Towards an Advanced Modeling System applying a Service-based Approach

Christian Gütl, Victor Manuel García-Barrios

Institute for Information Systems and Computer Media, Graz University of Technology, Austria
{cguetl, vgarcia}@iicm.edu

Abstract

Based on the overall aims of the AdeLE (Adaptive e-Learning with Eye-Tracking) project, and in particular motivated by the decision of using a strictly separated system's architecture, the requirements for the modeling system and its implementation as a micro-service-oriented architecture is addressed in this paper. Furthermore, experiences and lessons learned as well as an approach for a more advanced modeling system in the context of the AdeLE research project is discussed.

1. Introduction

Adaptive e-learning approaches are not new and date back to the early 1960s, as discussed in [1]. However, from our point of view a number of open problems and requirements are not considered sufficient by early as well as contemporary solutions. This brought us to initiate the four-year research project AdeLE (for details refer to [2]). The main specific objective of AdeLE is to provide a platform-independent, flexible and open solution. In particular, an easy integration of existing *Learning Management Systems (LMS)* with less implementation efforts is a strong requirement. The main general objective is to improve the learning process by well tailored learning activities through the usage of a fine-grained user profile. For this purpose, user behavior information gained by an eye-tracking system is linked with content tracking information at the fine-grained level of learning assets, e.g. the system may track that the user is observing the first graphic of the content page.

A short introduction into the AdeLE architecture (see Fig. 1), which is built up of strong separated client-side and server-side systems, is given in this section, however the emphasis is focused on the requirements for our *Modeling System (MS)*.

On the client-side, the *Web Client (WC)* provides content, control and navigation elements. The *Real-time Eye-Tracking System (ES)* reads eye movements and delivers dynamic user behavior data. Eye-tracking information is linked with fine-grained content-tracking information of learning assets processed by the *Real-Time Content-Tracking System (CS)*. On the server-side, the LMS compiles the learning content from the learning object repository as well as provides control and navigation interfaces. Adaptation of content, navigation and visualization is controlled by the AS by exploiting information from the MS.

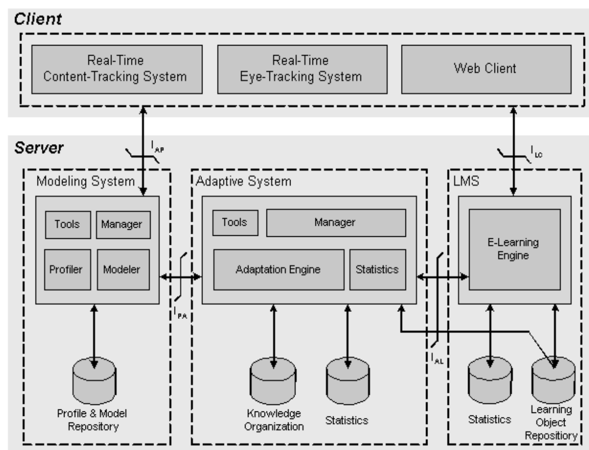

Fig. 1: Simple Architecture Overview of AdeLE

From the functional perspective, the main MS tasks are (1) to manage user information from ET and CT as well as from user interactions and adaptation processes, and (2) to generate assumptions about the user and deliver profiler insights. From the logical perspective, a *Profiler* unit subsumes simple functions (e.g. user traits representation or user behavior logging) and a *Modeler* unit comprises enhanced functionalities (e.g. logical reasoning). In addition, user profile information should be also set, inspected and updated by the learner itself, i.e. the MS should be *scrutable* [3]. The MS may also *proactively* initiate interactions with the user, e.g. a 'new' learner model requires explicit user feedback by proactively delivered form-based questionnaires.

Thus, the main requirements for the first implementation of the MS are: (1) basic features, such as recording of user behavior or management of facts and assumptions, (2) logical separation of learner profiles and models, (3) separated management of user traits and states, (4) data acquisition by sensor components, (5) scrutinity, and (6) proactive feedback.

3. Modeling System Implementation

In order to fulfill the requirements depicted in the last section, we have decided to develop MS and AS on a *Service-Oriented Architecture (SOA)* and to use two different techniques: *micro-service approach*, i.e. the system consists of small and specialized services, and *macro-service approach*, i.e. reducing the number of services by grouping them into functionally related main modules. The framework for the implementation is based on the *Openwings* specification (see [4]).

Our first modeling system is implemented as a micro-service approach, as illustrated in Fig. 2 (labeled circles represent micro-services, an arrow tip denotes a service provider interface and its opposite a service consumer interface). Services are grouped into logical components according to their context: *Tools* encompass auxiliary services, *Manager* embraces services for system's control, and *Profiler and Modeler* comprise services for particular purposes.

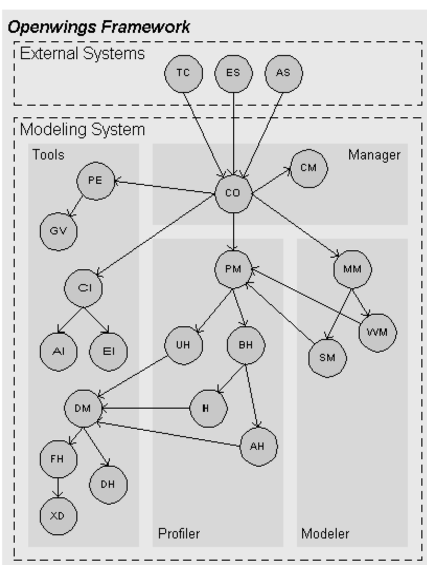

Fig 2: Architecture of the Modeling System

A flexible communication between MS and an *external system (ExS)* is enabled by the *Communicator (CO)* through the application of data structures encapsulated in a specific *Communication Object Instance (COI)*. For further processing, CO delegates the COI to the *Communication Interpreter (CI)*, which parses the particular data structure by means of a corresponding auxiliary service; i.e. the COI of the external *Adaptive System (AS)* service is handled by the *Adaptive System Communication Interpreter (AI)*, and the *Eye-Tracking System (ES)* service is handled by the *Eye-Tracking Communication Interpreter (EI)*. The parsed data structure is returned in an internal standardized representation to CO and sent to a *Context Manager (CM)*. The CM determines the context of the given data structure and delegates it for further processing either to the *Profile Modeler (PM)* or to the *Modeler Manager (MM)*, and the result is returned to a corresponding ExS. Further, specialized tools services may prepare particular result representations for ExS's, e.g. the *Profiler Editor (PE)* prepares a particular visualization by means of the auxiliary service *GUI Visualization (GV)* for the *Test Client (TC)* service.

Tasks in the profiler context are controlled by the PM and delegated for further processing to more specialized services. To clarify the principle, for example "user information records" are handled by the *User Information Handler (UM)* by means of tools services specialized in data management. Thus, *Data Manager (DM)*, *File Handler (FH)* and *XML File Handler (XH)* manage corresponding XML file structures. To give another example, user behavior information is handled by the *Behavior Handler (BH)*, and from here delegated either to the *Instruction Handler (IH)* or to the *Action Handler (AH)*. This information is managed in a database by means of DM and the *Database Handler (DH)*.

As already mentioned, in the modeler context more advanced functionalities are addressed than in the profiler context. The topology of the system reflects this by the fact, that modeler services itself use PM as auxiliary services, e.g. *States Modeler (SM)* handles user states and *WAVI Modeler (WM)* the Wholist-Analyst-Verbaliser-Imager cognitive model.

4. Lessons Learned and Future Work

Focusing on the subject of modeling systems let us give a short overview over the most relevant findings regarding the fields of SOA and adaptive e-learning.

The strict separation of functional and logical sub-systems corresponds to our general view on adaptive e-learning environments as extensible systems of tools supporting both, the learning and the teaching process. Further, the open and flexible architecture facilitates the application of different e-learning paradigms as well as the usage of external tools for learning activities (e.g. collaborative tools) because it is implemented as a general-purposed adaptation system.

Thus, different modeling techniques are may be added by inserting new services, e.g. in order to maintain separate but interrelated learner models, user groups, courses, knowledge domains or target output devices. In addition, the MS enables the integration of semantic technologies (e.g. ontologies may be used to generate taxonomical profilers or concept maps).

The most critical technical problem could be found in the application of the micro-service approach, because each additional functional module must be implemented as an atomic service. This fact may lead to communication overload (if the traffic of messages is too high) and to security problems (as each service must be defined with open interfaces). In order to provide a secure control access to services, AdeLE uses two separate Openwings platforms for MS and AS and utilizes self-defined user roles. Another technical obstacle arose with the separated treatment of user states that vary continuously, such as real-time eye-tracking data. The management of user states on run-time implies the application of relatively complex algorithms for consistent and synchronous updating and monitoring of models. Thus, the MS must be fast enough to simulate real-time snapshots of models.

Finally, let us state that separating Profiler and Modeler represents an advantage, because it enables not only to separate facts from assumptions, but also to deliver various views on and accesses to the MS via different PEs (i.e. enhancement towards scrutinity).

As can be concluded, the end of our tour towards an Advanced Modeling System (AMS) is not reached yet. The IICM will work further on developing a general-purpose MS for various application scenarios. In summary, at least the following issues should be considered in order to realize our idea of the AMS:

(1) Openness should be enhanced by the usage of what we call *functionlets*, which are pluggable external services, which on the one hand, may extend the functionality of the MS by means of proactive data collection and processing in an application-dependent manner, and on the other hand, enable external systems to source out modeling or profiling functionality.

(2) Critical issues concerning privacy policies of general models require more advanced security features in order to distinguish between trusted and untrusted external applications.

(3) Our MS should also manage many kinds of user state models, where each additional sensor may capture huge amounts of data in very short periods of time. Thus, the minimal solution is to use advanced Load-Balancing and Task-Scheduling services.

(4) A simple macro-service solution of the MS will be implemented in order to compare many technical and functional issues in context. We expect to be able to prove that the best solution should be a combination of component- and service-based technologies.

Finally and as a matter of completeness, let us give a summary of some issues, which distinguish the AdeLE MS from other (user) modeling systems:

- *Generic user modeling systems* and *shells* (see [5]) focus on user modeling, whereas our solution is a more general approach for managing separately models of different types (like domains, devices, courses, etc.).

- In agreement with [3] we also consider *scrutability of (user) models* to be a critical issue, because it involves the controlled visibility of the model. Our approach goes further in the context of application- and task-specific read/write access for each external system.

- Unlike most system solutions in other fields, e.g. Adaptive Hypermedia Systems (AHS) or recommender systems, our MS is developed as an *independent system* and not integrated as part of an adaptive engine.

5. Acknowledgements

The AdeLE project is partially funded by the Austrian ministries BMVIT and BMBWK, through the FHplus impulse programme. The support of the following institutions is gratefully acknowledged: Department of Information Design, Graz University of Applied Sciences (FH JOANNEUM); Institute for Information Systems and Computer Media, Faculty of Computer Science at Graz University of Technology.

6. References

[1] F. Mödritscher, V.M. García-Barrios, and C. Gütl, “The Past, the Present and the Future of adaptive E-Learning. An Approach within the Scope of the Research Project AdeLE”, Proc. ICL 2004, Villach, Austria, 2004

[2] C. Gütl, V.M. García-Barrios, and F. Mödritscher, “Adaptation in E-Learning Environments through the Service-Based Framework and its Application for AdeLE”, Proc. E-Learn 2004, Washington, USA, 2004, p. 1891-1898

[3] J. Kay, B. Kummerfeld, D. Carmichael, and A. Quigley, “The Scrutable Personalised Pervasive Computing Environment”, Proc. IUI, San Diego CA, USA, 2005

[4] OPENWINGS, Website, General Dynamics Decision Systems, last visit 2005-01-30, http://www.openwings.org

[5] A. Kobsa, “Generic User Modeling Systems”, User Modeling and User-Adapted Interaction, Kluwer Academic Publishers, 2001, p. 49-63