract) the ideas that follows. Epitomizing should be done on the basis of a single type of content (concepts, procedures, principles) and involves the presentation of a few fundamental or representative ideas that can form the basis for the lesson/course.

Merill’s Component Display Theory (CDT)

Merrill’s CDT is probably still the most detailed theory on how to teach a single idea or concept. It provides at a micro-level what Gagné-Briggs provide at a macro-level. Not surprisingly it is more concerned with cognitive issues than with instructional ones: CDT attempts to indicate what set of methods is most likely to optimize learning under some specified conditions. CDT classifies learning objectives on 2 dimensions:

1. Content
   - facts
   - concepts
   - principles
   - procedures

2. Performance
   - remember
   - use
   - find

(to be continued ..... )
Appendix 2 Instructional Design Theory: Sequencing & Chunking of Educational Material

Gagné’s steps of instruction

What is the optimal sequencing of courseware and how is it related to various types of learning? Gagné suggests nine universal steps of instruction (cf. Gagné 85 or Aronson 1983) which should be found in any instructional context:

1. **Gain attention** e.g. present a good problem, a new situation, use a multimedia advertisement.

2. **Describe the goal:** e.g. describe the goal of a lesson (task,...), state what students will be able to accomplish and how they will be able to use the knowledge, give a demonstration if appropriate.

3. **Stimulate recall of prior knowledge** e.g. remind the student of prior knowledge relevant to the current lesson (facts, rules, procedures or skills). Show how knowledge is connected, provide the student with a framework that helps learning and remembering. Tests can be included.

4. **Present the material to be learned** e.g. text, graphics, simulations, figures, pictures, sound, etc. e.g. follow a consistent presentation style, chunking of information (avoid memory overload, recall information)

5. **Provide guidance for learning** e.g. presentation of content is different from instructions on how to learn. Should be simpler and easier that content. Use of different channel.

6. **Elicit performance** “practice”, let the learner do something with the newly acquired behavior, practice skills or apply knowledge

7. **Provide informative feedback** show correctness of the trainee’s response, analyze learner’s behavior (or let him do it), maybe present a good (step-by-step) solution of the problem

8. **Assess performance** test, if the lesson has been learned. also give sometimes general progress information

9. **Enhance retention and transfer** inform the learner about similar problem situations, provide additional practice. Put the learner in a transfer situation. Maybe let the learner review the lesson.

Reigeluth’s “Elaboration Theory of Instruction

Elaboration theory (Reigeluth 83:342) proposes seven major strategy components:

1. an elaborative sequence
2. learning prerequisite sequences
3. summarizers
what to do beside.

“A typical study skill program is SQ3R [applicable to concept learning/D.S] which suggests 5 steps: (1) survey the material to be learned, (2) develop questions about the material, (3) read the material, (4) recall the key ideas, and (5) review the material.” (Kearsley: 93).
3. Instruction should be designed to facilitate extrapolation and or fill in
the gaps (going beyond the information given).

An other early contribution was Ausubel’s (63) subsumption theory con-
cerned with how individuals learn large amounts of meaningful material
from verbal/textual presentations in a school setting (as opposed to rote or
discovery learning). He initiated that instructional sequences should make
content more meaningful for the learner. He postulates (cf. Kearsley 93,
Reigeluth 83:339) that:

- Instruction (of verbal information) should start with general knowledge
  that subsumes content presented by successive differentiation, i.e. the
  most general and ideas of a subject should be presented first and then
  progressively differentiated in terms of detail and specificity.

- More generally, instructional materials should attempt to integrate new
  material with previously presented information through comparisons
  and cross-referencing of new and old ideas.

Both Reigluth’s (83) “Elaboration Theory” and Merrill’s (83) “Component
Display Theory” are based on work by Bruner and Ausubel.

Other more recent lines of research combine cognitivist information theory
with results from more traditional experimental memory research.

An example is the Act* Theory using Intelligent Tutors as a test bed (cf.
Anderson 87). “According to ACT*, all knowledge begins as declarative
information; procedural knowledge is learned by making inferences from
already existing factual knowledge. ACT* involves three types of learn-
ing: generalization, in which productions become broader in their range of
application, discrimination, in which productions become narrower in their
range of application, and strengthening, in which some productions are ap-
plied more often. New productions are formed by the conjunction or dis-
junction of existing productions. It is interesting to compare these three
types of learning with the three modes of learning (accretion, restructuring,
tuning) proposed by Rumelhart & Norman (.)” (Kearsley: 93).

Principles:
1. Identify the goal structure of the problem space to the learner.
2. Provide instruction in the context of the problem-solving task.
3. Provide immediate feedback on errors.
4. Minimize the working memory load.
5. Adjust the “grain size” of instruction to account for the knowledge com-
pilation process.
6. Enable the student to approach the target skill by successive approxima-
tion.

With partially automatized environments such as Hypertext course on the
Web, the student should be told how to use the material, how to read it and
nitive task behaviors (that are being used while learning or that are targets for learning). As an example, Kearsley (93) lists the following types of task behaviors:

- Searching for/receiving information (detects, observes, inspects, identifies, reads, surveys)
- Processing information (categorizes, calculates, codes, itemizes, tabulates, translates)
- Problem-solving (analyzes, formulates, estimates, plans)
- Decision-making (examines, chooses, compares, evaluates)
- Communication (advises, answers, directs, informs, instructs, requests, transmits)
- Sensory-motor processes (activates, adjusts, connects, regulates, tracks)

By combining those two kinds of typologies one can imagine the “haystack” Instructional Design theory is faced with when trying to operationalize how to learn what.

Other categories of learning types has been proposed such as the ones by Gagné (Aronson 83:81, Gagné 87: 64), i.e. (1) Intellectual Skill, (2) Verbal Information, (3) Cognitive Strategy (problem solving), (4) Attitude, (5) Motor Skill. In any case, it think it is useful in this context to distinguish at the least the following basic categories:

1. Factual Information & Concepts (Verbal Information): Remember and discriminate things
2. Problem Solving & Reasoning (Cognitive Strategy): Apply general or domain-specific heuristics to problem situations
3. Procedural skills: Learn how to do simple or complex tasks more or less automatically.

**Learning/Teaching Strategies & Principles**

How can we have the learner use an appropriate learning strategy? In some learning environments (specially the fully computer-based ones) learning and teaching strategies are integrated into its design. In others they are delivered apart. Principles and Strategies vary according to the type of learning and different theoretical orientations.

Bruner (66), inspired by Piaget, focussed on how people construct new knowledge. His constructivist approach (discovery methods and intellectual stages) still inspires current theories.

1. Instruction must be concerned with the experiences and contexts that make the student willing and able to learn (readiness).
2. Instruction must be structured so that it can be easily grasped by the student (spiral organization of the curriculum).
Appendix 1 Some Learning Theory Background

In a behaviorist view, “Learning” can be defined as something that occurs when a learner acquires the capacity to do something. The LE designer must provide the conditions for this process. For each type of learning, some conditions work best and some don’t. Let’s look at 2 classifications:

Types of Learning

(according to Kearsley 1993):

- Attitudes: "....Disposition or tendency to respond positively or negatively towards a certain thing (idea, object, person, situation)." Also: Choose to behave this or that way according to opinions and beliefs.

- Factual Information (Memorization): Processing of factual information and remembering is tied to previous knowledge. Memory research has also a lot to say about processing constraints.

- Concepts (Discrimination): Concept learning encompasses learning how to discriminate and categorize things (with critical attributes). It also involves recall of instances, integration of new examples and subcategorization. Concept formation is not related to simple recall, it must be constructed.

- Reasoning (Inference, Deduction): “Reasoning encompasses all thinking activities that involve making or testing inferences. This includes inductive reasoning (i.e., concept formation) and deductive reasoning (i.e., logical argument). Reasoning is also closely related to problem-solving and creative behaviors”.

- Procedure Learning: Procedures refer to being able to solve a certain task by applying a procedure. Once a procedure is mastered its excused usually does not take much effort (e.g. ftp a file). Cognitive theories like Act or Soar are interested in this, because procedures are important in diminishing cognitive load.

- Problem-Solving: A good example is Newell & Simons information processing paradigm for the study of problem-solving and the concepts of “means-ends-analysis” and “problem space”. According to their GPS framework, problem-solving involves the identification of subgoals and the use of methods (especially heuristics) to satisfy the subgoals. And important contribution was also the methodology of protocol analysis (of “thinking aloud methods” which has been extensively used by Anderson (87) to implement intelligent tutoring systems according to his Act* theory (Anderson 83).

- Learning Strategies: can be learned too to some extent. Very much dependant on what you want to learn

- Sensory-Motor:

Note that learning types can be strongly related to different kinds of cog-
So what can we do with the Web?

good filled in table will get a Swiss beer!

**Table 1: Filling in the Functionalities of a Learning Environment**

<table>
<thead>
<tr>
<th></th>
<th>teacher</th>
<th>monitor</th>
<th>fellow learners</th>
<th>learning material</th>
<th>Ext. info sources</th>
<th>tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple WWW</td>
<td>partly</td>
<td>no</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(guidelines)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WWW with forms &amp; server side scripts</td>
<td>badly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Q&amp;A)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WWW with local data processing clients</td>
<td>partly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(tutoring system)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WWW with intelligent server-side computing</td>
<td>partly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>computer or telephone talk</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>virtual reality</td>
<td>yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. teacher: His role is provide something between loose guidance & direct instruction. It can be a human agent (present or distant) or an intelligent agent. He provides information from the curriculum to the task level.

2. monitor: The Monitor ensures that something is learner. A role taken by either the teacher, the learner (self-control) or by some program.

3. fellow learners: Improve the learning process (some research tries to implement artificial ones)

4. learning material: Learning material contains what has to be learned in a very broad sense (e.g. knowing what, knowing how). It can be computational in various ways (exploratory hypertext, lesson & task oriented hypertext, simulation software, task solving environments, etc.

5. External info sources: All kinds of information which is not directly stored in the learning material (e.g. additional material, handbooks, manuals, etc.)

6. tools: Everything which may help the learning process other then the learning material (e.g. calculators, communication software, etc.)

Let’s examine the Table 1, “Filling in the Functionalities of a Learning Environment,” on page 10. The first WS participant who comes up with a
Courseware engineering & the learning environment

In more simple and technical terms, courseware engineering (cf. De Dian & van Schaik 93:193) is concerned with:

- transferring educational information
- organizing pedagogically optimized access to this information via an appropriate interface and structuring of the material
- implementation of instructional tactics, e.g.:
  - giving examples
  - multiple choice questions
  - asking the student to perform a task, etc.
  - telling the student what learning strategy to adopt with some material
  - ....
- implementation of instructional strategies, i.e. sequencing of teaching materials

Furthermore, Courseware engineering is not everything. Courseware alone rarely constitutes the full learning environment. Authors of Instructional Material on the Web should be aware of the following:

- Teaching and learning involves a learning environment. It is not good enough to hypertextify a text or implement form-based tests. Good Web based courseware is more global in design. Not everything must be built into html or server-side scripts, but somehow instructional tactics & strategies as well as learning strategies have to be “put” upon the learner or communicated to the learner. Last, but not least he needs assistance and very often cooperation with other learners.
- Delivery of instructional text (multimedia and hypermedia) has be structured according to some pedagogical & learning strategies. If the learner has to discover everything himself - fine - but tell him so!

So what can we do with the Web?

In discussing the role of technological support in education, Sandberg (94:225) identifies the components of a (technologically rich) learning environment (see “Overview of the Learning Environment (Sandberg 94:225)” on page 9). These components must all be there in order to optimize learning. However, they can be “implemented” in many different ways. Each component has functionalities for which we should insure:
• Knowledge Construction & Environments & Intellectual Toolkits

Now match that to those more technical items:

1. Information servers to look up information (manuals, books, expositions, bibliographies, programs, etc.)
2. Distribute educational material (texts, programs)
3. Provide curricula & guidance to lessons and exercises in hypertext format.
4. Implement collaborative work (dynamic hypertext, “News like conferencing system”, co-writing)
5. Implement Jigsaw puzzles
6. Question & Answering, (tests, Skinner & Bloom type of learning to some extent)
7. Interface to local clients (e.g. simulations, programming environments, tutors, etc.)
8. The same thing over the web (e.g. have intelligent nodes, cf. Mallery)
9. ....

The WWW, specially in conjunction with external local or server-side clients offers a lot of possibilities. But not everything can be done and it is important not to use in inappropriate learning paradigm for a given educational goal.

7 Educational Hypermedia

Integration of Hypermedia

There are 3 aspects:

1. Integration into a learning environment setup: What role in learning & teaching does hypermedia have? (see also “What learning activity can we do with the Web?” on page 6) What other tools do we need?
2. Integration into a learning environment architecture: How can we make integrate hypermedia with other computational learning tools?
3. Computational integration with other programs: How can we build achieve tight integration?
5 Research on Advanced Learning Environments

• For background information, see: “Research on Advanced Learning Environments” on page 17

Summary

• The Learner must be active

• A learning environment should be designed as powerful dedicated working environments. It must be rich and complex reflecting the essential properties of what has to be learned.

• The environment must be structured. If the richness of a learning environment is a quality, its complexity may reduce learning. It must provide optimal learning conditions in function of the learner’s stage of knowledge.

• Learning environments should be designed as hierarchical knowledge base generators

• Learning environments should present knowledge as a communication system. A learner must interact with agents, tutors.

At the current state of the WWW technology, it is not possible to implement this kind of advanced learning environment (without making use of external clients). However, there are points that WWW based courseware can adopt.

6 What learning activity can we do with the Web?

Here is a (short) list of different kinds of computational learning environments in use. They represent different learning paradigms and can be classified along several axes like "Instruction - Learning", "External - Internal Control". Each are still appropriate for certain kinds of learning.

• Programmed Instruction (little step by step transfer of content)
• Computer Assisted Instruction (Drills & Tutorials)
• Intelligent Computer Assisted Instruction (ITS Tutorials)
• Computer Based Learning (Simulations, Hypertext & Microworlds)
• Intelligent Learning Environments (Microworlds + tutors, helpers, experts)
• Cognitive Learning Support Environments (some hypertexts)
3 How is courseware related to instruction?

Courseware engineering is concerned with electronic learning environments. Such an environment is a combined system involving tasks, agents, courseware products, etc. which is aimed at supporting learning processes and in which learning takes place mostly in interaction between learners, courseware products, other tools, and to a lesser degree tutors (human or artificial). Courseware is always a combination of elements (cf. de Dinan & van Shaik 93: 193), such as:

- textual material (including textual representations)
- simulation models,
- exercises
- problems
- feedback information, etc.

Each type of courseware architecture organizes those resources in various ways. How complete are the materials supplied to the learner with respect to the information and support needed by the learner to achieve an instructional goal? Several ways of tackling this problem have been proposed.

We shall briefly introduce some ideas from (1) Instructional Design and from (2) Advanced Learning Environment Research.

4 Instructional Design Theory: Sequencing & Chunking of Educational Material

- For background information, see: “Instructional Design Theory: Sequencing & Chunking of Educational Material” on page 15

Summary (bad!)

Instructional Design Theory provides detailed prescription on how to organize teaching and learning at the global (curricula), lesson and task level. Most work is also grounded in some learning theory. Despite and maybe also because of the level of details those approaches attempt to formalize instruction their practical use is often debated. Some argument against reading much instructional design theory is that a good teacher with good practice intuitively knows and uses things like Gagné’s steps.

Most people agree that instruction needs principles, however some researchers feel that instructional theory should not just be grounded in learning theory but BE applied learning theory and to implement optimal learning conditions according to what we know about learning. This is the way most research in Advanced Learning Systems operates.
Some Learning Theory Background

1. The question whether to fly with an Airbus at all, concerns more HCI but there is an interesting point concerning transfer of something learned: Would you drive a car without steering wheel (using sticks or something like that instead)?)

2. Some Learning Theory Background

Some basic misconceptions about learning have to avoided: Reading or seeing does not imply much learning. Even being able to recall knowledge does not mean being able to apply knowledge. Efficiency is not measured by mastery of the exercises and tests of a courseware tool, but my mastery of the task. Also be aware of more subtle knowledge transfer problems: Even “micro”-competencies such as operating an Airbus vs. a Boing or programming in C instead of Pascal are difficult to teach. If more abstract things like “programming” instead of just C programming have to be learned, special teaching strategies have to be used (such exercising skills in varying contexts. Lastly, the complexity of a learning environment must be adapted to the learners skills (Would you teach C as an introductory programming language)?

Summary

Learning appears to be a complex matter. No doubt that this is the reason why all the various branches of learning theory do not even view the problem from a same angle. However, all academic traditions do provide the learning environment designer with important key ideas:

1. Learning must take place within optional external “conditioning” (behaviorism)

2. Learning is related to active problem solving and involves integration, construction and compilation of new content (cognitivism)

3. Learning is constrained by human cognitive capacities (experimental psychology, HCI theory)
Teaching & Learning with Internet Tools
A Position Paper


Daniel Schneider, TECFA, FPSE, University of Geneva
schneide@divsun.unige.ch - http://tecfa.unige.ch/

Introduction

This position aims to encourage discussion between WWW providers of educational material and researchers in the field of educational technology. This paper is still a early draft and I am afraid it will retain that status for a while

Introductory reference material in the Appendix sections has awful very rough draft status. It is useful only as far as it points to some real reading. The table of Contents is in a separate file (by pure laziness) for the moment. Everything is straight Framemaker to html translation without modifications...

1 Introduction

What is known about human learning relevant for the design of educational material? How should this knowledge be used in the context of more specific educational goals and WWW-supported learning & teaching activities? Generally speaking there are two important statements about learning and instruction:

1. One learns by doing something (psychology)
2. One learns by pursuing an instructional goal (education sciences)

The learning environment designer must take into account both perspectives. WWW-based courseware must not restrict itself to delivery of educational content. It must be grounded in some model of instruction and learning. Many possibilities exist and haven proven to be effective. However, each paradigm works under certain conditions in certain situations using some set of specific educational technology. For example, it can easily be argued that a good book is better than hypertext version of that book (why do people always print out things?). Also, general rules can be formulated such as “learning without doing is pretty useless in most domains”. Consider the following questions:

1. Would you take the plane if you knew that the pilot has read all the documentation about flying and successfully passed examination testing
Table of Contents

Table of Contents 1
1 Introduction 3
Introduction 3
Some Learning Theory Background 4
Summary 4
How is courseware related to instruction? 5
Instructional Design Theory: Sequencing & Chunking of Educational Material 5
Summary (bad!) 5
Research on Advanced Learning Environments 6
Summary 6
What learning activity can we do with the Web? 6
Educational Hypermedia 7
Integration of Hypermedia 7
Courseware engineering & the learning environment 8
So what can we do with the Web? 8
Some Learning Theory Background 11
Types of Learning 11
Learning/Teaching Strategies & Principles 12
Instructional Design Theory: Sequencing & Chunking of Educational Material 15
Gagné’s steps of instruction 15
Reigeluth’s “Elaboration Theory of Instruction 15
Merill’s Component Display Theory (CDT) 16
Research on Advanced Learning Environments 17
Educational Hypermedia 23
Related Projects 23
Bibliography 25
Cited (well, some) 25
Learning Theory & Instruction Theory 26
Advanced Learning Environments 26
Hypertext & Hypermedia (totally uncomplete) 29
Textual Virtual Realities, Virtual Classrooms, CSCW 29
WWW 30
30