A Framework for Agent-Supported E-learning

Steffen Mencke, Reiner R. Dumke

Department of Computer Science, Otto-von-Guericke University of Magdeburg

Key words: *E-learning, Agent technology*

Abstract:

This paper presents a framework to describe the crossover domain of e-learning and agent technology. Furthermore it is used to classify existing work and possible starting points for the future development of agent techniques and technologies order to enhance the performance and the effectiveness of several aspects of e-learning systems. Agents are not a new concept but their use in the field of e-learning constitutes a basis for consequential advances.

1 Introduction

E-learning is nowadays one of the most interesting of the "e-"domains available through the Internet [1]. In general it refers to a wide range of applications and processes designed to deliver instruction through computational means [2]. Its well accepted importance in modern education is beyond all question, but still can be increased by e.g. intelligently improving several aspects of corresponding systems.

Agent technology is such a modern approach. The agent idea goes back to works of Carl Hewitt in the field of artificial intelligence in 1977. He described an object "actor" being interactive, independent and executable. Furthermore it was intended to have an internal state and being able to communicate with other objects [3].

The crossover domain of agent-supported e-learning is focused on fundamental research about the applicability of agent techniques and technologies in order to enhance the performance and the effectiveness of several aspects of e-learning systems. Agents are not a new concept but their combination with AI in the field of e-learning constitutes a basis for consequential advances. In [4] Wilson et al. exemplary listed the following:

- Pedagogic diversity: By agent technology a diverse set of learning models can be parallel implemented, because it becomes feasible to configure low-level elements of the architecture. Thereby distinct pedagogical and business models can be realized.
- Pedagogy-driven implementations: Modular processes can be offered as services by agents to drive system implementation by pedagogical imperatives rather than by the construction technology itself.
- Support of collaboration between organisations: Agents can decompose complex tasks in order to provide basic services in a kind of construction kit. Thereby for example needed applications can be easier defined and shared to provide functionality that is common to all institutions and to share information between them.
- Better return on technology investment: ROI is increased because applications or better functionality can be acquired as needed and integrated in the existing framework. That reduces purchasing and implementation costs, particularly in terms of staff development and training.

- Faster technology development: Functionality is separated from the interface and is replaceable more easily. By this a modular and flexible technology base is provided. Individual components can be implemented, added and replaced more easily

After some fundamental introductory notes about e-learning and agent technology in chapter 2 their crossover domain is described by the presentation of a novel framework in chapter 3. Recent advances in agent-supported e-learning are classified and exemplary outlined in chapter 4. In section 5 we finish with a conclusion and some remarks about future work.

2 Fundamentals

2.1 E-learning

E-learning is an already established concept. It is not intended to exclude existing methods and technologies of classroom education. An appropriate use might complement them [1]. First roots can be traced back to the 60ties with the PLATO and TICCIT experiences in the USA [5]. The first knowledge based tutoring application appeared in domain of artificial intelligence in early 1970s. The first applications were simple automated instruction tools. Next fundamental steps were taken in the early 1990s. Authoring systems for intelligent tutoring systems were designed and developed. Furthermore generic approaches were implemented, e.g. with the usage of task and domain ontologies [6].
A general view on E-learning systems, involved roles and components is visualized in figure 1. In a typical learning environment learners, authors, trainers and administrators are the main groups ([7], [8]). Sometimes these roles overlap, so trainers and authors can be the same person, especially for small E-learning systems. The content to be presented is created by the authors using authoring systems, stored in the learning management system (LMS) and thereby made available for the learner via a run-time system. Especially for larger applications additional roles can be identified e.g.: content expert, instructional designer, programmer, graphic artist and project manager [8].

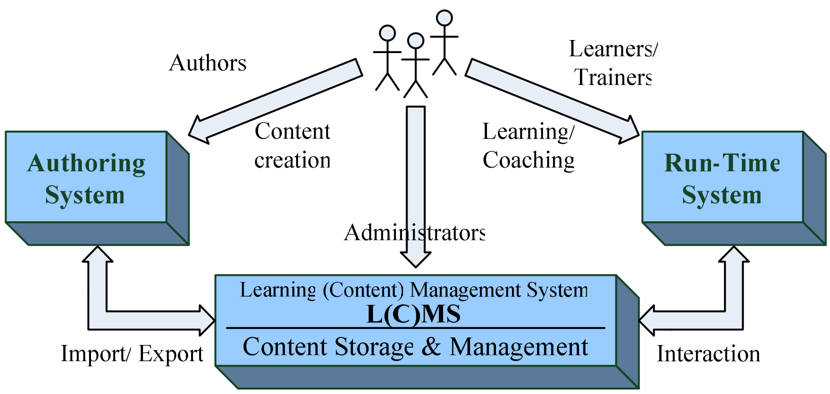

Figure 1. General E-learning system and involved roles (cp. [9])

2.2 Agent technology

There exists no single definition for agents, but a lot of discussion (e.g. [10], [11], [12]). Almost every author seems to propose own needs and ideas what leads to a variety of definitions depending on the targeted problem area. The expressed spectrum determines reasonable application areas as for example user interfaces, telecommunications, network management, electronic commerce and information gathering [13]. Russel and Norvig described this multiplicity aspect in this way [14]: “The notion of an agent is meant to be a tool …, not an absolute characterization that divides the world into agents and non-agents.”
The Foundation for Intelligent Physical Agents (FIPA) defines agents as computational processes that implement the autonomous, communicating functionality of an application [15]. Mostly corresponding systems, architectures and points of view are based on using attributes as defining entities. For example Wooldridge and Jennings define agents as software-based computer systems with certain properties like autonomy, social ability, reactivity and pro-activeness [16].

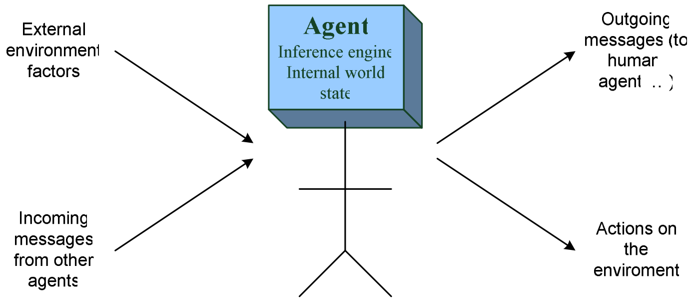

Figure 2. Agents and their interaction with the environment [cp. 17}]

3 A framework for agent-supported e-learning

Literature provides several approaches for the application of agent technology for the domain of e-learning. A “pedagogically neutral, content neutral, culturally neutral, platform neutral” [18] framework for the integration of possible architectural components is described below. It is intended to be used as an abstract representation of the functionality of certain e-learning artefacts that is provided or supported by a set of agents.
Some of the main proposed key features are e.g.:

- Adaptable architectural components with extensive (additional) agent support.
- Identification of approaches for agent-based support for e-learning systems.
- Separation and provision of basic and specialised services for reuse and optimised system development. Implementation aspects of basic aspects are hided from the user.
- Improved focussing on key elements as e.g. pedagogical issues becomes possible.
- Exchange of application functionality between organisations and interoperability are eased.
- Extensive evaluation capabilities of users and system artefacts.

The developed framework is based on the abstract framework [19] of the IMS Global Learning Consortium, Inc. and the SUN Microsystems E-learning Framework [20]. It is further refined by several aspects of related architectures and models as for example the Open Knowledge Initiative [21], the ADL Sharable Content Object Reference Model (SCORM)

[22], the IEEE Learning Technology Systems Architecture (LTSA) [18] and the Learning Technology System Architecture of the Carnegie Mellon University [19]. Special requirements and advantages evolve from the intended application and integration of agent-based technology. Thereby it is especially focused on adaptation, autonomy, support and flexibility.

The novel framework, visualised in figure 3, takes into account the diversity of users involved in learning processes in contrast to the functional models of the abstract IMS framework [19]. Next to the main groups of learners, authors, trainers and administrators, support for content experts, instructional designers, graphic artists and project managers is needed [8]. Their requirements for an e-learning system are grouped and depicted by several functional environments. Thereby the Presentation Environment (PE) is the basic platform for the integration and display of the other environments. It is a basic element connected to all other environments, like the Administration (AE) and Interaction Environments (IE), too. Appropriate and specialised access to functionalities for the learner is provided by the Delivery (DE) and Working Environments (WE). Authors, trainers, content experts, instructional designers and graphic artists benefit from support of the Learning Unit Environment (LUE) and the Content Environment (CE).

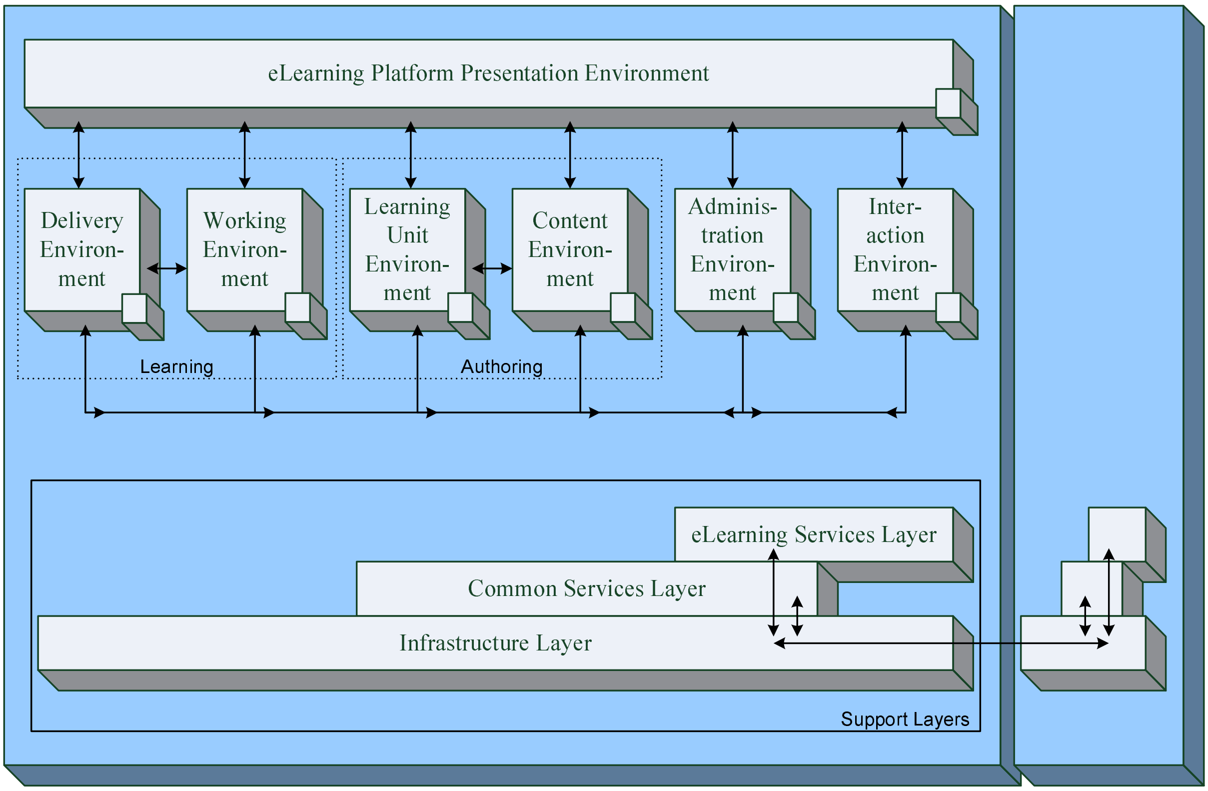

Figure 3. A framework for agent-supported e-learning

To guarantee flexibility, extension and interoperability the whole framework is based on three support layers. They are differently specialised and are providing infrastructural support, common services and e-learning services. We hereby define a service as a functionality providing entity, which can be potentially used in different environments. Meanwhile the environments are further hierarchically refined as described in the following subsections, fundamental needed and desirable services are horizontally integrated as provided by the support layers. The specific services can be ordered and used on demand. They also provide the basis for the connection and data exchange between certain implementations of the proposed framework. This abstraction of common facilities from the classic “LMS only” model was already proposed e.g. [19] and [20].

3.1 E-learning Platform Presentation Environment

The E-learning Platform Presentation Environment (PE) is the core of the graphical user interface (GUI) of every e-learning system implemented following this framework. It provides personalised access for the different possible user groups. Exemplary use cases are visualised in figure 4.
It mainly provides access to the learning, authoring and administration environments (as described in [19]), as well as to the interaction environment.
An important aspect of GUIs for e-learning is the adaptability; the personalisation of certain aspects based on collected information or assumptions about the user. That refers to all related environments and may result in adaptive navigation support, adaptive presentation and adaptive content [23]. Adaptive navigation support is related to the guidance of users and can be established by global and local support mechanisms, by local orientation, global support for orientation and by the management of individual views. Adaptive presentation can be achieved by the sorting of resource fragments, the adaptive content presentation due to different media formats and the adaptive provision of content because of differing quality, transmission contexts and different languages. Classic approaches like changes in font size, font type and font colour can be used for adaptive presentation, too. Methods for adaptive content are e.g. basic, additional and comparing explanations, explanation variants and the sorting of information fragments [23].
Context adaptability is supported by the advantage to integrate different implementations of the proposed environments, extended with capabilities to receive and process context-sensitive information. By this mobile, ubiquitous learning becomes possible.

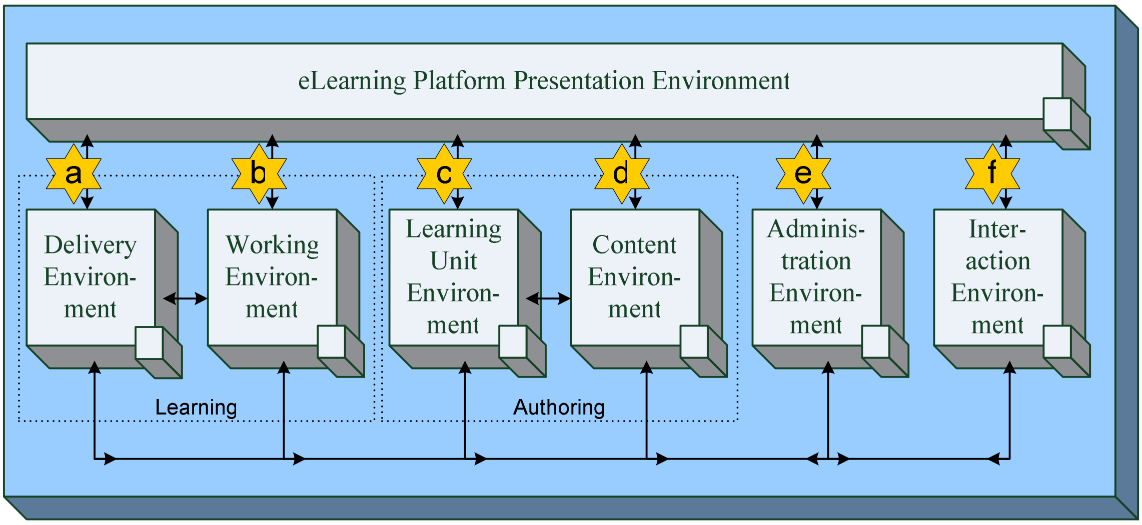

Figure 4. E-learning Platform Presentation Environment

Use case a: Request and presentation of the next part of a course
Use case b: Request and presentation of personal annotations to a certain topic
Use case c: Creation and management of courses or certain course substructures
Use case d: Creation and management of learning objects (LO)
Use case e: Update of entries in a user model
Use case f: Interaction with other learners, tutors or experts

The different environments themselves may interact with each other. A first primary relation exists between the two learning environments. The DE and WE are closely connected, because of the high possibility of exchanging data. Functionalities provided by the WE, like media processing, can be requirements of certain tasks of the actual course presented in the

DE. Similar connections are needed for the LUE and CE. The learning objects are integral part of the courses that are authored within the LUE.
Nevertheless the AE and IE will exchange data with all other environments, because each one needs to be administered and collaboration between different users is always possible, too.

3.2 Learning Environments

The Delivery and Working Environments are grouping the functionalities of learning systems to enable the learning itself. Therefore they mainly fulfil requirements demanded by learners.
The DE presents the course, its structure, course metadata, enables course catalogue browsing, realises the registration and is responsible for other all functionalities that are directly connected with the presentation of and working with learning content during the learning process.
The WE is grouping functionalities for the support of the learning process. That refers to e.g. to classic requirements known from classroom learning. Components for web search as well as for the access to certain repositories are needed to get additional information about the topic of the course. It is important for the personal learning progress to be able to make private annotations to the course content and to manage own additional information, e.g. as a list of links or in a private file system. A scheduler for collaborative work and time management and the access to office tools are needed under certain circumstances. Figure 5 visualises these chosen aspects for parts of the learning environments.

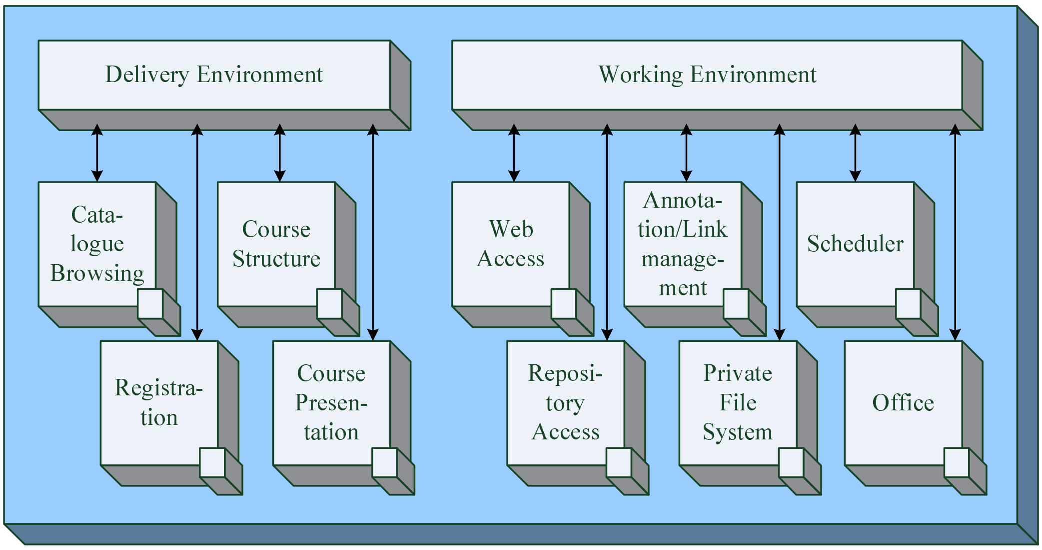

Figure 5. The Learning Environments

The learning environments need connections to the Administration and Interaction Environments and to the support layers. Administration for example is needed for the management of individual preferences; meanwhile interaction is fundamental for collaborative learning tasks. As for the other environments the support layers are providing access to basic information, repositories and functionalities that are needed for the functionality of the actual environment itself.

3.3 Authoring Environments

The Learning Unit Environment and the Content Environment are focused on functionalities to support the authoring process of educational content (e.g. basic content, learning objects, assessments/tests, courses). The process' nature is iterative: the planning, design and production cycle is followed by a new iteration after an evaluation for continues improvement (cp. figure 6) [8].

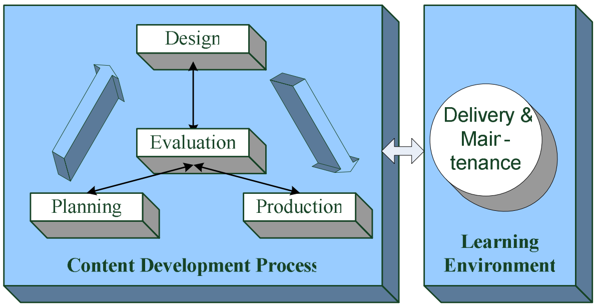

Figure 6. Content Development [8]

The CE provides functionalities for the planning, design, creation, assembly and management of basic content fragments. Thereby different media types need to be taken into account.
The LUE is focused on the processing of more complex content. Therefore we define a learning unit as a piece of information that is more complex than the content fragments and whose usage is targeted to education. Entire courses and course substructures like assessments or tests are learning units.
The development and authoring of strategies for course assembly is a new key element of the proposed framework. Those, e.g. didactical, strategies are needed for the high quality of assembled learning resources, because they provide expert knowledge und user guidance for this complex task. Figure 7 is presenting chosen aspects of the CE und LUE.

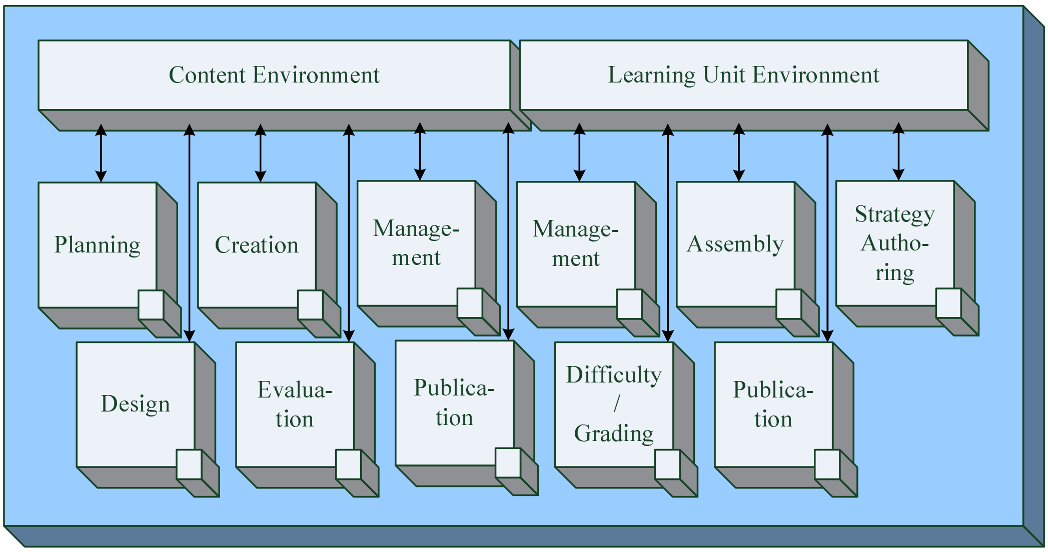

Figure 7. The Authoring Environments

Like the learning environments, the described authoring environments need connections to the Administration and Interaction Environments and to the support layers by the same token.

3.4 Interaction Environment

Following Brown and Druguid in [24] learning is "a remarkably social process. Social groups provide the resources for their members to learn." There are several social reasons for interactivity. It decreases isolation of the participants and increases the flexibility to adapt new conditions. Furthermore it involves more human senses into learning and increases the variety of learning experiences (multi-cultural environments, communication capabilities,

etc.). Furthermore interactivity builds a sense of group identity and community. Nonetheless interaction sometimes is a fundamental requirement for certain courses [25]. Figure 8 is dedicated to chosen fragments of the Interaction Environment.
The proposed framework integrates multiple communication channels as technical support for human-to-human respectively human-to-computer interaction and is extended by additional support tools. An avatar is used as a human representative for e.g. personalisation, identification, anonymisation and as backup in case of absence. Another component is the grouping tool, which is intended to form groups of learners for certain collaborative learning tasks based on user model information and appropriate psychological theories.
Interaction approaches can be distinguished in synchronous and asynchronous. Synchronous tools can provide text-, audio- or video-based chat, application/screen sharing, synchronous Web browsing, shared whiteboards, etc. Asynchronous tools can span e.g. email, wikis, forums, mailing lists or audio/video replay [26].
The IE technically needs close connections to all other environments, because collaborative learning and working may occur in every proposed environment.

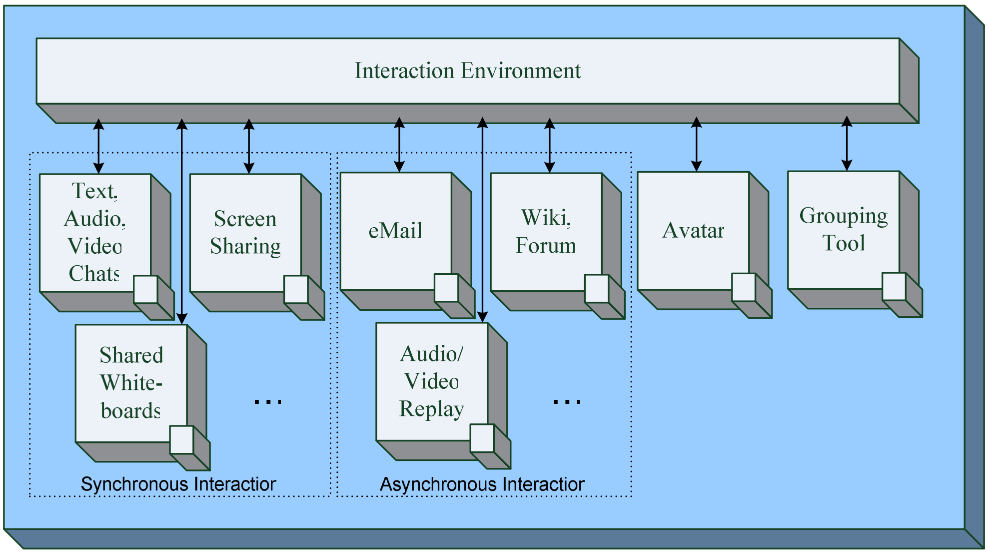

Figure 8. Interaction Environment

3.5 Administration Environment

The administration environment provides access for the management of all environments, system components and support layers. The possibilities are ranging from simple observation to the integration of new components or the update of existing ones. The access to components and the provided functionalities is limited by the access restriction of a particular user.
The most extensive access is possible for the administrators. All other user groups have access to their specific objects and to the adjustment capabilities of the environments where they have access to.
A very important example of needed accessibility is the manageability of the user model for the depicted learner. If it is available and manageable for individuals it gives learner control and responsibility [27]. Thereby it supports meta-learning activities like the monitoring of learning, the setting of personal learning goals; it is the basis for planning goals and supports the reflection about and the tracing of the learning progress by the comparison of set goals.
As presented in figure 9, the AE needs connections to all other environments.

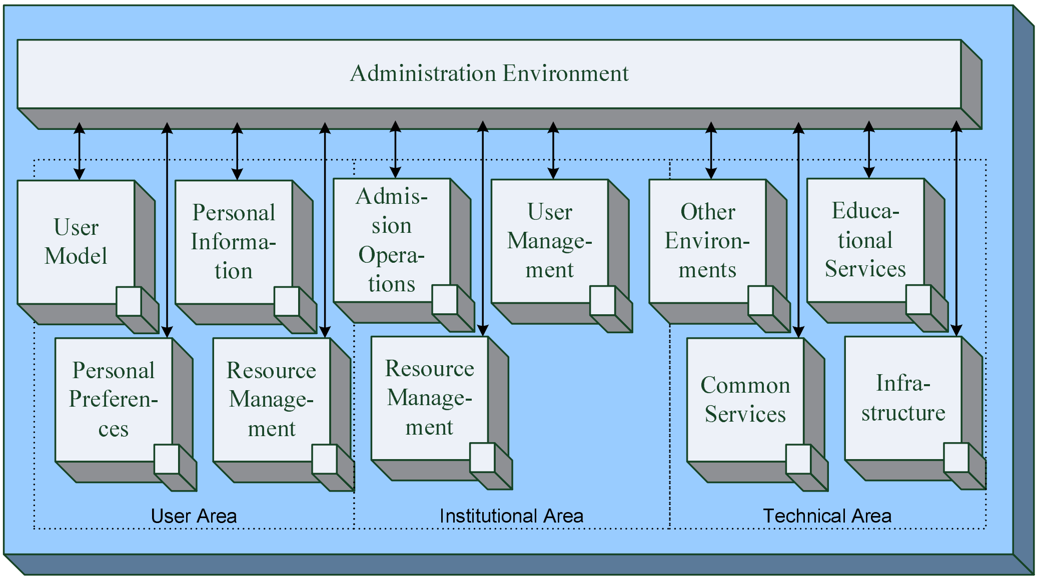

Figure 9. Administration Environment

Regarding functionalities we grouped in the user, institutional and technical area. Within the user area all aspects are pooled that are related to specific user tasks. Thereby not only learners, but all possible users have access to administration functionalities that are targeted to them, their tasks or resources. Institutional management facilities provide access to services, functionalities and resources that are related to the management of meta-activities within the specific institution as e.g. user management, course management, class management, study specification management and certification management. Management capabilities for the classic administrator role are pooled within the technical area.

3.6 Infrastructure and Common Services Layers

The infrastructure and common support layers provide basic functionalities for the e-learning services layer and the parts of the environments. This separation idea was adopted from [21] and [19] and is based on the same motivation. The intended goals are twofold.

- Thereby more complex functionalities of the upper framework elements do not need to re-implement already existing ones; redundancy is avoided.
- By the separation an easier intra-institution work sharing is possible, due to increased portability of the system.

This presented framework differs in the assignment of specific functionalities to certain support layers and environments, as descried below. The infrastructure layer is responsible for basic networking and data transport, selected services are e.g.:

- Exchange of data structures in terms of physical communications, messaging and transaction needs [19]
- Support of complex multi-zone agent communication [28][29]
- Provision of the needed agent platform [30] [15]
    - Agent management
    - Message transport service
    - Agent directory
    - Services directory
    - Agent communication language (ACL)

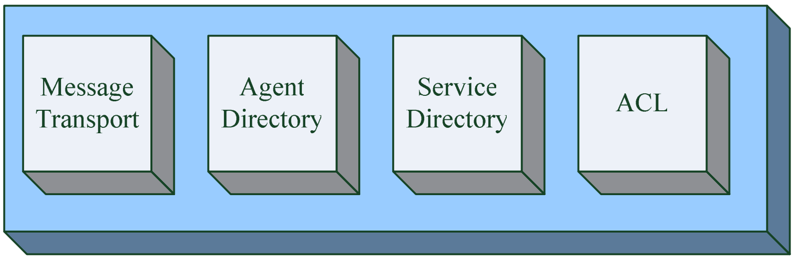

Figure 10. FIPA abstract architecture [30]

The common service layer provides generic functionalities for the upper layer and the framework environments like (adopted and extended from [21]):

- Authentication and authorisation [21]
- Rights management, validation
- Service discovery, database control e.g. for [20] [21]:
    - Learning content
    - Learning meta data
    - Learning assessment
    - Learning administration
    - User repository
- Filing [21]
- Automated resource update [27]
- Logging of technical system aspects
- Virtual centralisation of remote resources [27]

Additional possible services are summarized in a brief overview in [31] as part of the ELF Initiative that is targeted towards a service-oriented approach for e-learning.

3.7 E-learning Services Layer

This layer provides specialised e-learning functionalities. Therefore they can be based on services of lower support layers to provide them to the upper environments. Thereby the provided services reveal fundamental educational and/or crossover nature for the certain environments.
As the most specialised support layer this collection of e-learning specific services represent a second dimension of the proposed framework. The more vertically specialised functionalities of the environments are based on and are supported by multiple adopted implementations of the proposed services. In figure 10 the hierarchy of environmental components is depicted in the upper blue boxes, meanwhile the dots within the net below visualise potential cooperation with the educational services.
To profit from the agent-supported realisation of this framework we propose the implementation and offer of certain e-learning-specific functionalities of the presentation environments as educational services. That e.g. relates to:

(1) Content assembly and sequencing service [20][22]
(2) Content adaptation service
(3) Scheduling service [21] [32]
(4) Learning planner [19]
(5) Annotation/link management service
(6) Cataloguing service [19] [18]

(7) Grouping tool
(8) Interface to external office tools
(9) Brokering service for educational material [28]

More fundamental services are for example:

(10) Evaluation (of e.g. learning progress, learning results, content usage, course usage, user preferences, strategy usage, …) [18]
   - Collecting evaluation data: logging of education-related events, like learner profiling
   - Storing evaluation data
   - Processing evaluation data
   - Evaluation provision
(11) Educational resource management (e.g.: content, learning unit, strategies) [27]
(12) Registration for new courses [33]
(13) Knowledge management
(14) Report management
(15) Dictionary [21]
(16) Mobile learning management [34]
(17) User model service (management, update, …) [27]

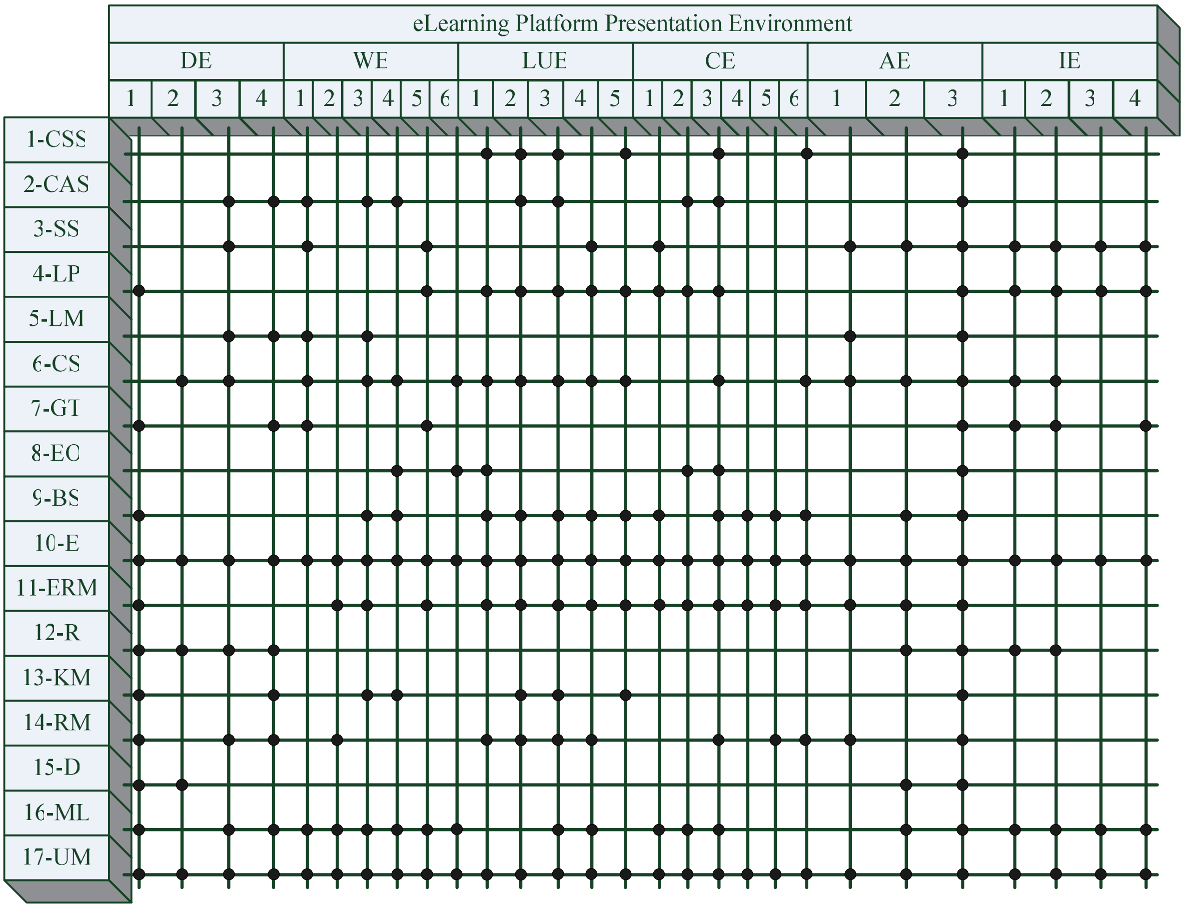

Figure 11. Two-dimensionality of environmental functionalities (h) and services (v)

3.8 Agent-based representation of framework artefacts

Within the last sections a framework for agent-supported e-learning consisting of several artefacts was developed. In this section an agent-based representation of these specific artefacts will be introduced. Therefore we abstract from standardised representations like [30] and focus with the definition on the presented framework.
At first it is important to state, that a certain framework artefact can, but not necessarily must be implemented using agent technology. Either it is an agent or a component following another technological paradigm as e.g. service-orientation or object-orientation.
We define an aL-agent for the proposed framework as a 6-tuple $A = (I, E, P, D, C, T)$ with

- I as a finite set of identification information of this agent: agent identifier, system identifier (ID of the actual implementation of the framework), description …
- E as a finite set of preconditions that describe e.g. other framework artefacts needed for the creation of the actual agent: upper framework container, parallel artefacts, services …
- P as a finite set of parameters for the parameterised creation of the agent: place of creation, rights, timer, functionality-specific initialisation parameters …
- D as a finite set of communication descriptors for the description of the agent's communication interface. Each communication descriptor itself is a triple $D_i = (\Pi, \mathrm{M}, c(\mathrm{O}))$ with Π as the used communication protocol, M as the corresponding communication method of the agent and $c(\mathrm{O})$ as the concept of the ontology O that describes the communication's semantic (extended approach from [35]) as well as input and output parameters.
- C as a finite set of capabilities of the agent, where C_i is a tuple $C_i = (F_i, N)$ with F as a functionality provided the agent and N as a set of needed functionalities N_{ij}, defined as the tuple $N_{ij} = (F_k, G)$ with F_k as the needed functionality and G as the functionality providing framework component.
- T as a finite set of termination conditions: logout of user, time, successful task processing, successful task forwarding …

As an example we present the agent-based representation of three framework artefacts, namely a chat agent, a container agent and a more complex user model agent.

Chat agent: This agent is a recommended part of the collaboration environment providing synchronous communication functionality.
- Information: chat1, system1, "providing text-based synchronous communication functionality"
- Existence conditions: container artefact in collaboration environment, access to user model, access to message exchange server
- Parameters: login information for actual user, actual layout information …
- Chosen communication descriptors:
  - FIPA-ACL, getActualUser(), user(metaInformation)
  - FIPA-ACL, getMessageServerID(); messageServer(metaInformation)
  - KQML, queryHistory(user x), history(personalInformation)
  - …
- Chosen capabilities of actual agent:
  - chatFunctionality, ((sendMessage, chatAgent), (receiveMessage, chatAgent), (sendEmoticon, chatAgent), (receiveEmoticon, chatAgent), …)

  - fileExchange, ((fileStorage, PersonalResourceRepository), (fileLookup, PersonalResourceRepository), …)
  - …
- Termination conditions: user logout, user initiated termination

Container agent: A container agent is a structuring graphical element within the graphical user interface (GUI) of the e-learning application. That can be a window for example.

- Information: container1, system1, "providing space for sub-GUIs of specialised agents"
- Existence conditions: container artefact in the certain environment
- Parameters: actual layout information …
- Chosen communication descriptors:
  - FIPA-ACL, getActualSubguis(), childs(componentDescription)
  - …
- Chosen capabilities of actual agent:
  - spaceFunctionality, none
  - …
- Termination conditions: user logout, system termination

User model agent: The exemplified user model agent compromises functionality as described in [27], namely the update of a user model. Within the presented framework it is part of the e-learning services layer.

- Information: UMAgent1, system1, "providing user model related functionality"
- Existence conditions: e-learning services layer
- Parameters: user id, user model database access …
- Chosen communication descriptors:
  - FIPA-ACL, updateInformation(senderComponent, ontology.concept), update(communication)
  - FIPA-ACL, getInformation(ontology.concept), getInfo(communication)
  - …
- Chosen capabilities of actual agent:
  - update, ((addIndividual, UMAgent), (createConcept, UMAgent), (createRelation, UMAgent), (askSupportAgent, supportAgent1), …)
  - addIndividual, ((accessDB, UMAgent), …)
  - createConcept, ((accessDB, UMAgent), (accessOnto, UMAgent), …)
  - createRelation, ((accessDB, UMAgent), (accessOnto, UMAgent), …)
  - mergeOntologies, ((update, UMAgent),(askSupportAgent, supportAgent1), (askSupportAgent, supportAgent2),…)
  - askSupportAgent, ((findAgent, catalogueService), …)
  - …
- Termination conditions: system termination

4 Recent advances in agent-supported e-learning

The basic question when applying a technology is its usefulness. When is it possible and beneficial to integrate it? Milgrom et al. answered this basic question in [36] for the agent-oriented paradigm by defining some guidelines validated by case studies (e.g. [37], [38]). Their argumentation starts with a statement that agent-oriented design and implementation will have its greatest scope of applicability in systems with following characteristics:

- subsystems and subsystem components forming a system;
- high-level interactions between subsystems and subsystem components in terms of size and complexity;
- changing interrelationships over time

Common problems types that can be solved with agent technology where described in [39] and [40]. That may include system characteristics like dynamics, openness, complexity and ubiquitousness as well as problem qualities like physical distribution of components, data and knowledge. Agents can be helpful to solve these problems because of their scalability and their ability to improve latency [1]. The key argument for the limitation of the applicability of agent technology was argued in [36]. The principle of "avoiding overkill" refers to some philosophical background. It mainly concerns to the adjustment of requirements and solution. Not everything that is possible to design with agents should be implemented with it. Otherwise it is a waste of time and effort.

4.1 Application of agent technology in e-learning

For our work of the domain of agent-supported e-learning we did an intensive literature research about possibilities of application of agent technology for several e-learning aspects. Agents are implemented for different reasons and are affecting different target types.
Possible affected targets of processing are the user itself, internal application components, internal databases and external applications (as proxy). Figure 12 elucidates the focus of actual research towards user-centred agent technology for e-learning.

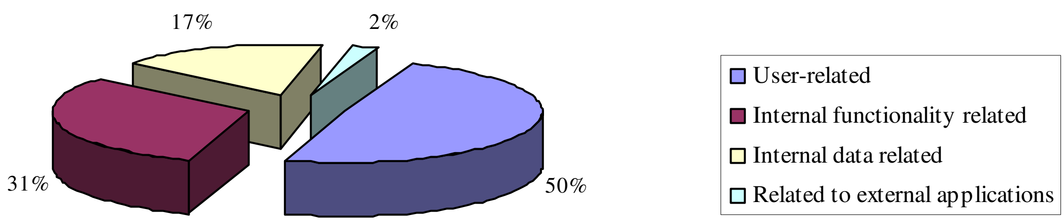

Figure 12. E-learning data artefact coverage by agents

Figure 13 visualises application possibilities of agents for certain types of e-learning functionality. Again user centred functionality is one main aspect for the usage of agents; that refers to e.g. knowledge delivery, notification, motivation and several objectives of human-computer-interfaces in general. Chosen observable targets are the user, learning objects, other knowledge resources and certain system artefacts. The "support" class of functionality summarises aspects like decision taking, recommendations, tutoring and search capabilities. Furthermore agents are used to manage knowledge, system components, learning activities and several aspects of user models, meanwhile another application area for this technology is the processing of several knowledge as for example content, several learning units or evaluation data. Agents are used for adaptation and generation, too.

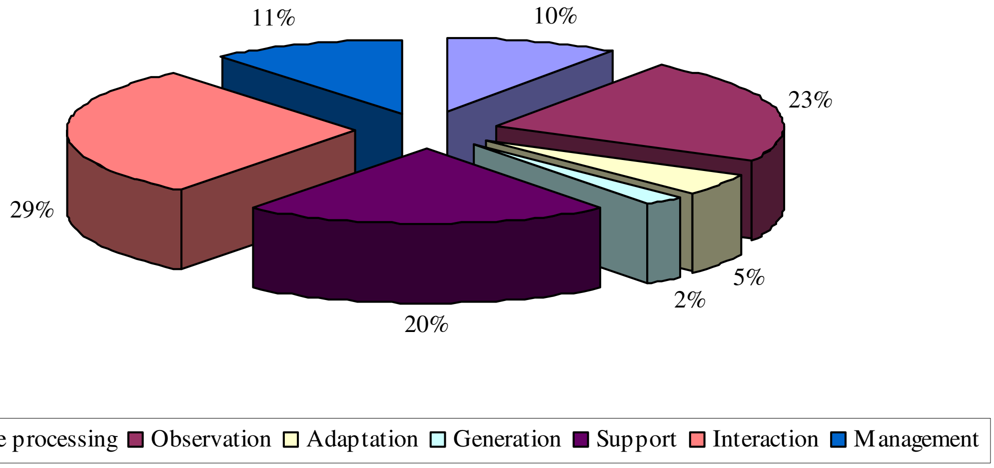

Figure 13. E-learning functionality coverage by agents

In connection to the presented framework we identified a quite well-balanced distribution of agent-based support approaches. The pattern emerges because of the possible application of the approaches in different environments.

Delivery Environment: 52
Working Environment: 69
Learning Unit Environment: 58
Content Environment: 50
Administration Environment: 52
Interaction Environment: 47

E-learning Services Layer: 52
Common Services Layer: 10
Infrastructure Layer: 0

Relations to other platforms: 4

The different pattern among the three services layers has its origin in the specialised e-learning focus of the analysed resources and the increasing fundamental nature of the lower layers.

4.2 Classification of existing approaches for agent-supported e-learning

In this section a classification of chosen existing approaches within the targeted domain of interest is presented.

Platform Presentation Environment:
- Agent-based GUIs for e-Learning [41][42]

Delivery Environment:
- Catalogue browsing:
  - Access and dynamical interpretation of LOs [3]
  - Proactive educational courses in distributed and mobile environments [33]
- Course presentation:
  - Knowledge delivery [1] [7] [43] [44] [45] [46] [47] [48] [49] [50] [51]
  - User guidance/facilitation [1] [7] [45] [51] [52] [54] [55] [56]
  - User motivation [45] [52] [53] [57]
  - Learning path adaptation [58] [59]

Working Environment:

- Scheduler:
    - Proactive class schedule [32]
    - Resource scheduling [60]
- Link/annotation management:
    - Individual portfolio agent [61]
- Repository access:
    - Library agent [61]
- Web access:
    - Web search agent [61]
    - Integration of external office tools [1]

Content Environment:
- Creation:
    - Media conversion [62]
    - Integration of external tools [1]
- Management:
    - Resource management [63] [64]

Learning Unit Environment:
- Assembly
    - Recommendation of LOs [7] [63]
    - Didactic decision taking [60] [65]
    - Hypothesis generation about future user actions [59]
- Strategy authoring
    - Modelling pedagogy [60]

Administration Environment:
- User model:
    - Autonomous management of user models in lifelong learning [27]
    - Management of learning activities [60] [66]
- User management:
    - Evaluation of teaching results [60]
    - Student performance storage and evaluation [61]
- Resource management:
    - Course management [6] [46] [64]
    - Information management [67]
- Infrastructure:
    - Management of system components [68]

Interaction Environment:
    - Message board provision [1]
    - Message delivery [51] [69]
    - Syntax and semantic error detection [70] [71]
    - Broker agent for interaction partner identification [61]

E-learning Services Layer:
    - Knowledge/information/data search [1] [7] [45] [55] [61] [62]
    - Knowledge/information/data processing/generation [62] [64]
    - User profiling [1] [8] [47] [48] [49] [50] [53] [59] [60] [63] [64] [68] [72]

Common Services Layer:
    - Service discovery and negotiation [55] [60]
    - System evaluation [73]
    - Authentication, authorization and encryption [62]

Infrastructure Layer:
    - InfoStation-Based Multi-Agent System Supporting Intelligent Mobile Services Across a University Campus [29]

5 Conclusions and further work

In this paper we have presented a framework describing several components of e-learning systems from an agent-based point of view. Additionally we presented a formal definition of aL-agents for the framework and illustrate it with examples. Thereby and with the framework itself we achieved the intended goal to provide a classification and characterization possibility for the usage of agent within the domain of e-learning. We concluded with the presentation of the actual usage to show the up-to-dateness of this crossover domain.
The proposed set of services, which are realizable by agent-technology, is definitely not exhaustive, but it constitutes a basis for discussion and subsequent work in this field. Other, more technical aspects should be taken into account, too.

References:

[1] Anghel, C.; Salomie, I.: JADE Based Solutions for Knowledge Assessment in eLearning Environments EXP - in search of innovation (Special Issue on JADE), 2003.

[2] Juneidi, S.J.; Vouros, G.A.: Engineering an E-learning Application using the ARL Theory for Agent Oriented Software Engineering, 2005 AAAI Fall Symposium, MIT press, 2005.

[3] Nwana, H.S.; Ndumu, D.T.: A Brief Introduction to Software Agent Technology Agent Technology: Foundations, Applications, and Markets, Springer-Verlag New York, Inc., pp. 29-47, 1998.

[4] Wilson, S.; Olivier, B.; Jeyes, S.; Powell, A.; Franklin, T.: A Technical Framework to Support e-Learning, Technical Repost JCLT, 2004.

[5] Cerri, S.A.: Human and Artificial Agent's Conversations on the GRID, Proceedings of 1st LEGE-WG International Workshop on e-Learning and GRID Technologies: Educational Models for GRID Based Services, 2002.

[6] van Rosmalen, P.; Brouns, F.; Tattersall, C.; Vogten, H.; van Bruggen, J.; Sloep, P.; Koper, R.: Towards an Open Framework for Adaptive, Agent-Supported E-learning International Journal of Continuing Engineering Education and Lifelong Learning, vol. 15, pp. 261-275, 2005.

[7] Pankratius, V.; Sandel, O.; Stucky, W.: Retrieving Content with Agents in Web Service E-Learning Systems, The Symposium on Professional Practice in AI, IFIP WG12.5 - First IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI), 2004.

[8] Giotopoulos, K.C.; Alexakos, C.E.; Beligiannis, G.N.; Likothanassis, S.D.: Integrating Agents and Computational Intelligence Techniques in E-learning Environments, International Enformatika Conference (IEC'05), pp. 231-238, 2005.

[9] Pankratius, V.; Sandel, O. & Stucky, W.: Retrieving Content with Agents in Web Service E-Learning Systems, The Symposium on Professional Practice in AI, First IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI), 2004.

[10] Wooldridge, M.J.; Jennings, N.R.: Agent Theories, Architectures, and Languages: A Survey Intelligent Agents: ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Springer, pp. 1-39, 1994.

[11] Franklin, S.; Graesser, A.C.: Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents, Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, pp. 21-35, 1996.

[12] Castelfranchi, C: To Be or Not To Be an Agent, Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, pp. 37-39, 1996.

[13] Sánchez, J.A.: A Taxonomy of Agents, Technical Report Universidad de las Américas-Puebla, México, 1997.

[14] Russell, S.; Norvig, P.: Artificial Intelligence: A Modern Approach, Prentice Hall, 1995.

[15] Foundation for Intelligent Physical Agents (FIPA): FIPA Agent Management Specification, Foundation for Intelligent Physical Agents, 2006

[16] Wooldridge, M.; Jennings, N.R.: Intelligent Agents: Theory and Practice, Knowledge Engineering Review, vol. 10, pp. 115-152, 1995.

[17] Hayzelden, A.L.; Bigham, J. (ed.): Software Agents for Future Communication Systems, Springer-Verlag New York, Inc., 1999.

[18] IEEE LTSC: IEEE P1484.1/D9 Draft Standard for Learning Technology: Learning Technology Systems Architecture (LTSA), Draft, 2002.

[19] IMS Global Learning Consortium, Inc.: IMS Abstract Framework: White Paper, IMS Global Learning Consortium, Inc., 2003.

[20] SUN Microsystems: E-learning Framework, SUN Microsystems, 2003.

[21] O.K.I. Initiative: O.K.I. Architectural Concepts, Massachusetts Institute of Technology, 2003.

[22] Advanced Distributed Learning (ADL®): Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Overview, Advanced Distributed Learning (ADL®), 2006.

[23] Kernchen, S.: Adaptive Web-Content in the Context of Lecture Notes, Diploma Thesis Otto-von-Guericke University of Magdeburg, 2005.

[24] Brown, J.S.; Duguid, P.: The Social Life of Information, Harvard Business School Press, 2000.

[25] Belanger, F.; Jordan, D.H.: Evaluation and Implementation of Distance Learning: Technologies, Tools and Techniques, Idea Group Publishing, 2000.

[26] SUN Microsystems, Inc.: Whitepaper: e-Learning Application Infrastructure, SUN Microsystems, Inc., 2002.

[27] Kernchen, S.; Dumke, R.: Autonomous Management of User Models for Lifelong Learning, IEEE Multidisciplinary Engineering Education Magazine, vol. 2, no. 2, pp. 44-50, 2007.

[28] Schools Interoperability Framework (SIF™) Association: Implementation Specification 2.0r1 Schools Interoperability Framework (SIF™) Association, 2007.

[29] Ganchev, I.; Stojanov, S.; O'Droma, M.; Meere, D.: An InfoStation-Based Multi-Agent System Supporting Intelligent Mobile Services Across a University Campus, Journal of Computers, vol. 2, no. 3, pp. 21-33, 2007.

[30] Foundation for Intelligent Physical Agents (FIPA), FIPA Abstract Architecture Specification, Foundation for Intelligent Physical Agents (FIPA), 2002.

[31] Wilson, S.; Blinco, K.; Rehak, D.: An e-Learning Framework - A Summary, DEST and JISC-CETIS and Industry Canada, 2004.

[32] Kernchen, S.; Kunz, M.; Dumke, R.: Proactive Class Schedule, Proceedings of the International Conference on Interactive Computer Aided Blended Learning, Florianópolis, Brazil, May 07 -09, 2007.

[33] Kernchen, S.; Dumke, R.: Proactive Educational Courses in a Distributed Environment, Proceedings of the International Technology, Education and Development Conference (INTED2007), Valencia, Spain, March 07 - 09, 2007.

[34] MOBIlearn Project Consortium, MOBIlearn - Final Project Report, MOBIlearn Project Consortium, 2005.

[35] Foundation for Intelligent Physical Agents (FIPA), FIPA Ontology Service Specification, Foundation for Intelligent Physical Agents (FIPA), 2001.

[36] Milgrom, E.; Chainho, P.; Deville, Y.; Evans, R.; Kearney, P.; Massonet, P.: MESSAGE - Methodology for Engineering Systems of Software Agents (Final Guidelines for the Identification of Relevant Problem Areas where Agent Technology is Appropriate), EURESCOM, 2001.

[37] Chainho, P.; Massonet, P.; Garijo, F.; Sanz, J.G.; Pavon, J.; Caire, G.; Leal, F.; Stark, J.: Analysis of the UPA Case Study (PIR4.2 Case Study 1 B1), EURESCOM, 2000.

[38] Kearney, P.; Stark, J.; Leal, F.; Rodrigues, J.P.: Analysis of the ACSOSS Case Study (PIR 4.2 B.0), EURESCOM, 2000.

[39] Jennings, N.R.; Wooldridge, M.J.: Agent Technology - Foundation, Application, and Markets Springer Publ., 1998.

[40] Ferber, J.: Multi-Agent Systems Addison Wesley, 1999.

[41] Kernchen, S.; Zbrog, F.; Dumke: ABEL-GUI: An Agent-Based Graphical User Interface for E-Learning, Proceedings of the 3rd International Conference on Web Information Systems and Technologies (WEBIST 2007) - Volume SeBeG/eL, Barcelona, Spain, March 03 – 06, pp. 491-494 2007.

[42] Kernchen, S.; Dumke, R.: Developing Adaptive and Self-Managed Graphical User Interfaces, Proceedings of the Second International Conference on Interactive Mobile and Computer Aided Learning (IMCL 2007), Amman, Jordan, April 18 – 20, 2007.

[43] Mohammed, P.; Mohan, P.: Agent Based Learning Objects on the Educational Semantic Web, Proceedings of the 2nd Annual e-Learning Conference on Intelligent Interactive Learning Object Repositories (I2LOR-2005), Vancouver, Canada, November 16 – 18, 2005.

[44] Silveira, R.A.; Gomes, E.R.; Pinto, H.; Vicari, R.: Intelligent Learning Objects: An Agent Based Approach of Learning Objects, Proceedings of the 7th International Conference in Intelligent Tutoring Systems, Maceio, Brazil, pp. 886 – 888, 2004.

[45] Schlenker, M.; Mimkes, J.; Hilf, E.R.: Dynamic Thesaurus and Dynamic Discovery of Distributed eLearning Materials, Proceedings of the 10th International Conference of European University Information Systems, Ljubljana, Slovenia, 2004.

[46] Jafari, A.: Conceptualizing Intelligent Agents for Teaching and Learning, EQ, Educause Quaterly, pp. 28 – 34, 2002.

[47] Kabassi, K.; Virvou, M.: Learner Modelling in a Multi-Agent System through Web Services, Proceedings of 2003 IEEE International Conference on Advanced Learning Technologies: Technology Enhanced Learning (ICALT 2003), Athens, Greece, July 09 – 11, pp. 115 – 119, 2003.
[48] Belkada, S.; Cristea, A.I.; Okamoto, T.: Measuring Knowledge Transfer Skills by Using Constrained-Student Modeler Autonomous Agent, Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2001), Madison, USA, pp. 375 – 379, August 6 – 8, 2001.
[49] O'Riordan, C.; Griffith, J.: A Multi-Agent System for Intelligent Online Education, Journal of Interactive Learning Research, vol. 10, no. 3/4, pp. 263 – 274, 1999.
[50] Soh, L.; Miller, L.D.; Blank, T.; Person, S.: ILMDA: Intelligent Learning Materials Delivery Agents, IEEE International Conference on Electro Information Technology, Lincoln, USA, May 22 – 25, 2005.
[51] Yang, C.: Intelligence in MAS-Based Distributed Learning Environments, In: Designing Distributed Learning Environments with Intelligent Software Agents, Idea Group Publishing, 2005.
[52] Webber, C.; Bergia, L.; Pesty, S.; Balacheff, N.: The Baghera Project: A Multi-Agent Architecture for Human Learning, Proceedings of the Workshop Multi-Agent Architectures for Distributed Learning Environments (AIED 2001), San Antonio, USA, pp. 1060 – 1069, May 19 – 23, 2001.
[53] Angehrn, A.; Nabeth, T.; Roda, C.: Towards Personalised, Locially Aware and Active e-Learning Systems - Illustrated with the Agent-Based System K-InCA, CALT White Paper, November 2001.
[54] Chen, W.; Wasson, B.: Intelligent Agents Supporting Distributed Collaborative Learning, In: Designing Distributed Learning Environments with Intelligent Software Agents, Idea Group Publishing, 2005.
[55] Chen, P.; Ding, W.: Knowledge Management for Agent-Based Tutoring Systems, In: Designing Distributed Learning Environments with Intelligent Software Agents, Idea Group Publishing, 2005.
[56] Lin, F.; Esmahi, L.; Poon, L.: A Human Collaborative Online Learning Environment Using Intelligent Agents, In: Integrating Agents and Web Services into Adaptive Distributed Learning Environments, Idea Group Publishing, 2005.
[57] Angehrn, A.A.: Computers that Criticize You: Stimulus-Based Decision Support Systems, Interfaces, vol. 23, no. 3, pp. 3 – 16, 1993.
[58] Garro, A.; Leone, N.; Ricca, F.: Logic Based Agents for E-learning, Proceedings of 18th International Joint Conference on Artificial Intelligence (IJCIA 2003), Acapulco, Mexico, pp. 36 – 45, August 09 – 15, 2003.
[59] Roda, C.; Angehrn, A.A.; Nabeth, T.; Razmerita, L.: Using Conversational Agents to Support the Adoption of Knowledge Sharing Practices, Interacting with Computers, vol. 15, no. 1, pp. 57 – 89, 2003.
[60] Shi, H.; Shang, Y.; Chen, S.: A Multi-Agent System for Computer Science Education, Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2000), Helsinki, Finland, pp. 1 – 4, July 11 – 13, 2000.
[61] de Azevedo, H.J.S.; Scalabrin, E.E.: A Human Collaborative Online Learning Environment Using Intelligent Agents, In: Designing Distributed Learning Environments with Intelligent Software Agents, Idea Group Publishing, 2005.
[62] Kargl, F.; Illmann, T.; Weber, M.: CIA - a Collaboration and Coordination Infrastructure for Personal Agents, Proceedings of the IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems II. (DAIS 1999), Helsinki, Finland, pp. 213 – 218, June 28 – July 1, 1999.
[63] McGreal, R.; Anderson, T.; Hubick, C.; Lin, F.; Sosteric, M.; Tin, T.; Yasynska, O.: Case Study: eduSource and the Athabasca University Digital Library Project, E-Learning News, vol. 1, no. 1, 2005.
[64] Koyama, A.; Barolli, L.; Tsuda, A.; Cheng, Z.: An Agent-Based Personalized Distance Learning System, Proceedings of the 15th International Conference on Information Networking (ICOIN'01), Beppu City, Japan, pp. 895 – 899, January 21 – February 02, 2001.
[65] Ritter, S.; Koedinger, K.R.: An Architecture for Plug-In Tutor Agents, Journal of Artificial Intelligence in Education, vol. 7, pp. 315 – 347, 1996.
[66] Shi, H.; Revithis, S.; Chen, S.: An Agent Enabling Personalized Learning in e-Learning Environments, Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002), pp. 847 – 848, 2002.
[67] Andronico, A.; Carbonaro, A.; Casadei, G.; Colazzo, L.; Molinari, A. & Ronchetti, M. Integrating a Multi-Agent Recommendation System into a Mobile Learning Management System, Proceedings of the 5th Conference in Artificial Intelligence in Mobile System (AIMS 2003), Seattle, USA, October 12 – 15, 2003.
[68] Virvou, M.; Katsionis, G.: Web Services for an Intelligent Tutoring System that Operates as a Virtual Reality Game, IEEE International Conference on Systems, Man and Cybernetics, Washington, D.C., USA, pp. 378 – 383, October 05 – 08, 2003.
[69] Smolle, P.; Sure, Y.: FRED: Ontology-based Agents for Enabling E-Coaching Support in a Large Company, Second International Workshop on Ontologies in Agent Systems (OAS 2002), held at the 1st International Conference on Autonomous Agents & Multiagent Systems, Bologna, Italy, July 15, 2002.

[70] Wang, Y.; Wang, W.; Lin, C.: An Intelligent Semantic Agent for e-Learning Message Communication, Journal of Information Science and Engineering, vol. 21, no. 5, pp. 1031 – 1051, 2005.
[71] Ayala, G.; Yano, Y.: Learner Models for Supporting Awareness and Collaboration in a CSCL Environment, Intelligent Tutoring Systems, Springer Verlag, pp. 158 – 167, 1996.
[72] e Paiva, A.A.: Communicating with Learner Modeling Agents, Architectures and Methods for Designing Cost-Effective and Reusable ITSs, 1996.
[73] Markham, S.; Ceddia, J.; Sheard, J.; Burvill, C.; Weir, J.; Field, B.; Sterling, L.; Stern, L.: Applying Agent Technology to Evaluation Tasks in e-Learning Environments, Proceedings of the Exploring Educational Technologies Conference, Melbourne, Australia, pp. 1 – 7, July 16 – 17, 2003.

Author(s):

Steffen Mencke, Dipl.-Inform.
Otto-von-Guericke University of Magdeburg, Department of Computer Science
P.O. Box 4120, 39016 Magdeburg
mencke@ivs.cs.uni-magdeburg.de

Reiner R. Dumke, Prof. Dr.-Ing. habil.
Otto-von-Guericke University of Magdeburg, Department of Computer Science
P.O. Box 4120, 39016 Magdeburg
dumke@ivs.cs.uni-magdeburg.de