OPUS One - OLAT (An Artificial Intelligence - Multi Agent based Adaptive Learning Environment)

Attilio Pedrazzoli

Abstract -- For e-learning environments, "adaptive learning" or "learning path adaptation" is a critical issue to enhance teaching quality of an e-learning session. Adaptive learning provides "Subject Matter" content, adapted to a student's actual knowledge and learning style. Generally, elements of adaptive learning approaches include: monitoring student activity, interpreting the results, understanding students' requirements and preferences, and using the newly gained information to facilitate the learning process (Paramythis and Loidl-Reisinger, 2004).This paper proposes a concept based on OLAT, an Open Source, Java LMS, developed at the University of Zurich, to support "Learning Entities" (LE's), with an adaptive learning environment. The proposed concept is based, among other, on an Artificial Intelligence – Tutoring System, to identify, monitor and adapt the student's learning path, according to the students actual knowledge, learning habits and preferred learning style. The proposed concept is not imposing any didactical boundaries or implied course structures. The concept heavily relies on "real time" adapted profiles, to ensure an open, generic approach for the instructional designer or course author. The proposed concept supports tutored, collaborative sessions and activities, today increasingly used in modern pedagogy. The integrated AI based tutoring facility is intended to support the "human tutor" with valuable "Learning Entity" (LE) performance- and activity data, based on the integrated "Behavior Recorder" (BR), that allows the human tutor to confirm or overrule actions suggested by the AI tutor. Furthermore, the student has the possibility to select the level of AI tutoring interventions or switch to a "subject matter" exercise mode if he feels to do so. The concept presented is target driven, it combines a personalized level of surveillance, learning activity and learning path suggestions to ensure the students learning motivation and success.

Keywords: OPUS One, OLAT Learning Management System, eTutor, Artificial Intelligence based tutoring, Adaptive Learning Environment, Open Source Software,

I. INTRODUCTION

While there has been a sustained research effort in the application of artificial intelligence to education over the past decade with notable success stories, intelligent tutoring, adaptive learning and learning path adaptation has had relatively moderate impact on education and training in the world. The creative vision of intelligent computer tutors has largely arisen among artificial intelligence researchers rather than education specialists. The bottom line is that intelligent tutoring systems should generally be evaluated according to the capability of the concepts in interpreting and responding to student behaviors, needs and educational effectiveness -rather than with respect to artificial intelligence criteria's.

The key of success is the ability to provide a complete tutoring concept represented by a combination of an "automatic tutor" covering the majority of the needed tutoring requests with the possible intervention of a "Human Tutor" able to supervise the activities of the learning entity (LE) with the possibility to take additional personalized actions for the LE. Having a "Human Tutor" involved will ensure the needed "personal" contact for the LE in the elearning environment.

Tracing the student's step-by-step solution enables the tutor to provide personalized instruction in the problem solving context. Prototypically tutors provide immediate feedback on each problem solving action: recognizably correct actions are accepted and unrecognized actions are rejected. The proposed eTutor concept do not try to diagnose student misconceptions and does automatically give problem solving advice. Instead, it allows the student maximum opportunity to reason about the current problem state, monitoring and assisting his/her approach.

Generally, tutors do provide a feedback messages (hints) if the student appears confused about the nature of the current problem definition or a problem solving attempt. The proposed concept recognizes three general levels of advice: a reminder of the current target, a general description of how to achieve the goal, and a description of exactly which problem solving action should be taken. Each of these three levels may be represented by multiple guidance steps.

Manuscript received Monday, June 22, 2009, accepted for the WCECS 2009-ICEIT-09 on July 7,2009. This work was supported by GSi Edu-Research, a nonprofit Organization involved in advanced AI assisted eLearning research projects.

A. Pedrazzoli , PhD, MSc, is CTO and Senior Executive Consultant with GSi Edu-Research, Zurich/Lugano – Switzerland. (email avp@gsiserver.com)

II. OLAT - OPUS ONE CONCEPT AND ARCHITECTURE DESCRIPTION

The OPUS One / OLAT platform consists of the following building blocks [Figure 1.] :

- *OLAT Core Learning Management System with standard key features like:*
  - *Personalized learning environment*
    - Management of OLAT bookmarks
    - Management of user files via HTTP or WebDAV
      - Configuration function for the personal user standard home page
      - Forum and Folder modification via e-mail or RSS feed
      - Full-Text Search

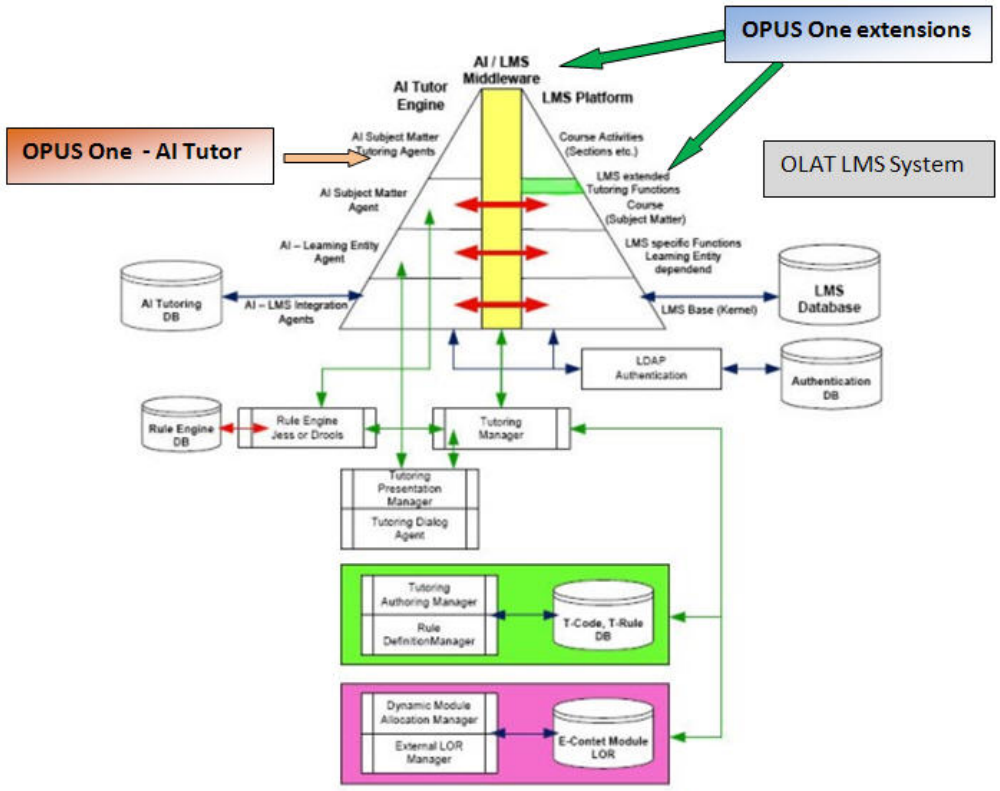

Figure 1: OPUS One Global Architecture

- *Flexible course system*
  - Based on IMS Learning Design concepts
  - Building of any user-defined course structure with the standard course editor
- *Shibboleth integration, global single sign-on facility*
  - Controls the access to courses by AAI attributes (via LDAP, Active Directory etc.)
- *Basic Support for collaborative work*
  - Sharing resources with other authors (learning resource repository)
  - Building project groups outside the course context
  - Using discussion forums, file share space, contact forms etc.
  - Virtual presence and chat functionality (via integrated Jabber IM system)
  - Internal Wikis in courses and groups
  - Built-in calendar
- *Administrative course tools*
  - Course editor: creation of courses using OLAT course elements
  - Group management: management of group / learners relationship
  - Rights management: grant specific user access roles to courses
  - Assessment tool: extensive test and assessment functions
- *Compliance with global e-learning specifications / standards*
  - IMS Question and Test Interoperability (QTI) 1.2.1 runtime engine and editor
  - IMS Content Packaging (CP) 1.1.2 runtime engine
  - SCORM 1.2 runtime engine
- Online help, Context sensitive help, Comprehensive help, manual, tutorials and discussion forum
- *Multilingual*
  - German, English, French, Italian, Spanish and a growing number of
  - other languages (Portuguese, Russian, Czech, Chinese, Farsi, etc.)
  - Online tool for additional translations (UTF-8 support)
- *Java architecture*
  - OLAT-MVC framework
  - Supports any database (MySQL is the reference database)
  - YAML CSS-Layout framework

- *OPUS One OLAT extensions are :*
  - Course module, DB based Logging / Tracking facility interfaced with the AI – "Behavior Recorder" allowing a real time granular, learning progression analysis and immediate LE profile update.
  - eTutoring Portlet / Tutoring Administration / Tutor Assistance facility based on the extended LMS "Role profiles".
  - Improved OLAT course navigation with visual sequence status and course navigation flow control

- Extended "User Role" based "Homepage" personalization showing exclusively user related functions like:
  - "My courses" portlet, "My Groups" portlet, "My Roles" portlet. This feature will only show Student / Author /Teacher owned resources on the personalized Homepage.
  - Personal "Notes Board", including following functions :
    - Course Notes – Multiple personal notes per course identified by Subject, Date/Time, Keywords.
    - Collaborative Notes accessible by same "Group Members" (Project Group or Learning Group) identified by Subject, Date/Time, Keywords.
    - Free Form general personal notes identified by Subject, Date/Time, Keywords.
  - Collaborative writing facility (Personal-, Course-, Group based) as generic function or course module integrated into the course editor.
  - Multimedia aggregator facility as generic function or course module integrated into the course editor.
  - eTutor "tutoring on demand" requests, "Walk trough" mode selection (Exercise Mode), learning path suggestion request functionality driven by the actual user profile
- Video conferencing facility including a "state of the art" Whiteboard, Chat, Desktop sharing, File sharing and Recording functionality (using OpenMeeting as general VC facility, DimDim as course module, integrated into the course editor). The facility allows the creation of public-, group-, temporary Meeting Rooms or Auditoriums for collaborative Videoconferencing or "one on one" sessions. The user access mode is defined in the OPUS One user profile. Foreign Videoconference Members can be invited via email.

- *OPUS One / OLAT bidirectional asynchronous external environment wrapper*
  - The external environment wrapper is a 6 agent AI community designed and implemented to tightly integrate external, reusable learning content into the OPUS One / OLAT LMS. The external wrapper agents are profile driven, able to capture data structures and data to be transferred and integrated into and from the OPUS One / OLAT LMS.
    A practical example is the integration of LAMS eContent sequences as generic OLAT course modules, able to pass data, as an example assessment/test results, done on the external LAMS environment, into the native OLAT LMS "My Achievements" structure.
    This functionality is also used to synchronize the global Student "Grade book" in the ePortfolio facility (Mahara System).

- *OPUS One User Area based on Apache Pluto Portlet Container*
  - The Apache Pluto Portlet Container provides a runtime environment for portlets implemented according to the Portlet API specification. The portlet container is not a stand-alone container like the servlet container; instead it is implemented as a thin layer on top of the servlet container and reuses the functionality provided by the servlet container. Architecturally, it provides an interface between the portal (OPUS One/OLAT) and portlets. The OPUS One User Area is a collection of portlets running in the Pluto Portlet Container. Following portlets are available in the OPUS One User Area :
    - Del.icio.us API portlet facilitate the access to the private del.icio.us user space with a possible feed (RSS) into the LE Mahara ePortfolio.
    - Facebook API portlet (same as above)
    - Wordpress MU / Wordpress Buddy environment portlet, allowing users to access the global multiuser/multiblog tool and publishing environment with the possibility to integrate threads into the private Mahara ePortfolio.
    - More "social activity portlets" are presently under development

    The access to the user area is established from the Student Home page of the LMS.

- *OPUS One 3HTutor (Cognitive Tutor), an AI – Multi Agent System (AI-MAS) community concept developed using the AI Cougaar framework*

III. AI TUTORING ENGINE

The Ai Tutoring Engine is a Cougaar framework based, AI MAS implementation to perform the eTutor functionality to the LMS platform. The implementation can be viewed as a repository of agents organized in a 2 level hierarchy.

- *Level-1 - Activity Management Agents* like
  - Learning entity agents
  - Subject matter agents
  - Presentation agents
  - Prediction agents etc.
- *Level-2 – Tutoring function performing Agents* like
  - LE tutoring request agents
  - Subject matter tutoring request agents
  - Hint agents
  - Rule access agents etc.

IV. AI Tutoring Agent definition and concept

Agent definition : An agent is a software entity which autonomously communicates with other software Agents to achieve domain-specific or function-specific actions. Multiple agents often collaborate as peers in a Peer-to-Peer (P2P) distributed network. Agents developed for the tutoring system act according to the associated agent profile. The associated profile will determine the type and activity of the agent in question. Level-1 agents are defined as primary- or management agents , they are able to duplicate (scale) themselves and collaborate with each other. Level-1 agents are "Main Function" oriented agents performing functions like *Learning Entity supervision* or *Subject Matter related functions*. Level-1 agents have a Supervisor – Role, they activate and supervise Level-2 agents to perform requested or dedicated, tutoring specific tasks.

A. AI Tutor Agent profiling

AI Tutoring Agents are profile driven. Agent profiles are loaded when the corresponding Agent gets activated. Agent profiles are defining the behavior of the Agent in question. Generally, Agents have a generic profile, if an Agent has to perform specific actions like for example supervising a learning entity (LE), the profile is the guideline for specific, foreseen activities. In the present version, dedicated profiles exist for LE related – and Subject Matter related Agents.

B. Profile definition

Agent profiles are parameter structures intended to characterize the Agent function. The profile contains all necessary information to allow the scheduled agent to perform his foreseen activity. Idle agents are per concept definition "generic", they become dedicated with the assignment of a profile.

Learning Entity Profile: Global parameters

- AI surveillance level
  This parameter shows the surveillance intensity – Value 0 to 5 initial value 1
  0 = Silent Mode
  1 = AI Tracking and Logging ON
  2= + "Subject Matter" Hint facility enabled
  3= + Tutoring Engine enabled, tutoring on request ON – Silent Mode
  4= + Learning path suggestion ON – Advise Mode based on tracking history
  5= + Tight supervision, maximum surveillance
- Suspend parameter (0 or 1) if 1 LE is suspended to proceed until Human Tutor interaction

Learning Entity parameters

- Profile ID - User ID from the OLAT LMS
- Tutor ID – Primary Human Tutor ID (multiple entry possible)
- Personal involvement level (Activity measurement – set by the AI Engine)
- Personal difficulty Level (from AI monitoring)
- Personal tendency (+ / - from AI monitoring)
- Personal "on track" level (Value set by the AI Tutoring Engine)
- Personal cumulative grade (Value added by the LMS or Human Tutor)

Per Course parameter block (dynamically assigned when entering a course, based on previous course activities, in most cases supplied by the Behavior Recorder facility)

- Course ID (from LMS)
- Previous Section ID (ID from dynamic learning path agent)
- Next possible Section ID's Proposed learning path suggestion if enabled from dynamic learning path agent based on rules.
- Course surveillance level
- Course involvement level (Activity measurement – set by the AI Engine)
- Course difficulty Level (set by the Behavior Recorder)
- Course tendency (+ / - set by the Behavior Recorder)
- Learning path indicator (Learning path indicator rules)
- Tutor request level (AI monitoring)
- Real time sequence control (Yes / No enforcing a learning path)
- Minimum achievement level requested (Value)
- Knowledge base ID (Knowledgebase profile ID)
- Rule base ID (used for Rule Engine access)

Knowledge base profile

- Knowledgebase ID (Pointer for other profiles)
- Actual Section ID (from dynamic learning path agent)
- Section Rule sequence ID (used for the Rule Engine)
- Learning path indicator (Sequence value)

V. OPUS One as an Adaptive Learning Environment

The term "adaptive" is one of the "trends" in the eLearning industry today. It's being associated with a range of system characteristics and capabilities of Learning. Therefore, it is necessary to qualify the qualities one attributes to a system when using the term. A learning environment is considered adaptive if it is capable of: monitoring the activities of its users; interpreting these on the basis of domain-specific models; understand user requirements and preferences out of the interpreted activities, appropriately representing these in associated models; and, finally, acting upon the available knowledge on its users and the subject matter at hand, to dynamically facilitate the learning process.

A. Categories of adaptation in learning environments

Adaptive Interaction, refers to adaptations that take place at the User interface level and are intended to facilitate or support the user's interaction with the learning platform, without, however, modifying in any way the learning "content" itself. OPUS One/OLAT allows to a certain extend to personalize the User Home page in enabling or disabling functions and setup the graphical appearance according to user needs.

Adaptive Course Delivery, represents the most common used anthology of adaptation techniques applied in learning environments today. In particular, the idiom is used in reference to adaptations intended to alter a course (or, series of course sequences) to a specific learning entity. A major factor behind the implementation of adaptive techniques include the compensation for the lack of a human tutor (who is capable of judging the learner capacity, goals, etc., and advising him on a personalized base), improving subjective evaluation of achievements by the learner. The most typical examples of adaptations in this category are: dynamic course (re-)structuring; adaptive navigation support; and, adaptive selection of alternative (sequences of) course material (Brusilovsky, 2001,2003). OPUS One eTutor facility and the integrated "Behaviour Recorder" (BR) are supervising the learner in "real time" according to the learners profile. If the learner encounters difficulties in solving course sequences (detected by the BR) or if the learner request tutoring, a "sequence (context specific)" of rule driven advise mechanism will be initiated and additional, tailored "problem solving" content will be proposed. This is considered a individual "exception" event and will be logged accordingly.

External Content Discovery, refers to the discovery and storage of learning material / "content" from external sources, environments like other LMS's or specialized repositories. The adaptive component of this process lies with the integration and publication of this additional material among the course community. OPUS One / OLAT has the capability to detect, using the eTutor surveillance facility, the external search activity of the learning community, advising the learners to publish or share his additional material among his learning community and integrate the additional material in the course repository.

Adaptive Collaboration, refers to the involvement between multiple learners or group activities (and, therefore, social interaction), proposing collaboration towards common objectives. This is an important dimension to be considered as pedagogy is moving more and more towards collaborative approaches, which is what modern learning theory increasingly emphasizes, the importance of collaboration, cooperative learning, communities of learners and social negotiation (Wiley, 2003). Adaptive techniques can be used in this direction to facilitate the communication / collaboration process, ensure a good equivalent between learning communities. OPUS One / OLAT supports a variety of "tutored collaborative activities" integrated as components into a course or as standalone functionality. Functions like Wiki, global or dedicated Forums, collaborative writing or collaborative assessment functions are just a few examples. In an exception case, the eTutor facility is able to propose additional collaborative actions to learners if the exception rules foresee such an action.

B. Models used in an adaptive learning environment

The categories of adaptation in learning environments are based on well-established models, procedures and processes. OPUS One supports the described models with dedicated, AI based agent communities using fine grained sub models to the main domain model category.

***The domain (subject matter) model*:** The task of the domain model is focused on adaptive course or content delivery. The domain- model is usually a representation of the course being presented. In OPUS One, every "subject matter" is controlled by a subject matter profile, and serviced by a subject matter agent. Every domain- or subject matter model has access to a knowledge base profile, where adaptive-/ personalization rules and procedures are defined. Every adaptive action is the result of an action request generated by a "knowledge exception" usually originated by the "Behaviour Recorder" or by a learner initiated tutoring request. The possible adaptations to the subject matter can be summarized as follows : Adapt the learning path with additional, problem- or context oriented content to overcome the student's difficulties, invoke repetition of sections in question with an increased tutoring and surveillance level. Every adaptation action is tracked and logged by the Behaviour Recorder, the student profile is updated in real time accordingly.

***The" learning entity" (Student) model*:** In OPUS One the term learning entity or Student model is used to reference the characteristics of the learning entity defined in the learning entity profile. The specific approach to modelling and adaptation is accomplished by combining decision parameters from the learning entity profile and the associated subject matter profile.

***Group entity model*:** Similarly to the learning entity model, group models seek to capture the characteristics of groups of users / learners. In OPUS One, Group- and Student models are considered "Learning Entities". The main differentiating factors between the two are: (a) The approach in tutoring collaborative activities (b) group models are based on the identification of groups of learners that share common characteristics, behaviour, etc. OPUS One handles Group tutoring and resulting adaptation actions according to a combined Group- and associated member profile decision parameters, and the specific Group subject mater action rules.

***The adaptation model*:** This model incorporates the adaptive theory of OPUS One. This theory is based on context sensitive , subject matter knowledge base entries with associated exception rules. The associated originating exception source (learning entity or group entity) based on different profiles and subject matter parameters, will propose an adaptive suggestion. Specifically, the (possibly implicit) adaptation model defines *what* can be adapted, as well as *when* and *how* it is to be adapted (adaptation profile and adaptation knowledge base rules)

VI. OPUS One user selectable adaptive functions

eTutor functions can be selected from the Header Function selection. Selecting this function will activate the eTutor administration agent, profile and role of the requester will be identified and a personalized eTutor function selection windows will be made available. All functions presented are under control of the Artificial Intelligence implementation, a separate environment from the LMS.

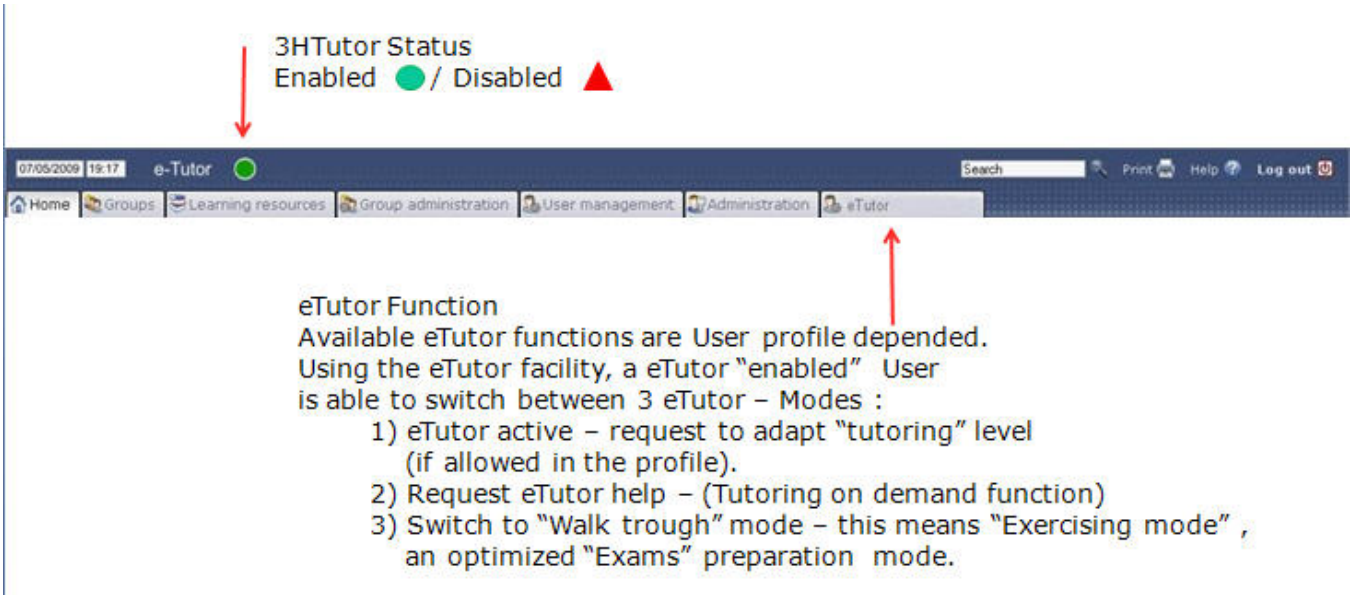

Figure 2 – Tutor Access / Functionality

Examples of such functions are: - LMS learning mode change (tutored, walk trough, silent), - On demand eTutor or "human Tutor" request, - eTutor Administration functions, - Author related functions like "subject matter" related function (profiles, rules, exceptions, predictions), knowledge base extension, etc. [Figure 2.]

VII. Work in Progress

A major area of development is "Authoring". Today every profile is created / edited with dedicated utilities. Same counts for Rules associated to the subject matter knowledge base, exception rules, adaptation rules etc. Furthermore Java or Flash based dedicated tutoring code is created using CTAT and concatenated with the corresponding agent community. All this manually performed activities will be integrated in the OPUS One Authoring tool, presently under development. Another area of development is the area of reusable content modules, dynamically assembled as result of a collaborative action. This implies the integration of a Learning Object Repository (LOR) into OPUS One.

VIII. Conclusion

The OPUS One extensions of the OLAT LMS platform supports the concept for an "Adaptive Learning Environment" using an AI based eTutoring concept, able to support the learner fulfilling his educational goals considering his learning style and actual knowledge level. The concept is able to support "human tutors" with accurate "learner centric" data to better qualify, judge and support the learner. The eTutor reliefs the "human tutor" from time consuming, low level tutoring interventions, supporting the learner directly with a variety of support tools. The "human tutor" can always overrule, add or modify eTutor proposed activities. Monitoring, support and tutoring capability of extensive collaborative functions (internal and external) allowing a more fine grained adaptation / personalization process. Using the learner adaptation monitoring data we are able to confirm the feasibility of the personalized actions for the learner. This issue has to be considered important, since we must have the ability to verify/ modify / add subject matter / knowledge base rules for the adaptation / personalization process to ensure maximum learning success. The OPUS One extensions will be released in the 4th Quarter 2009 - as "Open Source" modules.

Acknowledgements

We (the OPUS One Concept /Architecture Team) would like to express our profound gratitude to the MELS Team of the University of Zurich, for the excellent ongoing support, encouragement, and useful suggestions throughout this research project.

References

[1] Aleven, V.A.W.M.M., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor.

[2] Koedinger, K. R., Aleven, V., Heffernan. T., McLaren, B. & Hockenberry, M. (2004). Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration.

[3] Koedinger, K. R., Aleven, V. A. W. M. M., & Heffernan, N. T. (2003). Toward a Rapid Development Environment for Cognitive Tutors. In U. Hoppe, F. Verdejo, & J. Kay (Eds.), Proceedings of the 11th International Conference on Artificial Intelligence in Education, AI-ED 2003 (pp. 455-457). Amsterdam: IOS Press.

[4] Nhouyvanisvong, A. & Koedinger, K. R. (1998). Goal specificity and learning: Reinterpretation of the data and cognitive theory. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, (pp. 764-769). Hillsdale,NJ: Erlbaum.

[5] Eberts, R. E. (1997). Computer-based instruction. In Helander, M. G.

[6] Landauer, T.K., & Prabhu, P. V. (Ed.s) *Handbook of Human-Computer Interaction*, Amsterdam, The Netherlands: Elsevier Science B. V.
[7] Anderson, J. R. (1993). *Rules of the Mind.* Mahwah, NJ: Lawrence Erlbaum.
[8] Corbett, A.T. & Anderson, J.R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction.*
[9] B. Cheung, K.K. Loo- Fuzzy Logic And Data Mining for e-Learning Personalization. The Third IASTED International Conference on Artificial Intelligence and Applications (AIA 2003), Spain, September, 2003
[10] Zhang, J., Cheung, B. & Hui, L. (2001). An Intelligent Tutoring System: Smart Tutor. In C. Montgomerie & J. Viteli (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2001 (pp. 2130-2131). Chesapeake, VA: AACE
[11] Cheung B. Kwok L.K., "Teaching and Learning through Space Online Universal Learning (SOUL) Platform" in the E-Education Era. The 20th IASTED International Multi-Conference on Applied Informatics (AI 2002), Innsbruck, Austria, February 18-21, 2002
[12] OLAT - LMS Homepage : http://www.olat.org
[13] Cougaar AI Framework : http://cougaar.org
[14] CTAT Homepage : http://ctat.pact.cs.cmu.edu
[15] Mahara ePortfolio : http://mahara.org