Alan Lesgold, Franz Schmalhofer (editors):

Expert- and Tutoring Systems as Media for Embodying and Sharing Knowledge

Dagstuhl-Seminar-Report; 94;
01.08.-05.08.94 (9431)

ISSN 0940-1121

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloss Dagstuhl
Universität des Saarlandes
Postfach 15 11 50
D-66041 Saarbrücken, Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

Expert Systems and Tutoring Systems as Media for Embodying and Sharing Knowledge

Alan Lesgold
University of Pittsburgh

Franz Schmalhofer
DFKI, Kaiserslautern, Germany

The goal of the seminar was to search for new commonalties in the fields of expert systems and intelligent tutoring systems. Since the academic research, the industrial developments and the practical applications have greatly matured the fields of expert systems and intelligent tutoring systems during the last decade, valuable insights could be achieved by pursuing such a comprehensive goal.

Originally, the two fields had relatively separate, although in many respects similar, goals and agendas. The task of developing expert systems was seen as encoding the knowledge and the competence of human experts. In a similar fashion, intelligent tutoring systems were intended to capture expert knowledge plus the added pedagogical expertise of the skilled teacher. However, the agenda of tutoring systems research first moved toward the special problems of modeling the not-yet-enlightened student and how human knowledge can be conveyed, while the work on expert systems focused on particular problems at the limits of human capability.

The presentations and discussions of the seminar focused on the knowledge that is shared between a teacher and a student, among practitioners of some field of expertise, and among various participants in complex and dynamic work and training situations in general. The seminar revealed that both expert- and instruction systems are improved by explicit representations of the objects and actors in work situations. In such situations, explanations play an important role, both in training and co-ordinated man-machine work.

While expert systems are mostly focused on the representations of the situations they are intended to address, tutoring systems are frequently tailored to the training for specific situations. As systems become applied to increasingly complex work and training situations, these differences are becoming less important. As both fields have matured, the shared focus has thus become more important and more achievable. With training emerging as an ever-higher cost for business and industry, its automation has become increasingly valued, as has the possibility of extending human capability with machine expertise. With a more unified understanding, these systems can thus be made more useful.

Session: Intelligent Tutoring Systems

Using Intelligent Tutoring Software Components for Job Modeling and Job Design

Alan Lesgold
University of Pittsburgh

Working with colleagues at the US Air Force Armstrong Laboratories, we have developed and made practical use of a job analysis technology which has been called PARI (for Precursor-Action-Results-Interpretation). In this technology, experts are first asked to pose problems to one another that they believe demonstrate the range of competences an expert in a job should have. Then, the sequence of actions taken by the expert solving each task is reviewed with him. A series of "stimulated recall" reviews probe for the expert's mental model of the situation under which each action was taken (its Precursor), the purpose of the Action itself, the expected Results, and an Interpretation of what was learned from by taking that action and noticing its results.

From this body of data, statements can be extracted about the process whereby experts represent tasks in their domain, their models of domain processes, and the goal structures that constitute their performance expertise.

From more recent work with other colleagues, notably David Hurley at Pittsburgh and Charles Bloom and Scott Wolff of US West Technologies, we know that it is possible to coach analysts as they convert the requirements statements for a software package into an object-based analysis and design. A coach that does this is being developed, which we call Sloop.

We now believe it is possible to combine the PARI approach and the Sloop approach to create a complete job analysis methodology that goes from initial expert interviews to object-based specification of the job environment and expert performance knowledge. The key is to see that just as software requirements refer to both the processes inside the software and the ways in which it will be used, job analyses refer to the processes inside the expert and the work environment in which that expertise is exercised. Plans for building a job analysis coach to reflect this approach are now being refined.

Implications of Case-Based Student Modeling to Intelligent Tutoring Systems

Gerhard Weber
University of Trier, Germany

The problem of student modeling in intelligent tutoring systems is often claimed to be intractable. This resulted in a shift from intelligent tutoring systems to more open learning environments. The question remains, whether student models can improve learning with these environments. To answer this question, we have developed a fairly elaborate episodic student model in the context of our LISP tutor. This student model implements a case-based reasoning approach to student modeling embedded in an elaborated help system to aid novices when learning LISP. This episodic student model (ELM) can be used advantageously to improve and individualize the cognitive diagnosis of program code and to find examples and so-called remindings.

Multi-Media Learning Environments and a Fuzzy Student Model

Shigeyoshi Watanabe
University of Electro-Communications, Chofu 182, Japan

Measuring the student knowledge state after concept learning in order to initially adapt a skill acquisition session according to a student's own necessities is a hard task. Typical approaches are the use of tests, or pre-defined initial parameters. The former is disrupting for learning and the latter too simple to deal with the broad possibilities faced. It is known that students show different behaviors during concept learning depending on the experience, background and actual understanding (the way a student is understanding a concept) during concept learning. Our approach is to classify the different behaviors through fuzzy propositions and link them with a student model through fuzzy rules to use in an expert system, and with it, select a suitable skill acquisition strategy. We apply this idea to a circuit analysis ITS where the concept learning session is carried out on a hypertext environment and the skill acquisition session on an interactive problem solving environment. By tracing the student's use of the hypertext environment we learn the student's behavior and use it as a premise in the fuzzy inference.

Session: Knowledge-Based System Technology

Problem Types and Reusable Problem Solving Components for the CommonKADS Library

Joost Breuker
University of Amsterdam

A typology of problems is presented that is used for indexing and accessing reusable problem solving components in a library that supports the CommonKADS methodology for building knowledge-based systems. Eight types of problems, such as planning, assessment etc., are distinguished, and their dependencies are explained. These dependencies suggest that the typology is to be viewed as a "suite" rather than the usual taxonomy of "generic tasks". Developing the suite has lead to some new insights and elaborations of Newell & Simon's (1972) theory for modeling problem solving.

- Tasks are distinguished from problem definitions. A task is constructed by finding and configuring problem solving methods (PSMs), which are suitable for solving the (well-) defined problem. Tasks and PSMs therefore have a one to one correspondence (O'Hara & Shadbolt, 1993), while there is a one to many correspondence between a problem definition (type) and PSMs.
- Three phases are proposed that turn spontaneous, ill-defined problems into well-defined ones, respectively into problem solving tasks.
- A complete *solution* consists of three components: a case model, an argument structure and a conclusion. The conclusion is a sub-part of both other components.
- Tasks (PSMs) package recurring chains of dependent types of problems in variable ways.
- The availability of behavioral models, or of structural/behavioral models in a domain determines to a large extent which types of problems can be posed and solved.

Representing, Sharing and Expanding Knowledge in Cooperative Hypermedia Environments

Manfred Thüring
empirica, Bonn, Germany

Recent developments in hypermedia, computer supported cooperative work (CSCW) and broadband networks open up new potentials for education and training. In many ways, these potentials resemble the ideas of three early hypertext pioneers: Vannevar Bush envisioned a device called *Memex* "in which an individual stores his books, records and communication, and which is mechanized so that it may be consulted with exceeding speed and flexibility." Ted Nelson proposed the *docuverse* which "is a structure in which the entire literature of the world is linked and forms a universal instantaneous publishing network." Doug Engelbart designed *NLS* "a computer-based environment containing ... documents, memos, notes and so forth but also supports planning, debugging and communication."

Combining these ideas and relating them to the field of education and training suggests a learning environment consisting of (a) individual workspaces equipped with sophisticated facilities for authoring and archiving (b) a global information space which can be accessed from the individual workspaces for retrieving as well as publishing multimedia information and (c) a broadband network with synchronous and asynchronous communication facilities for linking the individual workspaces. Such an environment could be provided by a "value added service" as it is currently discussed in research on "Intelligent Broadband Services and Networks".

The learning materials provided by such a service for education and training should take the form of hypermedia courseware offering a user interface especially designed to cope with the well known problems of hypertext readers, i.e., disorientation, lack of overview, insufficient comprehension of the relations between distinguished information units and difficulties in handling browsing and navigation.

In my talk, I described SPI - an interface which explicitly addresses these issues and offers a number of facilities to support orientation, comprehension and navigation (see Hannemann, J., Thüring, M., and Haake, J.M. ,1993, Hyperdocument presentation: Facing the interface. Arbeitspapiere der GMD Nr. 784. Sankt Augustin: GMD.).

SPI uses a static screen layout that displays structural information together with its corresponding content. For this purpose, it employs a combination of graphical browsers and content windows. To facilitate navigation, SPI reduces interaction overhead by supporting several convenient ways of moving through the document, such as clicking on nodes in browsers or using a tool called "Navigator" for global navigation.

Orientation is facilitated in SPI by indicating the reader's current position in the overall document structure and by visualizing options for moving back or further. Moreover, it is eased by the regular navigation semantics of the interface which maintains the structural and temporal context of the reader's current node.

Due to its tight coupling of different user interface components with a coherent hyperdocument structure it can contribute to reduce the readers' problems listed above.

Exploiting Context Information to Index into Reuse Libraries

Georg Klinker
Digital Equipment Corporation, USA

One of the challenges with providing a new solution for a workplace is to create descriptions of the solution components that the different groups that are involved with the development effort understand. Furthermore, the developers must compare descriptions of the components of the new solution with descriptions of previously defined solutions. If the descriptions are similar, the developers can reuse previously defined components to refine the components of the new solution.

This presentation introduces the Active Glossary. The Active Glossary is part of the Spark, Burn, FireFighter knowledge-engineering environment. It assists a development team with linking the descriptions of different components of a new solution to a vocabulary that all team members share. That is, the terms that comprise the common vocabulary are defined by their uses within a specific context. By exploiting this context information the Active Glossary can assist the development team with finding similar components.

Session: Facilitating Cooperation by Knowledge Media

Using Agent-Based Systems to Support Collaborative Design.

Michael E. Atwood
NYNEX Science & Technology, USA

The basic problem with system development is communication. It is not, however, communication between software modules or between hardware and software systems that is the problem; rather, the problem is communication between people. Developing a system, whether it is an expert system, intelligent tutoring system, an interactive system or a physical system requires the expertise of many people. While each person has unique knowledge relevant to a system development, all have in common a lack of knowledge about some aspect relevant to development. This situation reflects, as Rittel describes it, a "symmetry of ignorance".

Mutual education is needed if all the people involved in a development effort are to be able to communicate effectively. We propose "design intent" as a mechanism for this communication. All those involved in a development effort collectively agree how the to-be-developed system will interact with its users and the more global environment in which it will be fielded and simultaneously document this agreement. System specifications are then derived from this design intent. To facilitate continued discussion, the design intent is embedded in the developed system as "expectation agents". These agents monitor the system in use and are triggered when expectations are not met. These breakdowns provide an opportunity either to refine the system requirements or to educate the system users on how better to use the system.

Expert Systems as Intelligent Documentation and Communication Tools

Franz Schmalhofer
DFKI, Kaiserslautern, Germany

Expert systems that are developed by model-based knowledge engineering techniques are well suited for the type of situations for which they are designed for. However, when unexpected innovations occur in the specific field of application, such systems become obsolete. Intelligent documentation systems which allow for a situated application of the represented knowledge as well as for knowledge updates (knowledge base evolution) were proposed as a means for meeting such challenges of complex real life domains. The formation of suitable domain-related abstractions is seen as the core of such systems. A machine learning procedure for forming these abstractions was presented and its utilization in various application domains (mechanical engineering, medicine) was discussed.

Agent-Oriented Interaction in Cooperative Computer-Based Training Systems

Guy A. Boy
EURISCO, Toulouse Cedex, France

Computer-based training (CBT) has become a major field of investigation. However, authoring and modifiability of current CBT systems remain very open issues. On the software side, object-oriented programming allows to create and maintain libraries of reusable objects. On the cognitive side, even if these objects are designed as artifact metaphors, they remain passive and rarely include user knowledge. The aim of this paper is to introduce a new view of objects (called agents) that provide a continuity between real life artifacts and artificially created artifacts. In particular, they can be adaptive, and context-sensitive. They are intended to facilitate learning-by-doing. The design of such agents is based on Schank's learning architectures as well as other important training-relevant needs such as evaluation, instructor aids, cooperation and networking. Examples illustrate the applicability of these new concepts, and a discussion is started.

Session: Knowledge Sharing and Reuse

The Development of an Intelligent Tutoring System for Training Customer Contact Personnel

Charles P. Bloom
US WEST Technologies, USA

The Learn, Explore and Practice (LEAP) intelligent tutoring systems (ITS) platform represents the product of a two-plus year effort in technology research, development and transfer. LEAP is a general, multimedia ITS platform that provides customer contact employees (CCEs) with an intelligent, coached environment in which to practice customer contacts (i.e., conversing with customers to solve problems and sell products and services, while simultaneously interacting with a number of main-frame software applications). In LEAP, trainees can exercise their customer interaction skills by working through typical customer interactions in a tutoring environment that accurately emulates their job environment to facilitate transfer from training to work. As trainees work through contact scenarios, LEAP applies instructional strategies as appropriate for each individual trainee. In addition, trainees can request advice from LEAP at any time, and can review their performance after finishing each scenario. In addition, LEAP allows instructional designers to adjust LEAP's instructional and student modeling parameters to further individualize the delivered instruction.

Version 1 of LEAP underwent a comprehensive evaluation in the summer of 1993, the results of which have driven significant revisions to the system in order to meet a U S WEST corporate expectation that LEAP be deployed in the fall of 1994 to support training of all customer contact employees. In particular, it was felt that three primary areas in version 1 needed enhancement:

(1) LEAP used a dialogue management method that restricted all contacts to a single correct path (job analysis results indicated that there is significant flexibility on the ordering of some dialogue activities).
(2) LEAP only simulated a single application for CCEs to interaction with (CCEs often interact with as many as three or four applications simultaneously).
(3) LEAP's reflective follow-up functionality did not allow trainees to inform LEAP on how they thought they did, nor did it provide contact specific feedback.

Revisions included in Version 2 in response to these findings include:

(1) Increasing LEAP's dialogue flexibility to better support cross training and refresher training of more experienced personnel.
(2) Supporting the simultaneous presentation and interaction with multiple application simulations.
(3) Enhancing LEAP's reflective follow-up functionality to provide more instructional review options at the completion of each contact rehearsal.
(4) Conducting of traditional software engineering practices to produce a more robust and easier to maintain system, eventually to be transferred to a deployment and support organization.

An Interactive Learning Environment Using Focusing on and Perturbation of Media Representations

Kohji Itoh
Science University of Tokyo, Noda 278, Japan

According to J. Barwise and J. Etchemendy, "homomorphic representations" are representations in which the semantic structure of the represented world could be more or less directly observed in the syntactic structure of the representation. The use of such representations, e.g. diagrams, along with linguistic representations is ubiquitous (also called heterogeneous) in human reasoning.

It is argued that the increase of efficiency in reasoning comes from the ease of sensory motor operations (or their stimulation) of the representation by human problem-solvers, as well as from the often observed explicitness of (sub)goals in (sub)problem descriptions.

Interactive learning environment systems are proposed to provide students with building blocks which are used to construct and manipulate homomorphic and heterogeneous representations of their problems in collaboration and "agreement" with the system. They also endow the system with such abilities as numerically focusing and manipulating the same representation. Thereby, the invariants, cases, or conflicts are discovered that are to be suggested to the students.

An algorithm has been developed which enables students as well as the system to perturb their representation for preserving the already satisfied constraints in order to satisfy additional constraints or to discover invariants and cases.

The environment has been implemented as an object-oriented knowledge-based system. For illustration, sample dialogue sessions that could be held between the students and the system are used. Such prototype learning environments are currently under development in several domains (e.g. elementary geometry, electromagnetics, electrical circuits, and dynamics).

Integrating General Information and Knowledge with Specific Experiences in an Environment for Teaching and Learning Assistance

Agnar Aamodt
University of Trondheim, Norway

The basic idea of this work is to combine a knowledge-based approach, based on a *semantic network* representation of knowledge, with a *hyperlink network* for student/teacher browsing of concepts, relationships and submodels. The two networks represent two *views* of the same basic data structure - a densely connected network of objects and relations. Hence, a platform for knowledge sharing is established, since the same data structure can be interpreted and utilized as semantic network knowledge for the system and as hypertext information for the user. Knowledge (for example partial domain models) and information may therefore be shared between computer and users, given that the computer contains inference methods that are able to interpret the network structure in a similar fashion as a human user. Correspondingly, knowledge may also be shared between several users, e.g. teachers and students.

Upon this basis, we are studying methods for knowledge-intensive *case-based reasoning* for computer-aided instruction. The more general research agenda that lies behind this work is the reuse of experience for intelligent decision support in open and weak theory domains. Learning from experience is an emphasized issue, since it is hard to see how future user-cooperative systems - that have to deal with increasingly complex and continuously changing application domains, can do without adaptive learning abilities. A particular topic is how a model of general knowledge can be used to focus the reuse of past experiences (i.e. cases). Another topic is how a manual utilization of past concrete experiences can be combined with automated case-based reasoning and learning. Finally, the problem of learning in the 'system as a whole', i.e. student learning as well as machine learning, is discussed within the above context. These three topics should be viewed as an approach to *integration* along the three dimensions, defined by the three pair of end points: specific cases / generalized knowledge, manual problem solving / automated reasoning, learning in student / learning in system.

To represent network knowledge, we use a frame-based knowledge representation system (CreekL) with default inheritance, with self-descriptive (reflective) properties, and in which relations (slots) are also represented as concepts (frames) to be explicitly modeled and reasoned about. Integrated use of different knowledge types is at the core of our methodological approach, and we are continuously looking for synergy effects within, as well as between, the three dimensions. The integration view has consequences for each of the *separate methods* that underlie the sharing of knowledge across humans and machines. For example, how the case indexing problem is solved when there exists a body of general knowledge and information in which the cases are integrated, and how the case reuse methods are affected by a combined manual and automated reuse of cases.

Our first goal is to develop a framework and a high level system architecture that is suitable for describing the relevant properties of these dimensions, and for analyzing their interrelations. At the basis of this framework is a distinction between *information* and *knowledge*, according to their different roles in a learning process (or a decision process in general), and whether the point of reference is the human user or a machine. The framework will, in turn, form the basis for a system design in which case-based reasoning and learning is the 'running engine' in an environment for teaching and learning assistance, and in which also more data-intensive and knowledge-poor methods will be studied and compared.

Discussion Group: The Role of Case-Based Reasoning in Intelligent Tutoring Systems

Introduction

Agnar Aamodt and Peter Reimann
University of Trondheim, Norway, University of Freiburg, Germany

We find three important reasons why cases and CBR techniques begin to play an increasing role in instructional systems, ITS in particular. The first reason is a psychological one: Students - novices and beginners, but also often more advanced learners - make frequently and intensively use of case information, for instance in form of examples (worked out solutions to problems) by referring back to former problem solving episodes produced by themselves. CBR serves them both as a problem solving method and as a form of analogy-based learning: Repeated use of cases may lead to generalized knowledge structures. The second reason is a pedagogical one: Reasoning with cases is in many areas - in particular those taught at the university level - an established way of teaching: Think of the Harvard style of teaching law and business. The third reason is a pragmatic one: In those domains which are hard to formalize - such as law - or where using the 'deep' causal knowledge is too cumbersome for most practical purposes - such as medicine -, teaching with and learning from cases is often a practical way to provide computer-based instruction.

Looking at the current practice of using cases in computer-based instruction, we can distinguish four types of applications:

1. Diagnosis of students' knowledge. A good instance for this type of application is Gerhard Weber's ELM system, that is able to come up with a planning rationale for students (LISP-) programming behavior by using a knowledge base of programming knowledge (formalized as rules). Since operator-based plan recognition is a cumbersome and computationally expensive process, ELM stores the plan structure in an episodic memory.

   When attempting further plan recognition, ELM will first look into its case memory whether it hasn't analyzed similar errors before and if so, whether the former explanation could not also account for the current programming mistake. This lookup in the case library is computationally much cheaper than full plan recognition, and has the added advantage that tutorial advice to the students can be based on analogical remindings to former mistakes they may have made.

2. Conveying normative knowledge. Here, cases are used to tutor normatively correct knowledge. A good example are the ASK systems developed at Northwestern University and Bloch & Farrell's early DECIDER program. Cases are carefully selected and/or constructed in order to demonstrate to the students prototypical correct and incorrect behavior. Usually, presentation of cases is integrated into a more encompassing learning environment that forces the student to become actively involved into a decision making or problem solving activity. This is important for both the CBR component, since the case indexes are tuned towards a specific application context, and for the human student, so that she can mentally index case information adequately.

3. Capturing and structuring students' problem solving experience. In this instructional scenario, students construct their own cases by keeping more or less structured records of their learning and problem solving experience, records that can be stored into and retrieved from a case memory. This use of CBR techniques is particularly useful to support exploratory styles of learning. For instance, Thomas Schults CABAT program keeps track of the experiments and predictions students conduct in a computer-simulated physics laboratory, indexes them into a case-memory, and retrieves them at appropriate times to remind students of similar experiments and mistakes they have done before. In a similar vein, Reimann, Schult & Bellers SeeChess program allows a student to keep records of what she considers important about example solutions (for chess endgames) and retrieves these cases in later problem solving contexts.

4. Instructional planning and design. Programs such as pioneered by Kolodner's SCI-ED use case knowledge in order to design instructional settings (for instance, laboratory experiments) and plan instructional content presentation (for instance, an hour in a school class). In this application context, there is no direct interaction with the students and the CBR system: Design and planning is done off-line. Furthermore, the actual interaction with the student is executed by a human teacher, not by the system. CBR programs play more the role of an advisory system here than of an expert planner or teacher. For instance, the CBR system may retrieve from its case library a number of solutions for an instructional planning task, and it will be left to the teacher to select one to develop further.

Issues Raised During the Discussion

One point that became clear was that for instructional purposes, cases will usually need to be carefully assessed, processed and indexed. That is, case representations will not be only the 'raw' data, the episode as it took place in the world, combined with a couple of surface-feature-based indexes. Rather, cases representations need to be selective, partially abstracted, and flexibly indexed. One reason for this is that a case representation in an instructional context needs not only be understood by experts, but by novices with various degrees of general knowledge about the domain, and with various learning styles. Hence, 'case acquisition' becomes important - and becomes also a potential problem for system development, in analogy to the knowledge acquisition bottleneck in standard expert systems. A related notion is that cases will often be used as examples (both positive and negative ones), hence have to be constructed/selected in order to point out a typical good or bad move.

Another issue raised was that developers of case-based instructional systems must be clear about their overall goal: Are students supposed to induce generalizations from cases and form abstractions, or is the intended knowledge structure an enriched mental case memory? For instance, in teaching law to American students, the core knowledge structure students acquire is a mental case library. Little is done to advance their knowledge about the general characteristics of the law. Clearly, in other areas cases are used more in the sense of instances with the expectation that students will induce from cases generalizations and will use case-specific knowledge only to the degree that it adds to generalizations relevant additional information. Making clear to students what is expected from them right at the beginning is important, since human learners are quite adaptive to perceived task demands.

There was also agreement among the participants of the discussion that teaching in general must be more than presenting cases. This is even the case in areas where case-based reasoning is the predominant way of solving problems. Since we want students to acquire case knowledge that can be used flexibly, i.e., transferred to more than merely superficially similar situations, and can be used correctly, i.e., be modified according to new task demands, cases need to be indexed mentally in terms of not only domain and situation specific, but also abstract indexes, indexes that allow for 'far' transfer. In order to index cases in such a manner, one needs a potentially large amount of general background knowledge and domain specific knowledge captured in other than episodic form. Hence, the relationship between case-based reasoning and other reasoning methods must be clearly analyzed and the usefulness of hybrid approaches in instructional settings must be determined.

Both, in order to teach case-based reasoning in a domain and to teach case-based, one needs a detailed understanding of how cases are used by humans in the respective domains. It was mentioned in the discussion that for many areas, in particular for professional fields, we do not have such an understanding yet and that it would be helpful in general if more systematic studies of the actual construction and use of cases in diverse areas would be conducted.

An issue that was raised but time did not allow for deeper discussion was how to deal with cases that result from collaborative problem solving and how to adapt case-based instruction to collaborative teaching scenarios.

Multiple Student Modeling in Group Learning Environments

H. Ulrich Hoppe and Mitsuru Ikeda
GMD-IPSI, Darmstadt, Germany, Osaka University, Japan

Mainstream ITS research has recently been criticized for its neglect of supporting human-human interaction and social learning (as opposed to individualized instruction). Given the current state of networking and distributed computing, there are no longer practical or technical reasons for intelligent learning support being limited to the individual case. Though learner modeling for groups of learners (and possibly a teacher) poses new problems, the availability of human support and e.g. peer-to-peer cooperation can also help to avoid existing problems in generating adequate, personally meaningful, feedback to learners. Generally speaking, intelligent subsystems may help in the tasks of knowledge assessment and error diagnosis (Hoppe, H.U. (1994). Deductive error diagnosis and inductive error generalization. *Journal of AI in Education*, **5** (1), pp. 27-49), whereas the actual tutoring maybe left to human-human interaction of different types. Of course, also the cooperation between humans should be technically facilitated.

First, different group learning situations have to be distinguished and their specific requirements have to be formulated. Examples are teacher-centered classroom situations or unmonitored group learning in which several students have individual access to interactive learning environments. Whereas in the presence of a teacher or human tutor, individual problem solving phases (exercises) may only be monitored and analyzed locally and feedback may be given directly to the teacher, unmonitored situations require an integration of the multiple student models to infer the adequateness of cooperation between certain individuals. Of course this integrated use of multiple student models is more demanding and interesting.

Taking a more theoretical view on multiple student modeling, different types of student models can be analyzed with respect to their "additivity" (i.e. their formal characteristics for integration). This corresponds to the general problem of knowledge fusion from various, potentially overlapping, knowledge sources. Overlay models are a simple case, since here we can ideally assume complete "additivity" of the individual portions of knowledge. If "buggy versions" are associated to correct rules, things become more complicated but there are still some integration strategies that appear to be tractable. The most challenging variant of the problems arises for student models that are completely synthesized from examples (as e.g. in the FITS/THEMIS framework; Ikeda, M., Kono, Y., and Mizoguchi, R. (1993). Nonmonotonic model inference - A formalization of student modeling. *Proceedings of IJCAI '93*, Chambery (France). pp. 467-473). Here it is no longer clear which portion of student A's knowledge corresponds to which part of student B's knowledge, and even if such correspondences can be established, the knowledge may still be contradictory.

As a first approximation to these problems, the different approaches and their difficulties will be analyzed. However, on this basis we can already formulate research and implementation strategies.

Session: The Embodiment of Medical Knowledge

Intelligent Documentation and Decision Support for the Evaluation of Adverse Events in Clinical Trials

Heiner Gertzen
Hoechst AG, Frankfurt

The efficacy and safety of a new drug must be investigated in a series of clinical trials before it can be released. In order to guarantee the safety of the drug and estimate its risk-benefit ratio, adverse events which occur in the clinical trials must be carefully evaluated using all the knowledge and experience available at that time. In particular, it must be decided whether a reported adverse event is related to the experimental drug or due to other causes (e.g., other concomitant drugs or diseases).

An intelligent documentation and decision support system was designed to support the medical expert in this difficult evaluation and decision task. The system comprises three main components: 1.) an intelligent documentation component ; 2.) a decision support component; and 3.) a hypertext based interface.

A newly reported adverse event is processed by the intelligent documentation component. Subsets of biomedical knowledge and previously solved cases are retrieved, processed and applied to the case at hand, yielding an enriched case description. This is processed by the decision support component to reach a decision about whether the event is related to the drug under study, together with an explanation of the adverse event. At present, the algorithm actually used by medical drug safety experts is applied in this step.

A pilot system has been implemented in a Windows environment using MS Excel. Currently, knowledge acquisition techniques are used to enhance the biomedical knowledge base/intelligent documentation component, and case-based reasoning techniques are adapted to the needs of the evaluation task. We consider to incorporate additional decision rules into the system comprising phased decision strategies, heuristic decision models and connectionistic models.

Textbooks/Expert Critiquing Systems for Medical Education

Frank Puppe
University of Würzburg, Germany

Humans learn best by doing. The computer can provide an attractive learning tool by simulating the problem environment, where the student can solve cases. Our concrete scenario is as follows:

1. The student learns the basics of the domain in a conventional manner (e.g. with textbooks).
2. The student can study formalized expert knowledge of the domain.
3. The teacher presents (real) cases to the students and discusses them with respect to the material the student should know.
4. The tutor system presents the student cases to be solved in a manner as realistic as possible, i.e. audio-visual and stepwise presentation of data.
5. While solving the case, the student - if not in examination mode - has access to the textbook and formalized expert knowledge, preferably by an integrated hypertext system.
6. The student is not interrupted when doing something wrong, e.g. not recognizing correctly an audio-visual presented symptom or ordering an unnecessary test, unless asking for comments.
7. The tutor system can follow the student's actions and finally criticizes suboptimal performance.

The methodology necessary for this scenario is combining an expert system shell for building the knowledge base and for problem solving capability, a case presentation interface and a hypertext system. We present three example-applications in an advanced stage within our D3-framework in the domains of rheumatology, neurology and ECG-interpretation.

Parallel Computer Architectures for Artificial Intelligence and Knowledge Processing

Ivan Plander
Slovak Academy of Sciences, Slovakia

The paper provides a survey on the applications of massively parallel computer architectures in artificial intelligence (AI) and knowledge processing, discusses the most important areas of AI where the applications of massively parallel computers provide the greatest advantage and talks objectively about massively parallel SIMD architectures as the prerequisites to master truly massive parallelism represented by neural, optical, molecular and other massively parallel computers. The paper presents the main characteristics of massively parallel computers and defines the true parallelism where the number of processing elements is so large that it may conveniently be considered a continuous quantity. The most effective applications of massively parallel computers presented are:

- SIMD parallel heuristic search as a fundamental problem solving method in AI.
- Implementation of parallel logic programming languages on SIMD massively parallel computers. For some applications with large databases, the theoretical speed-up is of the order of 1000.
- Associative processing of very large databases and knowledge bases. Associative processing architectures are promising candidates for second generation massively parallel computers, offering the potential for application flexibility (scalability and programmability).
- Massively parallel computers provide a tremendous increase of computing speed in image processing. The massive parallelism was used e.g. for the automatic determination of depth from stereo imagery, image segmentation in data compression, Fourier transforms, the discrete cosine algorithms for image enhancement and edge detection.
- Simulation of neural nets on massively parallel architectures is performed in such a way that the conventional algorithm concept is replaced by that of training and learning a machine. Implementation of neural nets on large SIMD computers has the advantage of immediate availability of the hardware and easy reconfiguration of the net in software.
- Implementation of rule-based systems for real-time processing on specialized massively parallel SIMD architectures using special-purpose inference processors and associative parallel memories is perspectiveful. The simulation of a parallel inference processor on a SIMD-type parallel computer is advantageous in the possibility to study the rule set processing process in a bit-serial word-parallel manner.

AI is moving into a new phase characterized by a broadened understanding of the nature of knowledge, and by the use of new computational paradigms. A sign of this transition is the growing interest in massively and truly massively parallel computers, represented by neural, optical, molecular and other massively parallel analog computers.

Session: The Integration of Expert and Tutoring Systems

The Segmentation of Work with Expert and Intelligent Training Systems

Valerie L. Shalin
State University of New York at Buffalo

The functions of intelligent tutoring systems are designed to enhance learning. The function of expert systems, and workplace aids in general, are designed to improve user performance, often measured in the achievement of system goals with minimal resource expenditure. In complex work settings, the functions of intelligent tutoring systems may not serve as useful workplace aids for two reasons. First, the task decomposition represented in the expert model may isolate individual skills for pedagogical reasons, while in practice the skill interacts with a much richer context of goals and opportunities. In such contexts, the expert models within an intelligent tutor may in fact be quite impoverished, and provide little ultimate benefit to performance. Second, the functions of performance evaluation within an intelligent tutor often address blatant procedural errors, which occur less often and with less ultimate cost to system success than daily ill-informed decision making and problem prioritization. A promising and feasible approach to expert aiding is to complement rather than duplicate the available human expertise with knowledge-based aids that represent useful information and knowledge from related parts of the work system, to support self-evaluation and improvement according to system-oriented goals and values. More ambitious approaches to aiding depend on broader models of expertise in complex systems, including processes of problem identification and prioritization, and expert processes for deciding resource tradeoffs, conducting self-evaluation and correction. Such models may ultimately enrich tutoring goals as well, making expert systems and tutoring systems more similar.

Discussion Group: Corporate Memory

Introduction to Corporate Memory

Gerhard Strube
University of Freiburg, Germany

Corporate Memory (CM) is an initiative to help companies cope with the problems of managing information that is useful and important for the company's operations. Technically, CM strives to integrate existing technologies of databases, information systems, knowledge-based systems, and also technologies currently under rapid development, like multimedia, hypertext, and electronic documentation. The importance of organizational aspects, and hence, of methods of workflow, information flow, and job analysis was highlighted. CM was discussed in parallel to human memory, which is also not a passive receptacle, but an active system for task-oriented and context-driven retrieval. Finally, costs and benefits were discussed, as well as feasibility.

Example for an Application in Aerospace

Markus Durstewitz
EURISCO, Toulouse Cedex, France

We understand corporate memory (CM) as an assistance for knowledge sharing within a corporation. Especially, in highly interconnected domains such as aerospace industry it becomes necessary to provide people undertaking a task with complementary knowledge about other sectors. Knowledge acquisition and representation is based on process (task), product, and operator models. The CM encounters three parts of knowledge:

(1) normative knowledge (that already exists in form of standards and norms),
(2) procedural knowledge (procedures as the set of activities of the process),
(3) episodic knowledge (experience, facts).

The use of the knowledge determines the functionalities and tools to be realized:

(1) intelligent reference management;
(2) adaptive interfaces;
(3) case bases.

Session: Networks and Multimedia

Hypercomposition & Instructional Explanations

Baruch B. Schwartz
Hebrew University, Jerusalem, Israel

Research in teaching has shown that instructional explanations - the contributions of teachers and texts to learning - are complex goal states that demand high-level skills to be achieved. Important differences have been detected between experts and novices. For example, the identification of the problem (which elucidates the purpose of the explanation); or knowing the representational system (i.e., the set of devices that are communicative for the specific audience and on which explanations will be grounded; or finally completing verbal explanations to bridge between the representational system and the principles to be explained). The study that is reported here analyzes how the use of a hypercomposition tool could help the teacher articulate instructional explanations. It is shown that multiple-layered explanations turn to be feasible, a fact which suggests that instructional explanations with hypercomposition tools may lead to metacognitive learning. In addition, I show that most of the generally difficult kinds of instructional explanations were articulated during lessons in history and in mathematical problem-solving.

Problem Solving and Hypothesis-Testing with Intelligent Tutoring Systems

Claus Möbus
University of Oldenburg, Germany

The talk has five main topics. It will be shown that (1) a psychological theory of knowledge acquisition is necessary to derive some (2) design principles for intelligent tutoring systems. One of the design principles is that the system is sufficiently knowledgeable to (3) test hypotheses: students' solution proposals. It is discussed (4) in case studies how the hypothesis testing approach can be realized in three domains within three intelligent problem solving environments:

- in room configuration tasks (IKEA)
- in function programming (ABSYNT)
- in modeling distributed time discrete systems (PETRI-HELP)

It is shown how the same functionality especially concerning hypothesis testing can be achieved despite differences in the domains. These differences make it necessary to use very different artificial intelligence techniques:

- constraint-based search (IKEA)
- parametrized AND/OR-trees (ABSYNT)
- temporal logic, model-checking and machine-learning (PETRI-HELP)

In the last part of the talk (5) an epistemological motivated overview is given how students problem solving and hypothesis testing fit into a more general frame of theory revision. It is argued that learning in the sense of theory revision occurs in impasse situations where the student gets system-generated feedback to his hypothesis. Learning occurs by self-explaining the feedback contents on the basis of student heuristics and repairs. Thus it is not expected that expert knowledge is directly implanted. This is in accordance to cognitive science and cognitive psychology based research.

The Intelligent Learning Environment with Multimedia under Networking

Toshio Okamoto
University of Electro-Communications, Tokyo, Japan

The paradigm of educational computing has changed from architectures with mixed initiatives like conventional ITSs (Intelligent Tutoring Systems) to educational software which prompts and supports a student's active learning, i.e. interactive learning environments and open-ended learning goals. Such learning environments form Micro Worlds, where the environment reacts susceptibly to a student's activity and reminds the students of particular issues. It means that the environment itself reflects its process to student-thinking as a mirror. One educational point of view states that the students would thereby acquire the ability of metacognition with which the students cannot only learn the knowledge about the specific domain but also monitor the process of self-cognition. The process needs the learning cycle of hypothesis-testing and emphasizes its thinking process. The importance of situated learning is pointed out on the basis of this learning paradigm. That is, we should incorporate the circumstances of problem solving according to the situation, the scenario and further the function of something like role-playing into the system. The intelligent multimedia system of "macro economics" domain described in this paper has been studied and developed from these points of view.

In the macroeconomics system, the students can play the role of an agent in "macro economics world" on the basis of socially situated learning by using the function of something like a game/simulation under a network environment. The system incorporates an expert system representing the virtually smart student and the chairman-expert system which controls the whole system. The idea is derived from the concept under which the computer companion should be embedded in the system in order to support learning by observation (modeling learning).

The students have an imitative life-experience by the intelligent simulation system with multimedia on the recovering and growing process of Japanese economics until the present age after World War II. There, they try to play the game of social political situations for each age from an economic viewpoint. It seems that the students can learn the principle of macro economics. The domain contents of the system are on the Japanese economics and its contemporary history. The learning environment is suited for industry people, university students and high school students. The system is built with multimedia technology (pictures, sound, animations, graphics, and text) including an electronic dictionary. The simulator of the model of macroeconomics and the expert systems with the rule-base of the characteristic information are employed to realize the virtual environment of macroeconomics.

Dagstuhl-Seminar 9431

List of Participants

Agnar **Aamodt**
University of Trondheim
Department of Informatics
N-7055 Dragvoll
Norway
agnar.aamodt@ifi.unit.no
tel.: +47-73-59 18 38

Stuart **Aitken**
Deutsches Forschungszentrum
für Künstliche Intelligenz
Erwin-Schrödinger-Straße
67608 Kaiserslautern
Germany
aitken@dfki.uni-kl.de
tel.: +49-631-205-34 39

Michael **Atwood**
NYNEX Science & Technology
500 Westchester Ave.
White Plains NY 10604
USA
atwood@nynexst.com
tel.: +1-914-644-25 82

Simone **Bürsner**
Universität Heidelberg
Inst. für Medizinische Biometrie
u. Informatik
Im Neuenheimer Feld 400
69120 Heidelberg
Germany
sbuersner@krzmail.krz.uni-heidelberg.de
tel.: +49-6221-56 74 86

Charles **Bloom**
U S WEST
Advanced Technologies Suite 270
Applied Research and Multimedia Services
4001 Discovery Drive
Boulder CO 80303
USA
cbloom@advtech.uswest.com
tel.: +1-303/541-6411

Guy **Boy**
European Institute of Cognitive
Sciences and Engineering
10 avenue Edouard Belin
31055 Toulouse Cedex
France
boy@tls-cs.cert.fr
tel.: +33.62.17.83.11

Joost **Breuker**
Universiteit van Amsterdam
Dep. of Social Science Informatics (SWI)
Roetersstraat 15
NL-1018 WB Amsterdam
The Netherlands
breuker@swi.psy.uva.nl
tel.: +31- 20 525-6798/6789

Markus **Durstewitz**
European Institute of Cognitive
Sciences and Engineering
10 avenue Edouard Belin
31055 Toulouse Cedex
France
durstewitz@tls-cs.cert.fr
tel.: +33.62.17.83.11

Ludger **van Elst**
Deutsches Forschungszentrum
für Künstliche Intelligenz
Erwin-Schrödinger-Straße
67608 Kaiserslautern
Germany

Heiner **Gertzen**
Hoechst AG
Clinical Research H840
Corporate Clinical Quality Assurance
Brüningstr. 50
65926 Frankfurt
Germany
gertzen@msmkf1.hoechst.hoechst-ag.dbp.de
tel.: +49-69-305-49 51

H. Ulrich **Hoppe**
GMD-IPSI
Dolivostr. 15
64293 Darmstadt
Germany
hoppe@gmd.darmstadt.de
tel.: +49-6151-869-8 66

Mitsuru **Ikeda**
Osaka University
Institute of Scietific & Indus. Research
AI Research Group
8-1 Mihogaoka
Ibaraki Osaka 567
Japan
ikeda@ei.sanken.osaka-u.ac.jp

Kohji **Itoh**
Science University of Tokyo
Faculty of Industrial Science & Technology
Dept. of Applied Electronics
Noda 278
Japan
itoh@te.noda.sut.ac.jp
tel.: +81-471-24-15 01

Georg **Klinker**
Digital Equipment Corporation
MRO1-3/J14
200 Forest St.
Marlboro Ma 01752
USA
klinker@guess.enet.dec.com
tel.: +1-508- 467-2482

Alan **Lesgold**
University of Pittsburgh
LRDC
3939 O'Hara Street
Pittsburgh PA 15260
USA
al+@pitt.edu
tel.: +1-412-624-7046

Claus **Moebus**
Universität Oldenburg
FB 10 - Informatik
Ammerländer Heerstr. 114
D-26111 Oldenburg
Germany
Claus.Moebus@arbi.informatik.uni-oldenburg.de
tel.: +49-441-798-2900

Toshio **Okamoto**
University of Electro-Communications
Graduate School of Information Systems
1-5-1 Chofugaoka
Chofu-City
Tokyo 182
Japan
okamoto@ai.is.uec.ac.jp
tel.: +81-424-83-21 61

Ivan **Plander**
Slovak Academy of Sciences
Institute of Computer Systems
Computing Center
Dubravska cesta 9
842 37 Bratislava
Slovak Republic
upsyplan@savba.sk
tel.: +42(7)37 1004

Frank **Puppe**
Universität Würzburg
Lehrstuhl für Informatik VI
Am Hubland
97074 Würzburg
Germany
puppe@informatik.uni-wuerzburg.de
tel.: +49-931-705-61 10

Peter **Reimann**
Universität Freiburg
Psychologisches Institut
AllgemeinPsychologie
Niemenstr. 10
79085 Freiburg
Germany
reimann@psychologie.uni-freiburg.de
tel.: +49-761-203-24 82

Franz **Schmalhofer**
Deutsches Forschungszentrum
für Künstliche Intelligenz
Erwin-Schrödinger-Straße
67608 Kaiserslautern
Germany
schmalho@dfki.uni-kl.de
tel.: +49-631-205-34 65

Baruch **Schwarz**
The Hebrew University of Jerusalem
Computing & Computer Science Group
School of Education
Mount Scopus
91905 Jerusalem
Israel
msschwar@pluto.mscc.huji.ac.il
tel.: +972-2-881064

Valerie **Shalin**
SUNY at Buffalo
Dept. of Industrial Engineering
342 Bell Hall
Buffalo NY 14260
USA
valerie@valerie.eng.buffalo.edu
tel.: +1-716-645-23 57

Gerhard **Strube**
Universität Freiburg
Institut für Informatik u. Gesellschaft
Abteilung Kognitinswissenschaft
Friedrichstr. 50
79098 Freiburg
Germany
strube@cognition.iig.uni-freiburg.de
tel.: + 49 761 203 4933/34

Manfred **Thuering**
Empirica
Communications & Technology Research
Oxfordstr. 2
D-53111 Bonn
Germany
manfred@emp-d.uucp
tel.: +49-228-9 85 30-39

Bidjan **Tschaitschian**
Deutsches Forschungszentrum
für Künstliche Intelligenz
Erwin-Schrödinger-Straße
67608 Kaiserslautern
Germany
tschaits@dfki.uni-kl.de
tel.: +49-631-205-34 64

Shigeyoshi **Watanabe**
University of Electro-Communications
Dept. of Communications & Systems Eng.
1-5-1 Chofugaoka
Chofu-City
Tokyo 182
Japan
watanabe@sw.cas.uec.ac.jp
tel.: +81-424-83-21 61

Gerhard **Weber**
Universität Trier
Fachbereich I Psychologie
Universitätsring 15
54286 Trier
Germany
weber@cogpsy.uni-trier.de
tel.: +1-651-201-29 56

Zuletzt erschienene und geplante Titel:

B. Endres-Niggemeyer, J. Hobbs, K. Sparck Jones (editors):
Summarizing Text for Intelligent Communication, Dagstuhl-Seminar-Report; 79; 13.12.-17.12.93 (9350)

Ch. Brink, G. Schmidt (editors):
Relational Methods in Computer Science, Dagstuhl-Seminar-Report; 80; 17.01.-21.01.94 (9403)

A. Arnold, H. Seidl, B. Steffen (editors):
Algorithms in Automata Theory, Dagstuhl-Seminar-Report; 81; 07.02.-11.02.94 (9406)

K. Ambos-Spies, S. Homer, U. Schöning (editors):
Structure and Complexity, Dagstuhl-Seminar-Report; 82; 14.02.-18.02.94 (9407)

H. Noltemeier, T. Ottmann, D. Wood (editors):
Data Structures , Dagstuhl-Seminar-Report; 83; 28.02.-04.03.94 (9409)

J. P. Finance, S. Jähnichen, J. Loeckx, D. Smith, M. Wirsing (editors):
Logical Theory for Program Construction, Dagstuhl-Seminar-Report; 84; 07.03.-11.03.94 (9410)

R. Klette, W.G. Kropatsch, F. Solina (editors):
Theoretical Foundations of Computer Vision, Dagstuhl-Seminar-Report; 85; 14.-18.03.94 (9411)

A. Buchmann, S. Chakravarthy, K. Dittrich (editors):
Active Databases, Dagstuhl-Seminar-Report; 86; 21.03.25.03.94 (9412)

F. Meyer a.d. Heide, H.J. Prömel, E. Upfal (editors):
Expander Graphs, Random Graphs and Their Application in Computer Science, Dagstuhl-Seminar-Report; 87; 11.04.-15.04.94 (9415)

J. van Leeuwen, K. Mehlhorn, T. Reps (editors):
Incremental Computation and Dynamic Algorithms, Dagstuhl-Seminar-Report; 88; 02.05.-06.05.94 (9418)

R. Giegerich, J. Hughes (editors):
Functional Programming in the Real World, Dagstuhl-Seminar-Report; 89; 16.05.-20.05.94 (9420)

H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization , Dagstuhl-Seminar-Report; 90; 23.05.-27.05.94 (9421)

T. Dietterich, W. Maass, H.U. Simon, M. Warmuth (editors):
Theory and Praxis of Machine Learning, Dagstuhl-Seminar-Report; 91; 27.06.-01.07.94 (9426)

J. Encarnação, J. Foley, R.G. Herrtwich (editors):
Fundamentals and Perspectives of Multimedia Systems, Dagstuhl-Seminar-Report; 92; 04.07.-08.07.94 (9427)

W. Hoeppner, H. Horacek, J. Moore (editors):
Prinzipien der Generierung natürlicher Sprache, Dagstuhl-Seminar-Report; 93; 25.07.-29.07.94 (9430)

A. Lesgold, F. Schmalhofer (editors):
Expert- and Tutoring-Systems as Media for Embodying and Sharing Knowledge, Dagstuhl-Seminar-Report; 94; 01.08.-05.08.94 (9431)

H.-D. Ehrich, G. Engels, J. Paredaens, P. Wegner (editors):
Fundamentals of Object-Oriented Languages, Systems, and Methods, Dagstuhl-Seminar-Report; 95; 22.08.-26.08.94 (9434)

K. Birman, F. Cristian, F. Mattern, A. Schiper (editors):
Unifying Theory and Practice in Distributed Systems, Dagstuhl-Seminar-Report; 96; 05.09.-09.09.94 (9436)