Non-Invasive Adaptation Service for Web-based Content Management Systems

Kevin Koidl, Owen Conlan, Vincent Wade

Centre for Next Generation Localisation and Knowledge and Data Engineering Group, Trinity College Dublin, Ireland

{Kevin.Koidl, Owen.Conlan, Vincent.Wade}@cs.tcd.ie

Abstract. Most Adaptive Hypermedia Systems today focus on providing an adaptive portal or presentation/application through which adaptive retrieved content is delivered. Moreover the content used by such adaptive systems tends to be handcrafted for adaptivity, i.e. from closed corpus repositories. If adaptive hypermedia is to become more mainstream we need the ability to embed adaptive information retrieval and composition techniques within existing Web-based Content Management Systems (WCMS) e.g. Wiki, Drupal. However the effort and expense in fundamentally re-engineering such WCMS or developing completely new adaptive WCMS, is very high. This research explores the architectural and technical issues involved in providing a third party adaptive service which is more easily plugged into existing WCMS. The advantage of this approach is that it doesn't break the current browsing paradigm of freely navigating across different independent web sites on the web. In order to address this challenge we introduce a third party Adaptive Service that enables a unified cross-website personalized experience by discretely interfacing with separate independent WCMS. Thus the Adaptive Service strives to personalize the users experience when accessing each WCMS in a non-invasive manner by not adversely interfering with the web sites overall look and feel, content and functionality without losing the web sites identity and fundamental experience. This paper describes the design and an initial use case supported implementation example of this Adaptive Service. It also provides a survey that assesses the feasibility of the services integration with leading WCMS.

Keywords: Open Corpus, Adaptive Hypermedia, Web-based Content Management Systems (WCMS), Adaptive Hypermedia Strategies

1 Introduction

Although Adaptive Hypermedia Systems (AHS) have matured, they may still be seen as a niche application mostly with a manually managed closed corpus that is adapted to a well known user group. Furthermore the adaptive logic driving the adaptive process needs to be closely related to the data models and the content. This close relationship mostly results in limitations in the re-usability and interoperability of the AHS, also known as the "Open Corpus Problem" in Adaptive Hypermedia [2].

Recently more flexible AHS architectures have been introduced addressing the problem of re-usability and interoperability. E.g. by providing a distributed server architecture [1] or flexible rule engines facilitating the adaptive logic [4]. Nevertheless most AHS still tend to provide a central AHS portal restricting the user's flexibility; thus isolating the adaptive experience. This paper introduces an approach using a flexible AHS architecture and providing flexibility to the user.

The goal of this approach is to provide open corpus adaptivity by personalizing the browsing experience over several individual content hosting systems known as Web-based Content Management Systems (WCMS). To achieve this we introduce a third-party Adaptive Service discretely interfacing with the individual WCMS. The main advantage of this approach is that the adaptivity is instrumented by the WCMS and not by a central AHS portal. By bringing the adaptivity to the WCMS, the user can maintain the current browsing paradigm by freely navigating on the web and, at the same time, benefiting of personalized content. For this the Adaptive Service provides the individual WCMS with non-intrusive adaptive recommendations reflecting the overall intent/interest of the user. The unified and personalized browsing experience across different WCMS is defined as *adaptive flow*.

In order to illustrate the functionality of the introduced third-party Adaptive Service a use case is provided in section 4. This use case describes the individual steps in which a user can be assisted on the web by the introduced third-party Adaptive Service. Based on the use case one particular implementation example is also given. This example includes a query interception and augmentation by the third-party Adaptive Service connected to a WCMS. The implementation example is followed by a description of the overall architecture.

Motivating the use case, a survey of two prominent WCMS Drupal and Wikipedia's implementation platform MediaWiki is provided. The survey focuses on the suitability of WCMS for adaptivity. Furthermore a state of the art in discussing recent developments in flexible service driven AHS is also provided.

2 Adaptive Hypermedia and third-party services integration

In order to develop a third-party Adaptive Service, which integrates with different WCMS, three main challenges need to be addressed: (1) the identification and implementation of necessary adaptive features in WCMS, (2) the development of adaptive logic driving the personalization process across several independent WCMS and (3) providing a user model unifying the browsing experience across the different WCMS. The second and third challenge is addressed in the following state of the art by discussing current AHS and user modelling developments. However the first and most difficult challenge is addressed in the two following separate sections three and four.

For an AHS to allow more flexible adaptation across different WCMS the possibility of separating the adaptive logic from the content is essential. Recently several

research groups are driving their AHS towards more flexible AHS architectures. A subset of the numerous examples discussed in the literature is: KnowledgeTree [1], APeLS [4], MEDEA [13] and AHA! [6]. E.g. KnowledgeTree implements an AHS by providing a community of distributed servers to clearly separate concerns. This is done by distributing different functions to designated servers compared with one AHS bundling all functionalities together. For this KnowledgeTree provides four types of servers: learning portals, activity servers, value-added services, and student model servers [1]. Although the communications protocols between the different distributed servers are based on a simple HTTP GET request and not on Web Service protocols such as SOAP, KnowledgeTree provides a good example of distributed AHS integration relevant to this research.

APeLS, as another example, also provides a distributed architecture, although by using separate adaptive services not servers. Furthermore APeLS enables Web Service based communication between the different AHS services. The individual AHS services are (a) the adaptive hypermedia service providing the content and (b) the learning environment. The later is used to track the learner, based on the tutor's guidance in the form of learner profiles, assessment information and pedagogical constraints. The result integrates both services and is displayed in the client's browser. Furthermore APeLS allows the flexible design of adaptive logic based on pluggable rule engines managing different narratives driving the adaptivity [5]. In addition to the usage of Web Service communication the multi model approach of APeLS is relevant to this research. Nevertheless both KnowledgeTree and APeLS focus on closed corpus and not apply adaptation across multiple independent services in order to include open corpus in the adaptation process.

In the following the integration of user models within a third-party Adaptive Service is discussed. The main challenge is to provide a modelling approach that reflects a unified browsing experience over several different WCMS without the need to model separate user models for each WCMS. For this a distributed user modelling service unifying the different modelling parameters of the WCMS is necessary. For example Personis [8] provides such a distributed user model approach designed to provide a unified user model of several different systems. An additional distinctive feature of Personis is user model scrutiny, ensuring the user is involved in all user model related decisions. The architecture is based on a XML-RPC interface allowing third party service integration. A similar approach, centralizing the user model, is taken by CUMULATE [3]. Nevertheless Personis and CUMULATE lack flexibility due to the unified storage of user models in a centralized repository. A more flexible approach is followed by FUMES [15] providing a decentralized mapping framework to support the exchange of heterogeneous user models without the need of a central repository. In relation to the introduced third-party Adaptive Service FUMES provides a possibility of retrieving a single user model based on user information collected in several different WCMS.

3 WCMS survey for compatibility to support modular based Adaptive Service access

This section is based on a survey investigating the integration of third party adaptive services in Web-based Content Management Systems (WCMS). The following is only a brief extract of the entire survey introducing the most important conclusions for this research.

Taking a broad view on WCMS shows that this type of system refers to a whole range of different applications and functionalities with the main commonality of providing effective and accessible tools for publishing and organising web-based content. Goodwin and Vidgen define a WCMS as:

" [...] an organizational process, aided by software tools, for the management of heterogeneous content on the web, encompassing a life cycle that runs from creation to destruction" [14]

The term WCMS mostly refers to open source based Content Management System rather than to commercial systems which are referred to as Enterprise Content Management Systems (ECM). Prominent examples for WMCS are MediaWiki, Drupal and WordPress. Beside these systems numerous WCMS have appeared and have become an enormously popular application domain in day to day Internet activities.

The integration of adaptive functionalities in WCMS pose a new set of challenges towards the adaptive process with the following as the most significant:

- Any adaptive intervention has to maintain the entire look and feel i.e. branding of the WCMS.
- All internal policies regarding user rights, especially content-related rights, need to be obeyed by any adaptive intervention.
- The WCMS has to provide API or Web Service interfaces to enable third party Adaptive Service intervention.
- The WCMS needs to be extendible/pluggable in order to handle adaptive interventions.
- Semantic web functionalities have to be available within the WCMS, e.g. via a module based extension to the core platform.

To assess and tackle these challenges, two specific WCMS were examined, MediaWiki [9] well known as the basic WCMS used for Wikipedia and Drupal [7] known as one of the most flexible WCMS based on its flexible pluggable module architecture. Both systems provide a wide range of possibilities and are supported by a large and active developer group.

The architecture of MediaWiki is simpler than Drupals, but it is not as flexible. This is due to the fact that Drupal is based purely on a module based pluggable architecture.

However MediaWikis core strength lies in the management of content and not in the extensibility of the platform. Nevertheless extensions are possible, e.g. the *extensions* feature is based on simple scripts for the adding of different plugin types like Adobe Flash, Video streaming, RSS feeds, ratings and API based third party accessibility. It also has to be noted that recently more enhanced extensions were introduced especially for the adding of semantic structures and relationships within the different MediaWiki implementation [10].

The WCMS Drupal on the other hand is based on a flexible module based pluggable framework also referred to as "Content Management Framework" [7]. Drupal does not specialise on one specific type of content, like MediaWiki which focuses on encyclopaedia based content or Wordpress concentrating on blog based content, but it provides an extendible core implementation that can plug different modules depending on the application area. This high level of flexibility and abstraction comes with the cost that developers need to have good knowledge of the core architecture in order to extend it. Like MediaWiki, Drupal also provides a module to enable semantic annotations [11].

However it is essential to explore the extensibility of WCMS for it to engage with a third-party Adaptive Service and to provide the ability to use the adaptive interventions send by a third-party Adaptive Service. Currently both discussed WCMS provide the possibility to communicate with a third-party Adaptive Service. Nevertheless both need to be extended to cater for effective adaptivity from within. Currently it is not possible to take any WCMS deployment and apply adaptivity simply by using API function calls.

Fortunately the engineering of a WCMS is not a difficult task and especially in the case of MediaWiki and Drupal extensibility is possible without major changes to the WCMS architecture. E.g. MediaWiki API provides a powerful interface for fast and high level access to all data within the database. Functions include user login, content uploads and updates. In addition to the API MediaWiki provides specific extension points for the extension of its deployment. These extensions can be plugged into the core implementation at any time and do not trigger any re-deployment. The following extensions can be seen as relevant for enabling adaptivity in MediaWiki: "Hooks" to react to user actions, displaying "special pages" and "skins" allowing changes in the look and feel of MediaWiki.

Extending the main Drupal core on the other hand implies the extension with specific Drupal modules. These modules are simple to design, plug and unplug. From within the modules several functionalities to control the information flow from the database to the Front End are provided. Following events are useful for the "hooking" of adaptive interventions into Drupal: Content/Node creation, deletion and viewing, as well as user login/logout and user account/profile updates. Both WCMS, MediaWiki and Drupal therefore provide a good base for further development towards more flexible and distributed adaptivity.

Besides applying extensions to the individual WCMS an alternative approach is possible. This approach is based on the fact that most interactions between user and WCMS are based on the usage of a browser. Furthermore current browser technologies allow the integration of extension/plug-in which can be used to manipulate, augment or redirect the data before being displayed to the user. In relation to this research an interesting example is the "Smarter Wikipedia" Firefox plug-in, which adds a "related articles" box to MediaWiki's Wikipedia implementation [12]. This kind of browser central development presents an alternative approach which avoids the current need to extend WCMS, but may lead to constant updates due to changes in the underlying WCMS. Nevertheless a purely browser focused implementation is currently not part of the approach introduced in this paper.

In the following a use case is illustrated discussing the usage of the third-party Adaptive Service on WCMS.

4 Use Case

This use case illustrates the usage of the third-party Adaptive service providing a unified adaptive experience over several independent WCMS. As mentioned above this unified experience is defined as *adaptive flow*. It's main purpose is to retain a unified personalized information space for the user. To achieve this each of the WCMS communicates with a third-party Adaptive Service. As the user navigates, the Adaptive Service gains knowledge about their browsing over time. Thus, the Adaptive Service can provide improved non-intrusive adaptive recommendations to the WCMS. Figure 1 illustrates a specific scenario.

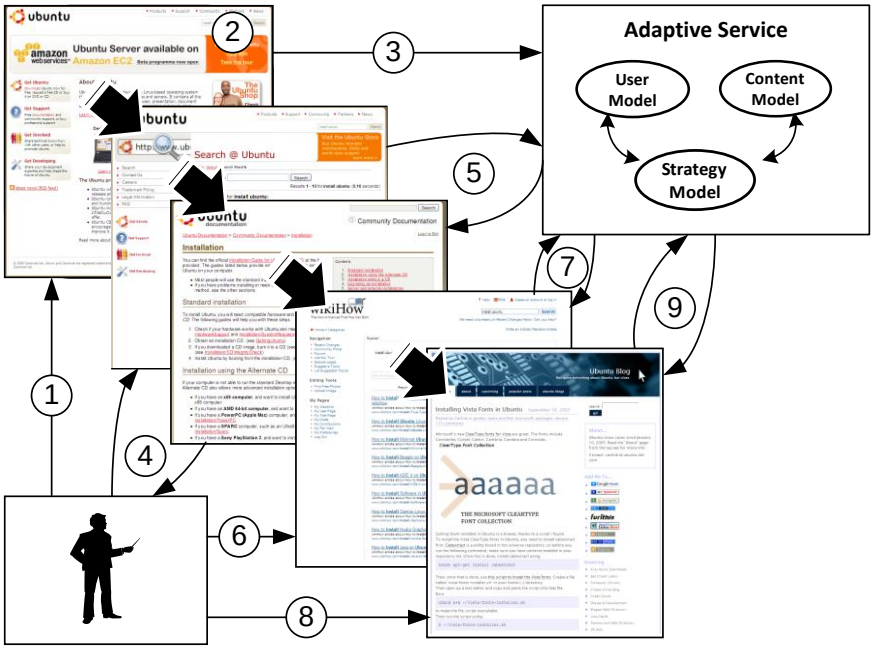

Figure 1 illustration of the overall adaptive approach indicating an *adaptive flow* over several different WCMS

1. User John is interested in installing the Debian GNU/Linux based Ubuntu operating system, but before he makes his final decision he wants to gather information about the installation process. For this he navigates to the Drupal based Ubuntu homepage. John's Adaptive Service is active and will provide a more personal browsing experience.
2. After navigating to the Ubuntu homepage John states the explicit part of his interest in the search field. He uses the term "install Ubuntu".
3. The search module of the Drupal based Ubuntu homepage informs the third-party Adaptive Service about John's query. The Adaptive Service registers the query and cannot find any previous interest related to this query. Therefore the Adaptive Service initialises a new "adaptation flow" session prompting the Adaptive Service to wait for more evidence coming from John's interaction with the Ubuntu homepage.
4. After stating the query John receives the result list from the Ubuntu homepage without any adaptive interventions. He starts clicking on different results from the original result list.
5. John's interaction with the result list is registered by the Adaptive Service. At this point the Adaptive Service remains in a non-adaptive state identifying John as being in an 'orientation' phase.
6. John believes he has enough high level information and leaves the Ubuntu homepage. Now he navigates to the MediaWiki implementation "wikiHow".

He wants to receive more in depth information about the installation process and hopes to find it at this point.

7. The Adaptive Service is informed by "wikiHow" and registers John's access to the MediaWiki based page. John uses this page frequently and knows that the Adaptive Service is interacting with the WikiMedia based WCMS. After the Adaptive Service receives the information that John is navigating to the new WCMS the Adaptive Service sends an extended query to the "wikiHow" page. John is now presented with a personalized result list based on his previous browsing experience on the Ubuntu Homepage, instead of seeing the main homepage in which John would have had to reissue his query.
8. After interacting with the personalized search result provided by the "wikiHow" webpage, John decides to navigate to the WordPress based blogging site "Ubuntu blog".
9. The Adaptive Service now maintains a well informed stream of experience from John's previous browsing pattern and is able to negotiate the most appropriate blog entries for John.
10. John believes he is well informed and decides to install Ubuntu.

The most essential part of the illustration in figure 1 is indicated with the diagonal arrows ranging over the different WCMS. It indicates the adaptive flow which can be seen as a development towards a more personalized browsing experience. Furthermore it is important to note that the illustrated open corpus adaptive scenario is controlled principally by the user. The Adaptive Service only provides adaptive recommendations which then can be used by the WCMS during their interaction. The use of the Adaptive Service allows different WCMS to share the latent aspects of the user's preferences and intent. These are typically lost as the user navigates between different WCMS on the web.

Compared with conventional distributed AHS the introduced third-party Adaptive Service follows a different approach in that it seeks to maintain attributes which apply across a variety of WCMS, i.e. by using content models representing semantic concepts.

To illustrate the functionality of the third-party Adaptive Service an implementation example is provided in the next section.

5 Implementation example

Taking the Drupal WCMS as an example this section defines the means by which the Adaptive Service can effect the recommendation of content to the user.

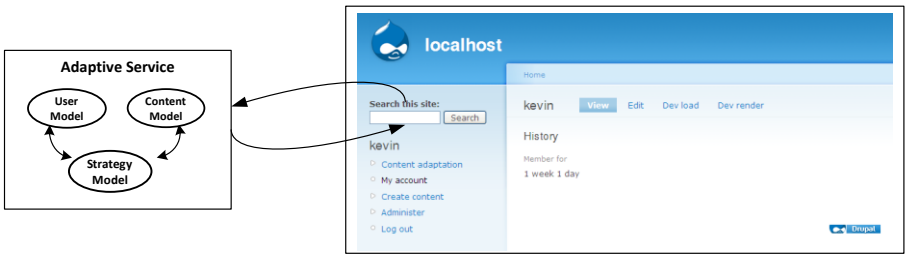

Figure 2 exemplifying a basic adaptive integration within the Drupal WCMS.

As indicated in figure 2 the basic Drupal search module was activated. Furthermore the Drupal *hook_search_preprocess* is used to intercept the user's query and send it to the Adaptive Service. In addition to the query interception, content related user activities are logged with Drupals *node_hook_api*. It registers content related activities and saves these in the underlying database. This information is then used to extend the user model of the Adaptive Service.

The Adaptive Service on the other hand has to handle the incoming information in order to send adaptive recommendations back to the WCMS. For this the Adaptive Service implements a JBoss Rules engine based on APeLS [4]. This engine allows the usage of flexible adaptive strategies manifested as individual rules. Based on the example illustrated in figure 2 the following steps are executed by the Adaptive Service:

- The Adaptive Service identifies and authenticates the user.
- The Adaptive Service waits for information to be sent from the Drupal based WCMS about the user's activities.
- After the user issues a query it is intercepted by Drupals hook_search_preprocess which sends it to the Adaptive Service.
- In addition the Adaptive Service can receive information about the user content related interactions from Drupals hook_node_api.
- Based on the available user and content model information a specific adaptive strategy is activated.
- The activated adaptive strategy orchestrates the information provided in the user and content model and uses it to augment the query.
- The Adaptive Engine sends the augmented query back to the WCMS.
- The WCMS executes the query and presents a personalized ranked list to the user.

This example illustrates a query interception and query augmentation scenario for Drupal. In this case the final result is a personalized ranked list. However this approach allows more complex adaptive scenarios like adaptive navigation support and adaptive presentation necessary for the overall research illustrated in figure 1.

6 Architecture

This section describes the overall architecture of the third-party Adaptive Service and its integration with different WCMS. The Adaptive Service is based on APeLS [4] and uses pluggable rule engines to facilitate the wide variety of WCMS architectures.

This architecture addresses three main challenges: (a) the sending of user and domain information as input for the overall personalization process from the WCMS to the Adaptive Service, (b) the processing of the send information by the Adaptive Service (c) the creation of appropriate adaptive recommendations by the Adaptive Service to personalize the output of the WCMS. Please note figure 3 illustrating the overall architecture.

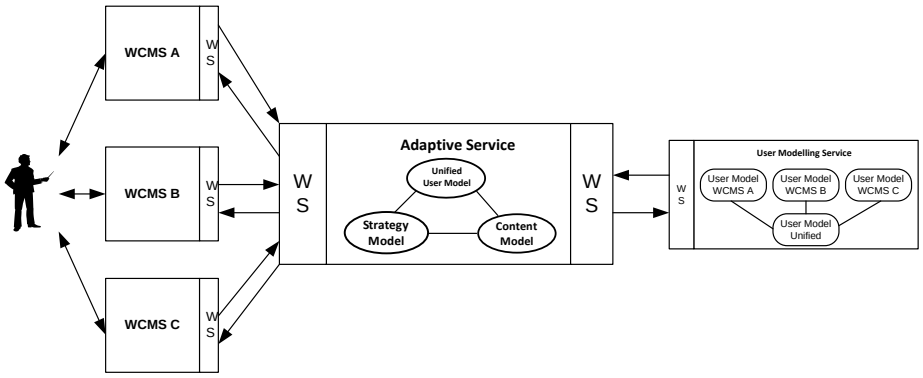

Figure 3 indicating the overall architecture of the third-party Adaptive Service

The input for adaptive personalization across independent WCMS is based on the user's browsing behaviour. This behaviour is collected by the WCMS and delivered to the Adaptive Service through a Web Service (WS) interface in the back-end of the WCMS. The user simply navigates within and between the different WCMS with a standard browser and is not explicitly aware of the back-end integration. This allows the user to remain in full control of the browsing experience.

The processing of all relevant information including the user and the domain model is handled by the Adaptive Service. For this the Adaptive Service communicates with an additional user modelling service e.g. FUMES [15] which provides a unified user model reflecting the current and previous browsing experience across separate WCMS. Together with a content model, which stores all information available about the structure and nature of the content within the WCMS, the Adaptive Service can compose an appropriate adaptive strategy reflecting the overall intent of the user.

Based on the appropriate adaptive strategy the Adaptive Service can send adaptive recommendations to the WCMS. The WCMS can use these recommendations to create a personalized output for the user.

The most significant feature of the introduced architecture is placing the Adaptive Service behind the WCMS allowing the user to retain the current browsing paradigm of free web navigation.

7 Conclusion and Future Work

This paper addresses the open corpus problem by introducing a third-party Adaptive Service providing adaptivity across different independent WCMS. The novelty of this approach is the complete separation of the Adaptive Service and the content host i.e. the WCMS. For this the Adaptive Service discretely connects with the WCMS providing non-intrusive adaptive recommendations. The user does not have to be aware of the adaptive interventions, thus this approach retains the current browsing paradigm of free navigation on the web. Furthermore the non-intrusive nature of the adaptive interventions maintains the branding, look and feel and operation of the individual websites modified.

To illustrate the usage of the third-party Adaptive Service a use case was provided and later specified by an implementation example proving third-party Adaptive Service integration with a Drupal based WCMS. However the example illustrates only one possible adaptive application within the overall architecture introduced. Further developments towards more sophisticated personalization include more flexible content and link adaptation. However the introduction of such advanced adaptive functionalities strongly depends on the architecture of the different WCMS. In order to gauge the feasibility of working with different WCMS a survey was discussed which indicates general key challenges for the integration of a third-party Adaptive Service in WCMS.

Future work will concentrate on three connected areas: (1) the integration of additional WCMS into the adaptive service framework, (2) the extension and empirical evaluation of resulting adaptive logic/strategies and (3) the further integration of a unified user modelling approach across different WCMS.

Acknowledgements. This research is supported by the Science Foundation Ireland (grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College, Dublin.

8 References

[1] Brusilovsky, P.: KnowledgeTree: A distributed architecture for adaptive e-learning. In: Proceedings of The Thirteenth International World Wide Web Conference, WWW 2004 (Alternate track papers and posters), ACM Press 104–113. (2004)

[2] Brusilovsky, P. & Henze, N.: “Open Corpus Adaptive Educational Hypermedia”. In The Adaptive Web: Methods and Strategies of Web Personalisation, Lecture Notes in Computer Science, vol. 4321, Berlin: Springer Verlag, pp. 671-696. Berlin (2007).

[3] Brusilovsky, P., Sosnovsky, S. A., & Shcherbinina, O.: User Modeling in a Distributed E-Learning Architecture. Paper presented at the 10th International Conference on User Modeling (UM 2005), Edinburgh, Scotland, UK, July 24-29, 2005.

[4] Conlan, O., Hockemeyer, C., Wade, V., Albert, D.: Metadata driven approaches to facilitate adaptivity in personalized eLearning systems. Journal of the Japanese Society for Information and Systems in Education 1(1) 38–45 (2002).

[5] Dagger, D., Conlan, O., and Wade, V. P.: An architecture for candidacy in adaptive eLearning systems to facilitate the reuse of learning Resources. In: Rossett, A. (ed.) Proc. of World Conference on E-Learning, E-Learn 2003, Phoenix, AZ, USA, AACE 49-56 (2003).

[6] De Bra, P., Smits, D., Stash, N., The Design of AHA!, Proceedings of the ACM Hypertext Conference, Odense, Denmark, August 23-25, 2006 pp. 133, and http://aha.win.tue.nl/ahadesign/, (2006).

[7] Drupal Community Page, http://drupal.org/getting-started/before/overview. (Last access 25.02.2009)

[8] Kay, J., Kummerfeld, B., and Lauder, P. Personis: A server for user modeling. In: De Bra, P., Brusilovsky, P. And Conejo, R. (eds.) Proc. of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems AH'2002, Málaga, Spain, pp 201-212 (2002).

[9] MediaWiki Foundation, http://www.mediawiki.org/wiki/MediaWiki (Last accessed 16/03/2009)

[10] Semantic MediaWiki, http://semantic-mediawiki.org/wiki/Semantic_MediaWiki (Last access 16/03/2009).

[11] Semantic Drupal, http://drupal.org/node/299584 (Last access 16/03/2009).

[12] Smarter Wikipedia, http://en.wikipedia.org/wiki/Smarter_Wikipedia (Last access 16/03/2009)

[13] Trella, M., C. Carmona, and R. Conejo: MEDEA: an Open Service-Based Learning Platform for Developing Intelligent Educational Systems for the Web, in Workshop on Adaptive Systems for Web-Based Education: tools and reusability (AIED'05). 2005: Amsterdam, The Netherlands. pp. 27-34 (2005)

[14] Vidgen, S. G: Content, content, everywhere... time to stop and think? The process of web content management. Computer and Control Engineering Journal. Vol. 13, No. 2, pp. 66-70 (2002).

[15] Walsh, E., Dagger, D. and Wade, V.P., Supporting “Personalisation for All” through Federated User Modelling Exchange Services (FUMES). in Towards User Modelling and Adaptive Systems for All Workshop at User Modelling 07, (Corfu, Greece, 2007).