An adaptative framework for tracking Web–based Learning Environments

Valentin Butoianu, Philippe Vidal, Julien Broisin

Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier
118 route de Narbonne
31062 Toulouse, France
{butoianu,vidal,broisin}@irit.fr

Abstract: Collecting and sharing attention information represents a main concern within the Technology Enhanced Learning community, as the number of works or projects related to this topic demonstrates it. Attention data are produced by various systems exploited by TEL actors during a learning session, thus raising a difficulty when it comes to collect traces translating all activities operated by learners, teachers or tutors. Thus, we propose an open framework based on a standardized and widely adopted approach to manage systems, networks and application. Our proposal is able to self-adapt to any attention information related to learning systems, resources and activities, and can meet the requirements of any web-based recommender systems. An implementation validates our approach by collecting and sharing attention data resulting from the manipulation of learning objects through heterogeneous applications.

1 Introduction

Providing useful support to participants during the learning process has become one of the major research concerns in the field of Technology Enhanced Learning (TEL). Instructors and designers need to improve their course designs in order to efficiently adapt their courseware to various users' learning paths [ZA02] [MA07a], whereas learners need help to find the learning materials that best suit their needs [OT05]. To achieve this goal, a lot of researchers try to exploit tracking information in order to allow teachers to draw conclusions about their students' learning curve, or to bring adaptations to a learning scenario according to a given learner [KE07].

To achieve the above objectives, there is a need for collecting attention information related to activities handled by TEL actors on a huge number of heterogeneous systems. Indeed, most of learners and teachers use to launch various types of web-based or desktop tools to achieve their learning objectives. Even if attention data that may be collected from heterogeneous systems appear disparate and unrelated, pedagogical processes such as personalization, reengineering, or intelligent tutoring should benefit from the most important number of attention information to provide an efficient learning support.

We propose here a framework able to gather tracking information produced by any web-based tools. This approach stands on an existing standard dedicated to system, network and application management. By extending the Common Information Model (CIM) [CIM98], we suggest some generic models allowing for representation of any attention data, together with a distributed architecture that ensures collect and storage of tracking information. Attention data specific to various learning tools are thus represented as a unified structure, and stored into a central repository. To facilitate access to this tracking repository and promote share and reuse, we introduce two dynamic services: one allows users to define attention data they want to collect, while the other is dedicated to receive traces sent by learning systems.

The paper is structured as follows: section 2 presents the context of these works by exposing some related works in order to introduce our approach. Section 3 represents the core chapter, since it describes both our generic attention models together with the dynamic services able to self-adapt to heterogeneous traces. To validate our approach, section 4 exposes an implementation and demonstrates how usage of learning objects can be tracked from two distinct learning systems. Finally, conclusions and future works are provided in section 5.

2 Context

[MA07b] propose a generic system for tracking users interacting with discussion forums. The attention information concerns: users and their role (i.e. learner, instructor, etc.), activities (i.e. browse forum, exchange messages on forum, etc.) and activity category (i.e. traces of users' activities on forum, traces of users' interactions on remote workstation, etc.). [ZH07] implemented a Learning Management System (LMS) log analysis tool called Moodog for tracking students' online learning activities. The goals of Moodog are to provide instructors with an overview about how students interact with online course materials, and to allow students to compare their progress with others students in the class. The data stored in Moodog is: data about the course, data about the students, data about resources, and data about access time.

The main drawback of the two above approaches is the specificity. Indeed, the first one focuses on users' activities related to discussion forums, while the other deals with users' activities connected to a LMS.

To tackle this issue, [NA05] exploits the Attention.XML specification [AT04] for describing usage of learning resources within various applications such as internet browsers, messaging systems, email clients, etc. This approach is used to enhance users' models, predict usage patterns and provide recommender systems with attention information. Since Attention.XML specifies only basic elements about users' attention (duration, last read, title, etc.), [NA06] extended this proposal in order to take into account information about user activities, applications on which these actions took place, and the context where resources were used. This framework thus brings a significant advantage, since the attention model is extensible and able to integrate any traces resulting from a learning activity on a resource. However, a specific system such as an XML database has to be deployed in order to store attention data.

Most of today operating systems natively integrate a tool to supervise logical and physical entities composing a computer and to allow their local and remote management. In other words, it extracts and stores the evolution of the file system, hardware components, processes, etc. This framework is based on the de facto WBEM (Web-Based Enterprise Management) standard [WBEM99] elaborated by the Distributed Management Task Force (DMTF) [DMTF99]: the Microsoft™ operating systems embed WMI [WMI00], whereas some Linux distributions such as RedHat natively integrate OpenWBEM [OWBM01]. Therefore, our approach consists in exploiting this framework to collect, store and reuse attention information related to TEL environments.

3 Modeling and managing tracking data

WBEM stands on a Common Information Model (CIM) to represent data to supervise that is supported by a distributed architecture ensuring scalability and interoperability. Figure 1 illustrates our global architecture based on the WBEM standard and divided in three parts:

- The first one represents learning systems: users' activities are collected from these tools.

- The tracking environment is composed of two WBEM components: a tracking repository is responsible for storing attention information, whereas a tracking manager is able to manipulate traces stored into the repository.

- The intermediate layer between the learning and the tracking environments offers an easy access to the tracking repository: indeed, WBEM components are often restricted to administrators because they store confidential information that should not be open to any users or applications. Through the services of this bridge, learning tools are able to easily provide and/or retrieve traces stored into the repository.

Since this architecture is presented in [BR06], we focus now on the model of traces that extends the native CIM model to represent attention information dedicated to TEL.

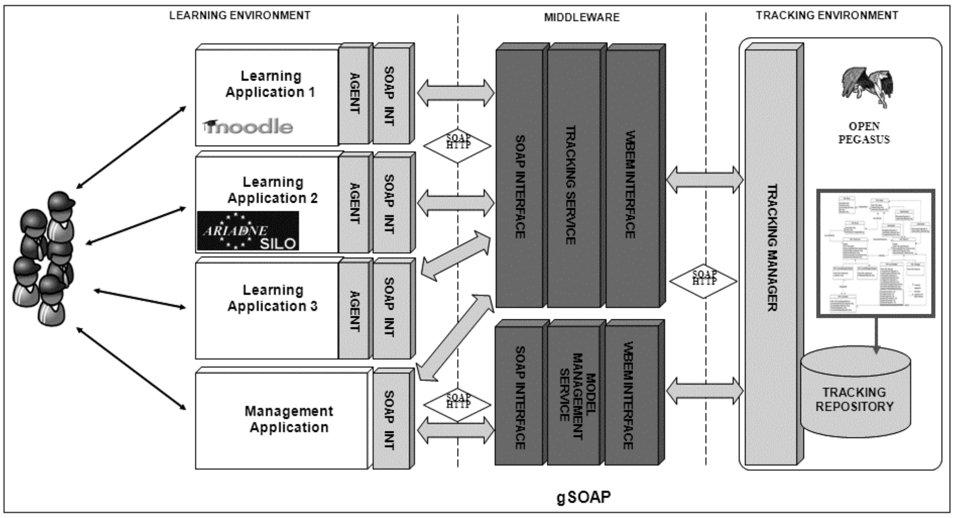

Figure 1: The global architecture

3.1 Some generic models

The tracking model must be able to take into account any attention metadata related to users, learning environments and resources, and activities. Therefore, we defined two generic models composed of two sub models: the model TEL_Environment (see Figure 2) focuses on learning systems and resources, whereas the model TEL_Activity (see Figure 3) aims at describing interactions of users with these systems and resources. Each of these models presents a high abstraction level, and offers the opportunity to define specific models according to some specific objectives. This section only exposes the higher models, since the specific models are related to the implementation presented in section 4. Moreover, the user profile is represented on Figure 2 as the class CIM_Identity; it is precisely detailed in [RA09], and includes the IMS-LIP standard [IMS01] together with some additional information.

The main classes of the generic TEL_Environment model are TEL_ ApplicationSystem and TEL_Resource; they respectively model any learning systems and resources. Since these systems/resources can be composed of others systems/resources, we introduced two composition relationships (respectively TEL_SystemComponent and TEL_ResourceComponent). In addition, another composition (TEL_SystemResourceComponent) expresses the fact that a system hosts resources. Finally, in order to link a user with a system/resource, we designed the associations TEL_IdentityOnSystem and TEL_IdentityOnResource.

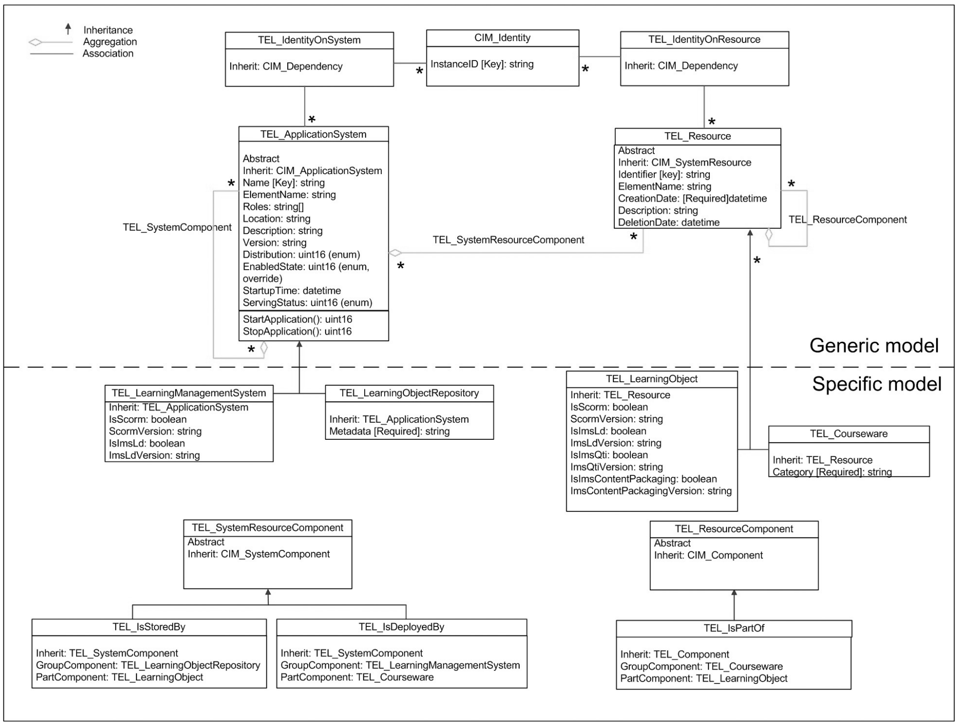

Figure 2: The TEL_Environment model

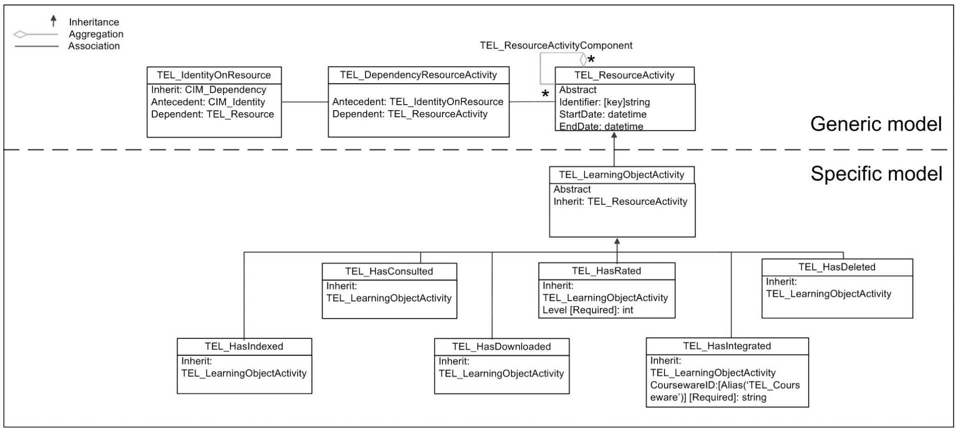

Figure 3: The TEL_Activity model for resources

Figure 3 only depicts activities related to resources. However, the generic model dedicated to activities operated on systems follow a similar reasoning. To identify an activity processed by a user on a resource, we introduced the association TEL_DependencyResourceActivity. Activities are represented through the root class TEL_ResourceActivity, and compositions between activities are expressed by the composition TEL_ResourceActivityComponent.

The generic models of traces exposed here offer a high abstraction level allowing for representation of any attention data including users, systems, resources and activities. To specialize these generic models according to specific pedagogical objectives, the next section introduces some services that facilitate definition of attention data to collect on one hand, and to manage these traces on the other hand.

3.2 Some dynamic services

The model management service

When specific data must be tracked, there is need for specializing the generic models. Until now, the only way to achieve this task was to manually compile classes and properties matching with attention information to supervise; to prevent this tedious step, we introduced the model management service (see Figure 1).

Some of the methods process on classes, while others work on attributes. The whole set of methods includes common data transactions required to manage data: insertion of a new class or a new attribute into an existing class, update of an existing class or attribute, and deletion of an existing class or attribute. However, some restrictions apply: one can only define new classes inheriting from the generic environment or activity models, and one can't modify the specifications of the generic models.

Of course, this service also offers the opportunity to retrieve the global tracking model. Basically, it returns as an XML string the environment model, the activity model, and the user model [RA09]. One can thus discover traces that are made available.

The model management service ensures extension of the model, and thus allows specification of any attention data. The service presented in the next section gathers traces resulting from users' activities implied within TEL environments, and insert the matching instances into the tracking repository.

The tracking service

The tracking service is composed of two methods: one is responsible for inserting traces into the tracking repository (through the tracking manager), while the other is able to retrieve attention information stored into the repository.

The *SetTrace* method is able to receive traces produced by any web-based system, and to build or modify the matching instances defined within the tracking model. In order to accomplish these tasks, the service has to take into account both the generic and the specific models. Therefore, this method queries the tracking manager to get the class' definitions and then builds the matching XML schema. Through this mechanism, this method is able to process any trace compliant with the global tracking model.

The *GetTrace* method explores the tracking repository in order to get a unified view of all traces (in other words, it retrieves all instances stored into the repository). Treatments associated to this method consist in querying the tracking manager to retrieve CIM and TEL instances of the model using the CIM Query Language [CQL04] elaborated by the DMTF. The service is dynamic so that even if new classes or attributes are defined into the model, they will be retrieved and returned to computer systems or users. Let us note that this service is able to return the trace according to a predefined format such as Attention.XML.

To validate our approach, a framework has been implemented to track usage of learning objects within two heterogeneous systems: the Learning Object Repository (LOR) of the ARIADNE foundation [ARN96] and one of the most popular LMS called MOODLE [MDL02]. Therefore, the next section details the specific models that integrate attention data related to these systems and resources, then introduces the software deployed within our implementation, and finally exposes a use case.

4 Implementation

4.1 The models specific to LOR, LMS and LO

The models specific to learning objects, LOR and LMS are described on the inferior part of Figure 2. Two types of system have been defined (TEL_LearningManagementSystem and TEL_LearningObjectRepository) and inherit from the root class to represent a TEL application, whereas two types of resource (TEL_LearningObject and TEL_Courseware) specialize the higher class to model a learning resource. Moreover, some compositions illustrated at the bottom of Figure 2 express the fact that (1) a learning object is stored into a LOR (class TEL_IsStoredBy), (2) a courseware is deployed on a LMS (class TEL_IsDeployedBy), and (3) a learning object may be part of another learning object or courseware (class TEL_IsPartOf).

Activities that can be processed on learning objects are depicted on Figure 3; we identified the indexation, consultation, download, rating and deletion operations. The class TEL_HasIntegrated translates the integration of a learning object within a courseware.

4.2 Some open source software

The implemented architecture is composed of the following components (see Figure 1):

– The learning systems to be tracked: the tool interacting with the ARIADNE LOR (called SILO on Figure 1) and MOODLE. These systems embed an agent responsible for collecting attention information when an activity is operated by a user.

– The tracking framework integrates OpenPegasus [OPE02], a C++ open source implementation of the WBEM standards; it offers both a tracking repository and its associated manager. The repository thus contains classes of the generic and specific models.

– The services of the middleware layer are developed using gSOAP [GSP02] tools for development of SOAP/XML Web services.

The management application (see bottom of Figure 1) offers two main functionalities: a unified view of traces stored into the OpenPegasus repository, and an interface to allow users extending the model.

The next section details interactions between the above entities during the collection of attention information resulting from the consultation of a learning object by a user connected to a MOODLE server.

4.3 Use case: consultation of a learning object from Moodle

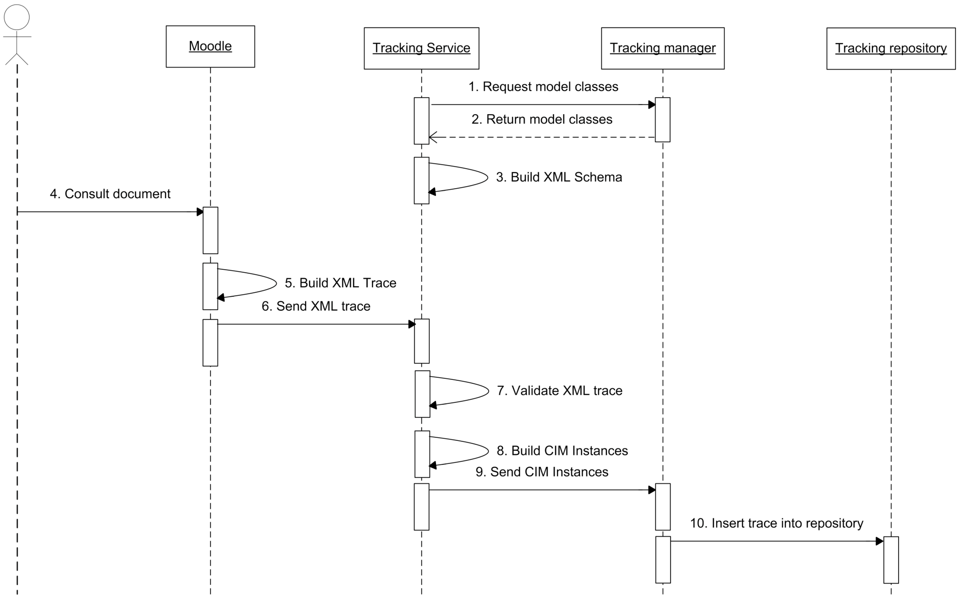

Figure 4: Collecting a trace

Figure 4 represents the UML sequence diagram illustrating exchanges and treatments required to ensure production and storage of a trace translating the consultation of a learning object by a student via MOODLE. When the tracking service is started, it queries the tracking manager (1) in order to retrieve classes of the tracking model (2), and finally creates the XML schema according to the models' specifications (3). When a learner consults a document on MOODLE (4), the integrated agent creates the matching XML trace (5) and transfers it to the tracking service (6); this trace contains information about the user, the learning object, the learning system and the activity. The tracking service then validates the XML trace against the XML schema produced in step 3 (7) and builds the matching instances of the models (8). These lasts are finally sent to the tracking manager (9) and stored into the tracking repository (10).

4 Conclusions and future works

In this paper we presented a generic approach to collect and share attention information produced by heterogeneous learning systems. Our framework is based on the WBEM standardized architecture which is natively implemented in most nowadays operating systems to facilitate local and remote management of hosted systems and applications. Therefore, the suggested modeling of traces can easily be injected in any WBEM component. In order to facilitate the process of managing specific attention data, we introduced a model management service that provides teachers and tutors with the opportunity of defining traces according to their pedagogical objectives. An implementation demonstrates how this approach can be exploited to store and share attention data related to learning objects manipulated within two different learning systems.

Since our modeling is compatible with applications hosted on a huge number of computers, it becomes easy to gather attention information resulting from users' activities. Indeed, the only task to achieve consists in embedding an agent into the supervised system in order to collect attention data; we plan to use aspect oriented programming approaches to automate this process [TA07] [LE07] and make management of attention information transparent. Moreover the WBEM architecture suggests a manager-to-manager communication, so that data stored into distributed tracking repositories can be exchanged without additional efforts.

Attention data collected by our works are until now poorly exploited; their exploitation is restricted to a graphical user interface that offers the possibility to users to visualize the usage of learning resources within various systems. The potential of these traces is much bigger; they can be used to personalize learners' access to distributed heterogeneous knowledge repositories [DO08], to generate graphical representations of traces that are easily visualized by instructors [MA05], to create Intelligent Tutoring Systems [LA07] which guide the students during the learning process depending on their knowledge, or to help tutors to improve the learning scenarios.

References

[ARN96]The Ariadne Foundation, 1996, available at http://www.ariadne-eu.org

[AT04] Attention.XML. Attention.XML specifications, 2004, available at http://tantek.com/presentations/2005/01/attentionxml.html

[BR06] Broisin, J.; Vidal, P.; Sibilla, M.: A Management Framework For Tracking User Activities In A Web-Based Learning Environment Based On A Model Driven Approach. In E. Pearson & P. Bohman (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2006 (pp. 896-903). Chesapeake, VA: AACE.

[CIM98] Systems Management: Common Information Model (1998). Open Group Technical Standard C804, DMTF, ISBN: 1-85912-255-8, 1998.

[CQL04] DMTF CIM Query Language Specification, 2004, available at http://www.dmtf.org/standards/documents/WBEM/DSP0202.pdf

[DMTF99]Distributed Management Task Force (1999). Common Information Model (CIM) Specification. Technical report, DSP0004, 1999, 96 p, available at http://www.dmtf.org/standards/documents/CIM/DSP0004.pdf

[DO08] Dolog, P.; Simon, B.; Nejdl, W.; Klobucar, T,: Personalizing Access to Learning Networks. ACM Transactions on Internet Technologies, Vol. 8, No. 2, Article 8, February 2008.

[GSP02] Engelen, R. A.; Gallivan, K.: The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks, in the proceedings of the 2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002), pages 128-135, May 21-24, 2002, Berlin, Germany

[IMS01] IMS LIP, 2001, IMS Learner Information Package Best Practice & Implementation Guide. Available at http://www.imsglobal.org/profiles/lipbest01.html

[KE07] Kepka, L., Heraud, J.M., France, L., Marty, J.C., Carron, T. Activity Visualization and Regulation in a Virtual Classroom. IASTED 2007.

[LA07] Lanzilotti, R.; Roselli, T,: An Experimental Evaluation of Logiocando, An Intelligent Tutoring Hypermedia System. International Journal of Artificial Intelligence in Education 17 (2007) 41-56.

[LE07] Leclet, D.; Lepretre, E.; Peter, Y.; Quenu-Joiron, C.; Talon, B.; Vantroys, T.: Améliorer un dispositif pédagogique par l'intégration de nouveaux canaux de communication. Environnements Informatique pour l'Apprentissage Humain, Lausanne, Suisse, p. 347-358.

[MA07a] May, M.; George, S.; Prévôt P.: Communication in Web-Based Learning Environments: for better collecting and exploiting learners' tracking data. International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2007).

[MA07b] May, M.; George, S.; Prevot, P.: Tracking, analyzing, and visualizing learners' activities on discussion forums. Proceedings of the sixth conference on IASTED International Conference Web-Based Education - Volume 2,Chamonix, France, Pages: 649 - 656, 2007

[MA05] Mazza, R.; Dimitrova, V.: Generation of graphical representations of student tracking data in course management systems. Proceedings of the Ninth International Conference on Information Visualisation (IV'05), p.253-258, July 06-08, 2005.

[MDL02]Moodle, 2002, available at http://moodle.org/

[NA05] Najjar, J.; Meire, M.; Duval, E.: Attention Metadata Management: Tracking the use of Learning Objects through Attention.XML. In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications. (2005). 1157--1161.

[NA06] Najjar, J.; Wolpers, M.; Duval, E.: Attention Metadata: Collection and Management, WWW2006 workshop on Logging Traces of Web Activity, Edinburgh, Scotland, (2006).

[OPE02] Open Pegasus, 2002, available at http://www.openpegasus.org/

[OT05] Otair, M. A.: Expert personalized e-learning recommender system. The first International Conference on E-Business and E-Learning (EBEL 2005), Princess Sumaya University for Technology (PSUT), Amman, Jordan

[OWBM01]Open WBEM, 2001, available at http://openwbem.sourceforge.net/

[RA09] Ramandalahy, T.; Vidal, P.; Broisin, J.: Opening Learner Profiles across Heterogeneous Applications. IEEE International Conference on Advanced Learning Technologies (ICALT 2009), Riga, Latonia, 14/07/2009-18/07/2009.

[TA07] Tarby, J. C. ; Ezzedine H. ; Rouillard J. ; Tran C. D. ; Laporte P. ; Kolsky C.: Traces Using Aspect Oriented Programming and Interactive Agent-Based Architecture for Early Usability Evaluation: Basic Principles and Comparison. Human Computer Interaction, Part I, HCII 2007, LNCS 4550, Beijing, Chine p. 632-641.

[WBEM99]Web Based Enterprise Management, 1999, available at http://www.dmtf.org/standards/wbem/

[WMI00]Windows Management Instrumentation, 2000, available at http://msdn.microsoft.com/en- us/library/aa394582(VS.85).aspx

[ZA02] Zaine, O. R.: Building a Recommender Agent for e-Learning Systems. Proc. of Int. Conf. on Computers in Education (ICCE’02).

[ZH07] Zhang, H.; Almeroth, K.; Knight, A.; Bulger, M.; Mayer, R.: Moodog: Tracking students' Online Learning Activities. Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007 (pp. 4415-4422). Chesapeake, VA: AACE.