AMASE: A framework for composing adaptive and Personalised Learning Activities on the Web

Athanasios Staikopoulos, Ian O'Keeffe, Rachael Rafter, Eddie Walsh, Bilal Yousuf, Owen Conlan and Vincent Wade

Knowledge and Data Engineering Group, School of Computer Science and Statistics, Trinity College Dublin, Ireland
{Athanasios.Staikopoulos, Ian.OKeeffe, Rachael.Rafter, Eddie.Walsh, yousufbi, Owen.Conlan, Vincent.Wade}@scss.tcd.ie

Abstract. Personalised Web information systems have in recent years been evolving to provide richer and more tailored experiences for users than ever before. In order to provide even more interactive experiences as well as to address new opportunities, the next generation of Personalised Web information systems needs to be capable of dynamically personalising not just web media but web services as well. In particular, eLearning provides an example of an application domain where learning activities and personalisation are of significant importance for engaging and enhancing the learning experience of a learner. This paper presents a novel approach and technical framework called AMASE to support the dynamic generation and enactment of *Personalised Learning Activities*, which uniquely entails the personalisation of media content and the personalisation of services in a unified manner. In doing so, AMASE combines state of the art techniques from both adaptive web and adaptive workflow systems.

Keywords: Adaptive Framework, Personalised Learning, Learning Activities, Adaptive Services and Workflow

1 Introduction

The Internet is increasingly being seen as a replacement for the desktop environment providing an integrated platform in which users interact with rich media content and services to carry out complex tasks. In order to further enhance the user's experience of the web, personalisation techniques need to be applied to web content, services and the workflow coordination of those services and content. Personalisation aims to ensure that media content and services are selected and tailored according to the user's personal needs, preferences, goals and context [1].

In order to enhance and improve the interactivity of the user experience, the next generation of Personalised Web information systems needs to be capable of dynamically personalising web media, services and workflow in a unified manner [2]. Typically most existing approaches to personalisation on the web have focused on tailoring only multimedia content, which is restricted to adaptive content, selection and navigation but have as yet omitted to consider adaptive workflow and adaptive services. In the AMAS project[1] we aim to develop innovative techniques and technologies to address this challenge and to support the dynamic and integrated personalisation of web media and services in the domain of eLearning.

eLearning is an inherently web-based domain where personalisation is of major importance. In eLearning, learning activities have been widely accepted as a means of providing greater engagement and enhancement of the learning experience [3]. Learning activities can be considered as specialised workflows, coordinating learning/educational content and tasks. In this case typical participants of the workflow are the learner and the educator. Many research projects such as LADiE [4] and successful learning activity environments such as LAMS [3] have investigated the pedagogic benefits of learning activities, such as for example, a peer review[2]. Personalised Learning Activities provide all the opportunities of learning activities but with the significant advantages of content, services (tools) and workflow being dynamically adapted to benefit individual learners. This customisation can be based on different "dimensions" of the learning occurrence such as the learner's preferences, prior knowledge, competences and context [5]. In AMAS we define the notion of a Personalised Learning Activity as a learning experience that involves the integration and personalisation of the selection, sequencing and presentation of both content and services.

This paper presents AMASE, a core technical framework of the AMAS project, which implements our approach for the dynamic generation and enactment of Personalised Learning Activities on the Web. AMASE combines the complimentary power of state-of-the-art techniques from the domains of both adaptive web and adaptive workflow systems. The actual adaptation process is a hybrid approach, utilising the capabilities of abstracted workflow and rule-based systems. In order to evaluate our approach and technical framework we have implemented an authentic case study involving undergraduate students of an SQL course.

The remainder of the paper is structured as follows: Section 2 presents the AMASE approach and framework. Section 3 describes the case study we have used to evaluate our approach and framework. Section 4 outlines the initial results of our evaluation. Section 5 presents a state-of-the-art review of related work. Finally, Section 6 summarises the main contributions of our approach.

2 AMASE Technical Framework

The AMASE framework supports the generation of Personalised Learning Activities, combining media content and (user centric) services in a unified manner. The framework is based on a number of interconnected components that generate, execute and support the user's interaction with a personalised activity. Fig. 1 portrays the AMASE architecture.

---

1 AMAS SFI project, please refer to http://kdeg.cs.tcd.ie/amas

2 In a peer review activity the learner reviews the work of one or more of his/her peers.

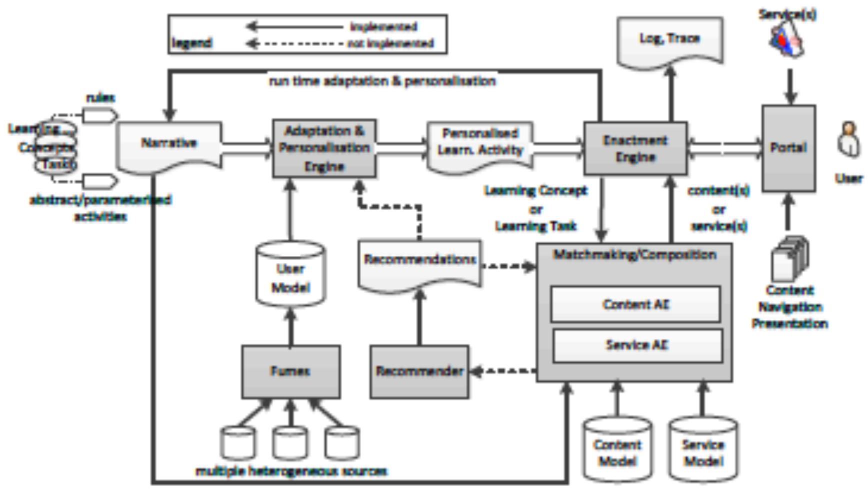

Fig. 1. AMASE Architecture Overview

Consider an educator (learning designer) who is designing a new course about Databases and SQL. Initially, he/she uses an authoring tool such as GRAPPLE [6] to conceptually design the course (Fig. 1 does not show the authoring phase of AMASE, rather the end product of that authoring). The design includes three key elements: the concepts and tasks from the learning domain (e.g. SQL commands and triggers), one or more abstracted or parameterised learning activities based on these concepts/tasks (e.g. SQL activity), and finally a set of adaptation rules that govern how the concepts and tasks should be adapted in different contexts (e.g. for novel/expert learners).

Three different types of adaptation rules are supported. Rules implemented with a concrete rule language such as Drools. These rules are stored within a repository, in this case the Drools Guvnor. Graph rules provide a more general and flexible mechanism in which learning designers can specify their own complex adaptation rules graphically. They are defined by a *when* clause (defining the triggering condition), a *before_pattern* (specifying the pattern to find in an activity graph) and an *after_pattern* (specifying the replace pattern). These rules are implemented similarly to the Graph Transformations rules defining the *pre* and *post* conditions on graphs [7]. Finally, there are Relationship rules that specify relationships among *Concepts* and learning *Activities* and which trigger adaptations. Based on these rules we have specified a number of predefined relationships that are available to a learning designer and which modify and compose an activity. For example, a *replaceWith* rule replaces one concept or activity with another and a *hasPrerequisite* rule applies a learning concept/activity before another.

Next, a Narrative contains the rules that are to be evaluated and used by the Adaptation and Personalisation Engine in order to create an executable Personalised Learning Activity.

A User Model stores the preferences, competencies, learning objectives and goals for each learner. User Models themselves are stored in a repository, such as an eXist database. In order to extract and combine information about a user from different sources such as their personalised learning environment (e.g. Sakai, Moodle) or social media, we use FUMES [8], a Federated User Model Exchange Service that implements a mapping-based approach to handle heterogeneity across different user models.

In AMASE we endorse a hybrid adaptation approach, combining the advantages and capabilities of workflow and rule-based systems. Rules are used to specify and evaluate the adaptation conditions as well as to trigger adaptations, while workflows are used to support the composition and coordinated execution of learning tasks. The adaptation process is depicted in Fig. 2. As inputs the Adaptation and Personalisation Engine receives: (i) an optional abstracted learning activity defining partial and parameterised workflows, (ii) a set of adaptation definitions, (iii) adaptation instances to generate in the first case rules to be evaluated and in the second case facts that will be inserted in the engine, (iv) a User Model to parameterise the adaptation with the preferences, competencies and objectives of a learner, and (v) a domain model defining the learning concepts and relations. In a case where an abstracted learning activity is not provided, the Personalised Learning Activity is constructed from scratch, exclusively by the adaptation rules. Finally, if a User Model is not provided then the adaptation process continues and generates a learning activity that applies to all users.

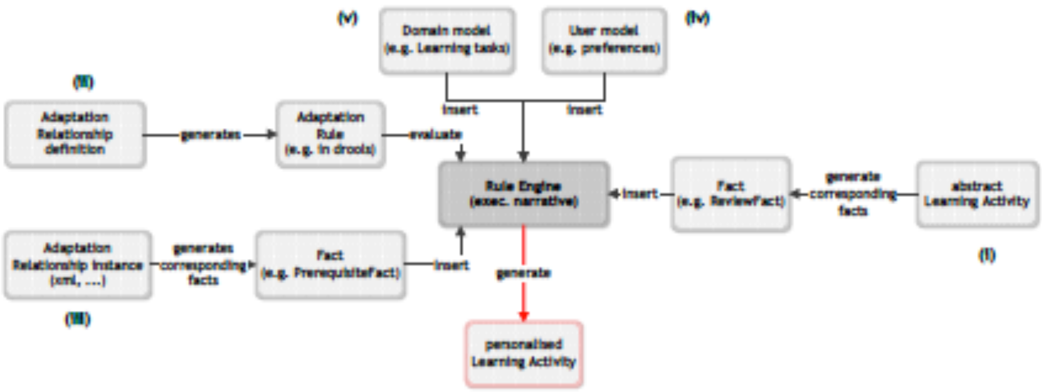

Fig. 2. Adaptation Process

As a result of the adaptation process, a Personalised Learning Activity is generated as a Business Process Model and Notation (BPMN) workflow specification, ready for execution. At this stage the activity has been personalised but remains abstract, as the appropriate content and services have not yet been selected in order to instantiate the tasks. The next step is to deploy the personalised activity to the Enactment Engine, so that it can be executed and made available to the learner. The Enactment Engine is a jBPM based workflow engine that supports the concurrent execution of multiple learning activities (BPMN workflows) assigned to individual as well as collaborating users. Learning activities are also stored in a repository so they can be reused and further customised for different domains and contexts. The current state of an executing learning activity is persistent and stored in a database. This enables us to support long-lived activities as they are dynamically loaded from a database. As the Enactment Engine provides the execution environment for learning activities, it interacts with the Narrative, to get adaptation rules that are to be evaluated at run time and accordingly trigger the dynamic adaptation of executing learning activities. The monitoring/logging mechanism monitors the instantiation and initiation of activities.

A learner interacts with the personalised activity through the web-based learning Portal (see Fig. 1). The Portal provides the learner with an environment in which both the content and services that make up the activity are available in an integrated manner. As the learner interacts with the Portal, requests are sent to the Enactment Engine in order to retrieve the appropriate content and services for the learner. At this point requests are passed to the matchmaking/composition service, which selects and composes if necessary, the appropriate content and services on a "just in time" basis. The service provides an interface for querying and composing available resources (content and services) based on the metadata descriptions that are stored in the corresponding content and service repositories. If the result of a query returns a set of results, the service will either compose content and services (based on a template) or select the most appropriate one (based on specific selection criteria). The decision to compose or select a service is based on the operating semantics of the service. As part of the matchmaking/composition mechanism, the Narrative is also used to provide rules that can influence the personalised selection and sequencing of both content and services. Similarly, we plan to use a Recommender system that will allow more serendipitous suggestions of content and services (or tools) based on user behaviour.

3 Case Study: A Personalised SQL Course

In order to evaluate our approach and technical framework we have implemented an authentic case study, where undergraduate students access an "SQL Database" course for a period of three months and from which they can interact with the Personalised Learning Activity that has been generated for them. We describe how the Personalised Learning Activity is generated for two different types of student in Section 3.1, and how they interact with it in Section 3.2.

3.1 Personalisation of the Learning Activity

In general, as part of the course users have to initially practice their SQL skills with a database sandbox environment and then to perform a Web Quest in which they must find and bookmark relevant material from the open web. This is followed by three parallel tasks: that of getting an assignment, designing a database with a design tool and implementing a database. Once users have completed all three activities they continue with the submission of their project. While users perform these tasks, they can participate in an online discussion forum and study specific assigned SQL topics.

In order to illustrate the personalisation of the course we consider two different learners: an expert (user01) and a novice (user02) who have individual learning preferences and prior knowledge. Fig. 3 provides screen captures of the Personalised SQL Course that is produced for user01 and user02. Due to their different competencies the adaptation process will assign a Practice Database for user02 but not for user01. In particular for user02 the matchmaker will associate that task with an Oracle database tool, due to his/her SQL preference for Oracle (over MySql). Next, for user02 the general Web Quest task will be replaced by a Questionnaire task, due to his/her learning preference to learn through examples (Constructivism). For user01 the matchmaker will resolve the Web Quest task to a combination of two services, the Search and Bookmarking. Both learners are assigned the Suggested Reading task. For user01 the Study SQL task is parameterised with a few SQL topics that are related to advanced and expert users, whereas for user02 the task is parameterised with more topics that are related to novice users. Next user01, due to his/her prior knowledge of information retrieval, will receive an assignment to implement a meta-search engine, whereas user02 due to his/her background in e-commerce, will have to build an airways reservation system. Both users are provided with the design database, implement database, and forum tasks. Next assuming that for user02, the submission period has expired, the submit task is changed into a late submission task. Finally, the review task will only be assigned to user01.

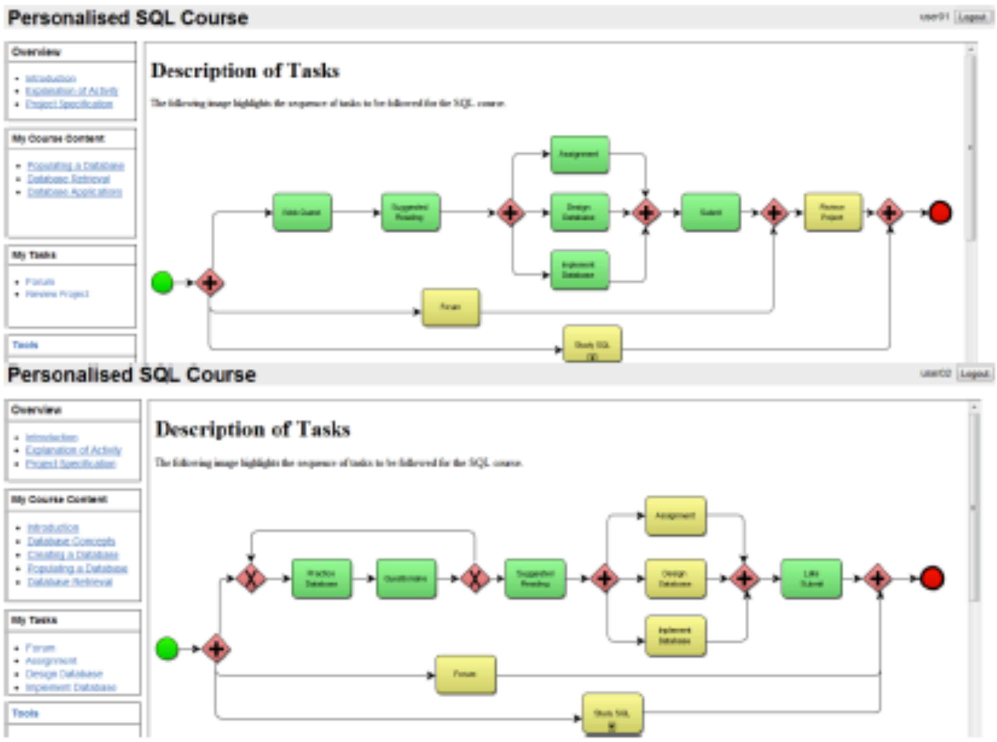

Fig. 3. Case Study for user01 (above) and user02 (below)

3.2 Personalised Learning Portal

Learners interact with the Personalised Learning Activity via a web Portal. The Portal interface (see Fig. 3) is divided into two main areas, the navigation menu on the left hand side of the screen providing access to general information about the course, the tools to be used, the assigned content and tasks, and the content panel on the right hand side. As shown in Fig. 4-left, once a topic is selected the content panel displays the content with the appropriate navigation options (submenu, buttons). Similarly, once a task is selected the content panel displays the appropriate services with which users can interact (see Fig. 4-right).

In order to realise the different tasks of the Personalised Learning Activity, the services need to be registered with the system. The metadata for the services are stored in a service repository describing the services/tools in terms of a service location (URI), type (e.g., SOAP, REST, Portlet), key-value pairs characterising the service, inputs, outputs, pre-conditions and post-conditions. Both services and tools offered by external and internal providers can be registered. Similarly, metadata about the attributes and characteristics of media content are stored in a content repository. In this case, most of the services are developed as Java portlets, which are deployed in a Liferay Portal server.

In this case study the services used are: 1) a Practice SQL Sandbox Service allowing students to try different SQL commands. 2) a Forum Service allowing for inter-student and tutor-student discussions, 3) a Search Service allowing students to perform web searches based on Microsoft Bing 4) a Bookmarking Service allowing students to keep track of their bookmarked links and submit them as part of the activity 5) a Submission Service allowing students to upload their project reports 6) a Late Submission Service replacing the submission service after a deadline is reached and penalising students for late submissions 7) a Review Service providing students with access to a set of reports that have been randomly allocated to them for review 8) a Recommender Service suggesting selected further reading based on the relevant resources that each student bookmarked 9) a Notification Service notifying students and educators with an email about their allocated tasks and 10) a Questionnaire service that randomly selects SQL questions to test the command skills of some learners.

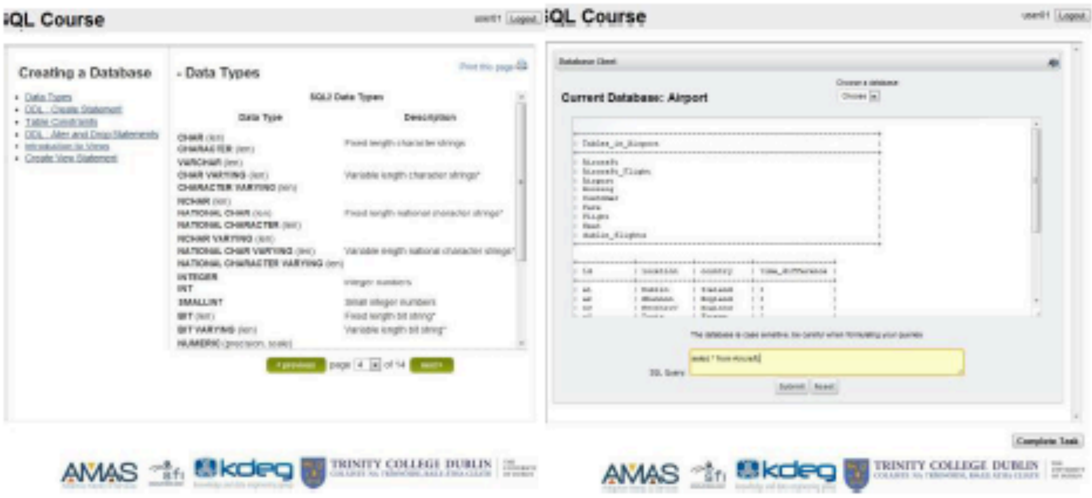

Fig. 4. Personalised Learning Portal for content (left) and services (right)

4 Evaluation

We have evaluated our approach and technical framework through the case study described in Section 3 during the 2011/12 academic year. The course involved ninety undergraduate students from 3rd and 4th year. In particular, we examined our system under three criteria: the students' perception of the usability of the system, the effectiveness of the system and its performance. Students' perception was evaluated via a questionnaire and elicited learner opinions about the quality of the experience in using the learning activities and web Portal. Overall the students' perception of the system was positive. For example 90% of students felt that the course (Personalised Learning Activity) generated for them reflected the answers they gave via a self-evaluation instrument about their database skills and knowledge. 84% of students felt the course was easy to navigate. 71% of students said the course generated did not appear disjoint, and 83% of students found the flow between the tasks appropriate. In terms of performance, given that this is the first version of the system, the system performed within acceptable time boundaries. For example, the time for building a Personalised Learning Activity was between [2942, 3335] milliseconds, the time each piece of content/task took to build and load was minimal, while the system could easily escalate to many concurrent users (e.g., 100) with no problem. Also upon analysis of the system log files we found that 93% of students completed all of the tasks assigned. In addition, we find that on average each piece of content (273 in total) provided was visited 44 times by each student over the 3 month period the course was given. As a result, we would argue that both these results support the fact that students found the course both usable and engaging. In the future we intend to further evaluate our system in terms of effectiveness. This will be based on a comparative analysis of the questionnaires and assignments that the students completed this year, and in previous years when Personalised Learning Activities were not used. For brevities sake we have not included the full experimental results but these are available online from our website[3].

5 Related Work & Discussion

Current integrated learning environments such as Blackboard, Sakai and Moodle provide the delivery of content and services. However, such environments do not go far enough in addressing the particular needs of a learner via personalisation and suffer from the "one size fits all" problem. Furthermore, despite their support for services, they do not provide any means by which a learning designer can control the sequencing of the services included in their activity. Another common limitation of such learning environments is their closed nature [9], limiting educators to only use services provided by the system. On the other hand, specialised adaptive hypermedia systems such as GALE [10] and ADAPT2 [11] handle content adaptivity but fail to address the requirement for services, (ADAPT2 provides limited support for services, treating them as a special type of content allowing very limited capabilities with respect to the type of control flow that can be used to sequence the services). The IMS Learning Design (LD) specification [12] can be used to describe pedagogically driven learning activities using a platform independent language. However, the specification itself provides only basic support for adaptivity and use of services, supporting only three types of services, namely: an email, a discussion and a search service. Additional services can be added, however the actual implementation of these services is left up to the platform. Where LD-based systems have been extended to support the "adaptive selection" of external services, as is the case with the GSI [13] and the

---

[3] http://kdeg.cs.tcd.ie/amas/initial_evaluation.pdf

Gridcole [14], their support is limited to the instantiation of abstract service definitions and manual selections, not taking into account the learner's needs.

From the perspective of adaptive services and workflow, eFlow [15] provides an approach for the dynamic composition, enactment and management of composite services. ADEPT [16] also supports the modification of processes during execution, both at definition and instance level. YAWL supports the dynamic selection of worklets [17] at runtime based on a set of rules that are written by the workflow designer. AgentWork [18] provides the ability to modify process instances by dropping or adding individual tasks based on events and rules. In addition, CAWE [19] is an adaptive workflow system that supports adaptation based on the individual user, the contextual properties and the device they are using. Finally, C-BPEL [20] supports the adaptive selection of services to instantiate the activities in a workflow at runtime. However, most of the workflow approaches outlined do not perform adaptations upon an abstracted and standard workflow language as BPMN, but rather upon concrete implementations that are tied to specific technologies such as for YAWL and WS-BPEL. That means educational designers need to be experts in these languages in order to design a complete and executable learning activity. Services and exchanged data are also hard bound to the workflow, therefore not allowing the dynamic resolution of tasks to services based on their descriptions. There are also even less approaches allowing the dynamic adaptation of workflow instances based on the just in time evaluation of rules. In addition most of these systems they do not consider adaptation from a personalisation and customisation perspective. As a result a user model is not captured and it does not play a significant role in the adaptation process. Finally, there are even less approaches considering the domain specific characteristics and design principles of learning activities.

6 Conclusions

In this paper we have presented AMASE, a novel approach and technical framework providing a personalised and interactive learning experience for learners. AMASE supports the dynamic generation and enactment of Personalised Learning Activities, consisting of both media content and (user centric) services in a unified manner that are personalised to suit different learners.

AMASE is based on a number of distinctive characteristics and provides several advantages over other state of the art offerings. In particular it has the ability to combine both media and services, which can be personalised both in their enactment and in their sequencing via a workflow. This combination of all of these techniques together in a unified system is something that has not been done before, and which we feel is particularly suitable for advancing both the interactivity and suitability of web-based systems, in particular web-based learning systems.

7 Acknowledgement

This work was funded by Science Foundation Ireland via grant 08/IN.1/I2103.

8 References

1. Brusilovsky, P.: Adaptive navigation support. The adaptive web. pp. 263–290. (2007).
2. De Bra, P., Aroyo, L., Chepegin, V.: The next big thing: Adaptive web-based systems. Journal of Digital Information. 5, (2006).
3. Dalziel, J.: Implementing Learning Design: The Learning Activity Management System (LAMS). In: Crisp, G., Thiele, D., Scholten, I., Barker, S., and Baron, J. (eds.) Learning. pp. 593-596. (2003).
4. Jeffery, A., Conole, G., Falconer, I.: LADiE Project Final Report, http://misc.jisc.ac.uk/refmodels/LADIE/www.elframework.org/refmodels/ladie.html.
5. Wade, V.: Challenges for the Multi-dimensional Personalised Web, (2009).
6. Dagger, D., Harrigan, M., Kamari, G., Wade, V.: GRAPPLE and Graph Drawing. (2008).
7. Baresi, L., Heckel, R.: Tutorial introduction to graph transformation: A software engineering perspective. Graph Transformation. 2505, 402-429 (2002).
8. Walsh, E., O'Connor, A., Wade, V.: Supporting Learner Model Exchange in Educational Web Systems. 7th International Conference on Web Information Systems and Technologies (2011).
9. Dagger, D., O'Connor, A., Lawless, S., Walsh, E., Wade, V.P.: Service-oriented e-learning platforms: From monolithic systems to flexible services. Internet Computing, IEEE. 11, 28–35 (2007).
10. Smits, D., De Bra, P.: GALE: a highly extensible adaptive hypermedia engine. Hypertext 2011. pp. 63-72 (2011).
11. Brusilovsky, P., Sosnovsky, S., Yudelson, M.V.: Ontology-based Framework for User Model Interoperability in Distributed Learning Environments. Exchange Organizational Behavior Teaching Journal. 2851-2855 (2005).
12. Consortium, I.M.S.G.L.: IMS Learning Design Specification, (2003).
13. De la Fuente Valentín, L., Pardo, A., Kloos, C.D.: Generic Service Integration in Adaptive Learning Experiences using IMS Learning Design. Computers & Education. 57, 1160-1170 (2011).
14. Bote Lorenzo, M.L., Gómez Sánchez, E., Vega Gorgojo, G., Dimitriadis, Y.A., Asensio Pérez, J.I., Jorrín Abellán, I.M.: Gridcole: A tailorable grid service based system that supports scripted collaborative learning. Computers & Education. 51, 155-172 (2008).
15. Casati, F., Ilnicki, S., Jin, L.J., Krishnamoorthy, V., Shan, M.C.: Adaptive and dynamic service composition in eFlow. Advanced Information Systems Engineering. pp. 13–31. Springer (2000).
16. Dadam, P., Reichert, M.: The ADEPT project: a decade of research and development for robust and flexible process support. Computer Science Research and Development. 23, 81-97 (2009).
17. Adams, M., ter Hofstede, A., Edmond, D., van der Aalst, W.: Worklets: A service-oriented implementation of dynamic flexibility in workflows. On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE. 291–308 (2006).
18. Robert, M., Ulrike, G., Erhard, R.: AgentWork: A Workflow System Supporting Rule-based Workflow Adaptation. Data & Knowledge Engineering. 51, 223–256 (2004).
19. Denko, M., Ardissono, L., Furnari, R., Goy, A., Petrone, G., Segnan, M.: The Context Aware Workflow Execution Framework. Citeseer. 1-21 (2007).
20. Ghedira, C., Mezni, H.: Through Personalized Web Service Composition Specification: From BPEL to C-BPEL. Electronic Notes in Theoretical Computer Science. 146, 117-132 (2006).