Evolutionary authoring tool for adaptive hypermedia with multi-modal navigation

Nuria Medina-Medina[a], Fernando Molina-Ortiz[a], Natalia Padilla-Zea[a], Marcelino Cabrera-Cuevas[a], Lina García-Cabrera[b] and José Parets-Llorca[a]

[a] *Department of Computer Languages and Systems, Technical College of Computer and Telecommunications Engineering, University of Granada, Spain. e-mail: {nmedina, fmo, npadilla, mcabrera, jparets}@ugr.es*

[b] *Department of Computer, University of Jaen, Spain. e-mail: lina@ujaen.es*

Corresponding author:

Natalia Padilla-Zea

Department of Computer Languages and Systems, Technical College of Computer and Telecommunications Engineering, University of Granada, Spain.

Email: npadilla@ugr.es

Tlf / Fax: +34 958241717 / +34 958243179

Evolutionary authoring tool for adaptive hypermedia with multi-modal navigation

This paper discusses the importance of user adaptation and software evolution in hypermedia applications, and reviews some of the most relevant approaches to both fields. The paper describes a model that has been conceived for the development, maintenance and navigation of adaptive hypermedia systems. This model, called SEM-HP, includes an incremental design process, a layered architecture and an authoring tool integrated with a navigation tool. SEM-HP architecture is composed of four subsystems which allow the separation of aspects related to knowledge representation, presentation, navigation and user adaptation. In addition, SEM-HP has a higher layer, which acts as a meta-system and allows a consistent evolution of the elements defined in each of the four subsystems, as well as their automatic co-evolution. Regarding user interaction, four alternative ways of navigating hypermedia information are supported. Finally, the paper shows the main results of two case studies carried out with the authoring and navigation tool, JSEM-HP, at the University of Granada, Spain.

Keywords: Hypermedia navigation; user adaptation; software evolution and authoring tools.

1. Introduction

Adaptive hypermedia and adaptive web-based information systems (AHS) permit a body of information (structured in nodes) to be browsed, using the connections between them (links) in a personalised way. Consequently, AHS (Graf 2006) offer many advantages, which include the associative retrieval of information, a variety of ways of accessing information, and the ability to adjust the hypermedia to the needs of each user. However, while AHS reduce many of the problems of the traditional hypermedia systems (HS), such as user disorientation and knowledge overload, others arise when using them. One of these drawbacks is that adaptation rules can perform changes to the hyperpages without their having been requested by the users (who may feel uncomfortable because it is not what they were expecting). A further problem is that some AHS force the user to meet all the pre-requisites associated with the desired information, even though this could actually be unnecessary. Consequently, we think that it is important to allow different modes of navigation, so that users can choose between a freer (weaker adaptation) or a more guided (stronger adaptation) navigation, according to their requirements.

From the author's perspective, *building AHS* is more complicated than building HS because it adds adaptation issues (to what, what, how and when adapt?). However, the extra difficulties can be mitigated by suitable authoring tools (Armani 2005; Cristea 2005). There are currently many authoring tools for educational hypermedia, among which we can mention Schoolbook (Kupka *et al.* 2004), GRAPPLE (De Bra *et al.* 2010), REDEEM (Ainsworth *et al.* 2003) or CADMOS (Katsamani and Retalis 2011), but there are not enough general purpose tools. One of the few examples of a general purpose tool is AHA! (De Bra *et al.* 2006), which allows forward and backward reasoning to be combined with user modelling and adaptation, creating systems with very different layouts (based on html frames). An interesting attempt to define a generic AHS is the updated vision provided in (Knutov et al. 2009). Here, the proposed reference architecture suggests the use of ontologies for integration, open corpus for flexibility and new methods for increasing adaptation (group adaptation, context adaptation, multimedia adaptation and adaptation of the adaptation).

We note the relevance of authoring in AHS because hypermedia applications must be constantly updated so that their content and presentation do not become outdated. Therefore, the most critical problem, in our opinion, is that the design, development and maintenance processes of AHS (whole life cycle) are not always treated with the importance they deserve. Specifically, none of the models (AHAM, Munich, GAHM and LAOS) listed as the most important in the review "AH 12 years later" (Knutov *et al.* 2009) defines a formal evolutionary framework. Similarly, none of the authoring systems highlighted in that review (AHA!, KBS Hyperbook, APeLS, Interbook) implements explicit evolutionary mechanisms (evolutionary actions, automatic propagation of changes, etc.) so that the author can modify the AHS with assurances of integrity.

In this context, *SEM-HP* (SEMantic, Systemic and Evolutionary model to develop HyPermedia systems) is presented as an alternative to building AHS. The SEM-HP model integrates:

- An adaptive approach, which enables the development of hypermedia applications capable of fitting to the user during operation.
- An evolutionary approach, which facilitates the evolution of these applications according to the modifications that the author requires to adjust the AHS to new requirements.

SEM-HP guarantees a correspondence between the structure and functionality of the AHS, allowing the development and maintenance processes to be semi-automated (from the author's perspective) and enabling a multi-modal navigation on graphical navigation structures (from the user's perspective). The associated authoring tool, JSEM-HP, implements the evolution of the generated AHS with a set of dynamic objects: evolutionary actions, constraints and propagation of evolutionary actions, and meta-system. This allows an explicit treatment of evolution to be carried out and a clear separation of this aspect at code-level. It should be clarified that the meta-system does not apply artificial intelligence. Therefore, a human must design and promote evolution, and then, the meta-system executes the corresponding changes in a process of "iterative evolution" (which from now on, we will just call evolution).

The tool is particularly suitable for assisted learning because it includes an important set of knowledge attributes in the User Model (UM). However, due to its multi-modal navigation, JSEM-HP can be used to create hypermedia information systems with a more general purpose (e.g., a technical manual, a recommender system or a TV guide).

Briefly, the main contributions of the paper are:

- An incremental design process and a systemic architecture that address the development of AHS from an engineering perspective.
- An authoring tool that provides support for evolution.
- A navigation tool based on graphical maps that make the semantics of the knowledge domain explicit.
- A multi-modal navigation system, which allows the scope of the model to be extended and user control over adaptation to be increased.
- A navigation mode by concepts, which generates customised summaries of the information domain.

These contributions are organised as follows: Section 2 tackles the software evolution in the hypermedia applications. Section 3 introduces the SEM-HP model. Section 4 describes the incremental design process. Section 5 details the layered architecture of the developed AHS and the different modes of navigation permitted. Section 6 outlines results from two case studies (from the perspective of the authors and of the final users). Finally, section 7 summarises the paper and presents our plans for further work.

2. Hypermedia evolution: A crucial and complex issue

The Web is undoubtedly an important pillar of the economy, culture and social relations of modern society, as well as a very successful example of a hypermedia system. However, many existing web applications are poor in design, documentation, usability, accessibility, etc. The problem is so acute that we may even speak of a crisis in the Web (Ginige and Murugesan 2001). Since "starting from scratch" is not a realistic approach, evolution is inevitable (Beck 2010). Moreover, evolution is an inherent characteristic of these applications, mainly due to four reasons: 1) special features of their life cycle (reduced times, lack of specialised tools, etc.), 2) highly dynamic application domains, 3) continuous technological and philosophical changes (rich internet applications, Web 2.0, cloud computing, etc.), and 4) strong interaction with a very diverse audience.

From our point of view, the evolution of web systems is a continuous process that often involves design changes, and which presents two main problems: the difficulty of anticipating an evolving plan (prediction of changes) and the global consistency after the change (impact analysis and change propagation). To address this complex problem, we propose a characterisation of the evolution based on three elements: origin of change, object of change and subject of change. The *origin of change* is the cause that triggers the need for evolution. The *object of change* is the artefact of the system affected by the modification. The *subject of change* is that which is responsible for the transformations.

Regarding the origin of change, we can distinguish two types: logical changes and technological changes. *Logical changes* are triggered by changes in the functional requirements of the hypermedia application (e.g. adding new information to the hypermedia domain or modifying the interaction of a page) and quality improvements (e.g. removing an obstacle to accessibility or increasing security). *Technological* *changes* include changes to the programming language, the execution platform, the architecture (e.g. SOA) and the paradigm (e.g. cloud computing). Regarding the object of change, again we can distinguish two groups: changes to the structure of the system and changes to the functioning of the system. *Structural changes* affect the network of nodes and links (e.g. deleting a redundant link), while the *functional changes* affect the presentation of the nodes and the user interaction (e.g. to distribute a general menu over a list of items). When change is *manual*, the subject of change is often the author of the system (sometimes this responsibility can also be given to the end user). When it is possible to automate the change, the subject of these *automatic* adjustments may be the system itself (adaptation) or a meta-system responsible for controlling its evolution.

Identifying the origin of change is critical to analysing both it and the hypermedia system in which it will be integrated. At this point, we may need to use reverse engineering, particularly if the documentation is insufficient. For example, *web reverse engineering* tools such as WARE (Di Lucca *et al.* 2004) and Revangie (Draheim *et al.* 2005) may help us to recover and to understand the architecture of the web application. Both logical and technological changes may lead to structural or functional changes in the system. In the case of technological changes, different migration techniques may be necessary. Some approaches to *web migration* are RETJAX (Mesbah and Deursen 2007) to migrate to AJAX and the stepwise approach of (Marchetto and Ricca 2009) to migrate to SOA. It is also necessary to take into account the fact that some structural changes may give rise to functional changes. Different mechanisms of *Change Impact Analysis* (CIA) may be used (design rationale, slicing and chopping, crosscutting, etc.) (Mehboob *et al.* 2009) to detect the need for a propagation process, which maintains the global consistency of the hypermedia architecture. The manual changes usually are carried out through mechanisms of reengineering and transformation. For example, a catalogue of *web refactorings* is defined in (Garrido *et al.* 2011) to improve the usability of existing applications by means of the composition of small, semantically equivalent transformations. The automatic changes are performed on-the-fly, and can be executed via transclusion or transcoding, among other possibilities. Transclusion (Nelson 2001) permits the inclusion of new content on a page and facilitates the maintenance between versions. Transcoding is used to adapt the functionality of the system, an example of which is the *web transcoding* proposed in (Sato *et al.* 2007) to improve the accessibility of a web page when it presents flash content. Another different approach to automatic evolution is the *model-driven evolution*, which uses the semantics of formal hypermedia models and meta-models to drive the changes. Some examples of model-driven evolution are the framework for development of web systems based on MDA proposed in (Chen *et al.* 2008), the synchronisation tool based on UWE models described in (Ruiz-Gonzalez *et al.* 2009) and our own proposal (detailed in the following sections).

3. SEM-HP model: Overview

As mentioned, SEM-HP is a semantic, systemic and evolutionary model that allows the construction of AHS, providing the author with three elements (Figure 1): a *development process* that establishes guidelines to create the system from a standpoint of software engineering (section 4), an *architecture* that defines the representation models used to capture the design decisions taken at each phase of the process (section 5) and an software *tool* that implements the creation and exploitation of the AHS (section 6).

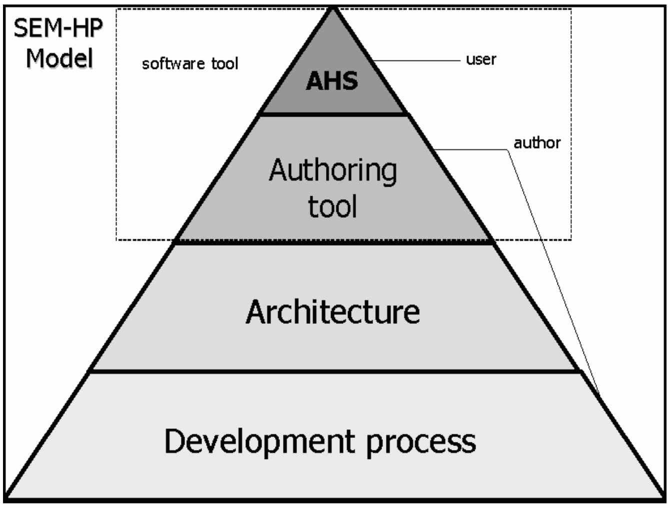


Figure 1. SEM-HP model.

The design and development processes are based on a cognitive model by which the authors can characterise the knowledge domain of the AHS using their own ontologies (García-Cabrera *et al.* 2002). This feature is important in order to maintain consistency with the efforts of the W3C and the IEEE LTTF towards the creation of the semantic Web, in the manner of other authors in this field (Conlan *et al.* 2003; Saksena and Cristea 2006; Frasincar and Houben 2002). To make the author's task easier and more flexible, the design process is divided into four phases, separating aspects of domain modelling, presentation, navigation and adaptation. This allows the author to address the construction of the AHS according to a divide-and-conquer approach, as used in AHAM (De Bra *et al.* 2002; 2004) and its successor GAF (Knutov *et al.* 2010), TANGOW (Carro *et al.* 2001), LAOS (Saksena and Cristea 2006) and CADMOS (Katsamani and Retalis 2011). We provide an authoring tool, JSEM-HP (Molina-Ortiz *et al.* 2009), that generates the AHS's architecture according to the specifications of our model. In other cases, such as the GRAPPLE adaptive learning environment (De Bra *et al.* 2010), it goes further and a suite of tools is offered (GAT: DM+CRT+CAM). In addition, JSEM-HP provides evolutionary mechanisms that automatically ensure the integrity of the changes that the author makes during the construction and reconstruction of the AHS (authoring tool); and supports four types of adaptive navigation from which the users can choose (navigation tool).

4. SEM-HP: Development process

In spite of having special characteristics, the development of an AHS is a software development process and, as such, requires engineering to improve the quality of the final product. Since the first hypermedia reference models emerged (such as Dexter (Halasz and Schwartz 1994) or Trellis (Furuta and Sttots 1990)), other interesting models have arisen for designing and developing HS, such as HDM (Garzotto *et al.* 1993), RMM (Isakowitz *et al.* 1995) or OO-HDM (Schwabe and Rossi 1995). However, these are not valid for AHS because they do not cover user modelling and adaptation issues. For this reason, some specialised models have appeared for AHS, such as the aforementioned AHAM, TANGOW and LAOS, or the Munich reference model (Koch and Wirsing 2002).

Nevertheless, we consider it necessary to make a new proposal to address the design of the AHS with a number of features not included in the existing models, among which we highlight: 1) the incremental development that allows the author to make alternative design decisions in each iteration, so that a set of AHS is generated from a single knowledge domain and 2) the automatic support for maintenance, which provides the author with a set of explicit evolutionary mechanisms to modify the AHS in a consistent way. This allows the integration of two very interesting attributes in our model: flexibility and consistency. This development process (Figure 2) is composed of four phases, and the completion of each phase produces a subsystem of the architecture (section 5).

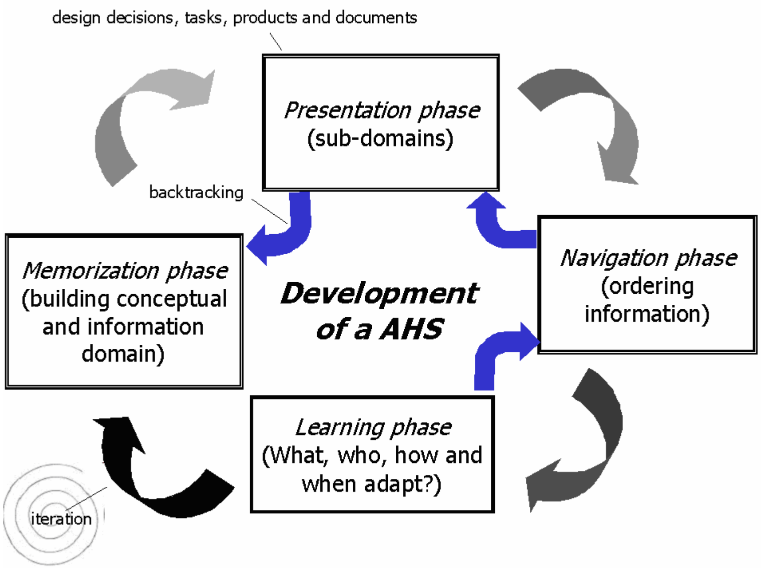


Figure 2. SEM-HP development process.

In the *memorization phase* the author models the information domain. The elements of information are called items and are multimedia pieces (html pages or any kind of electronic resource, such as a MPEG-4 file or a PDF document) of variable size (e.g. a paragraph or several pages). Unlike the basic hypermedia model that only provides content blocks and links, we enrich the content (items) with a body of concepts to explicitly represent the semantics of the information domain. We understand a concept as an idea or abstraction that can be tagged by the authors in order to make their knowledge about it explicit and comprehensible. The conceptual domain also includes the set of semantic associations existing among the concepts. The association between the information domain (data) and the conceptual domain (meta-data) permits the functional relations between the items and the concepts to be labelled according to their roles such as “introduction”, “definition” or “example”. For instance, in a hypothetical information system about a type of yoga called "Hatha Yoga", two possible multimedia items could be a video of the bow pose (“dhanurasana”) and a description of the benefits of the plow pose (“halasana”). The concept behind these two items would be the concept of “asana” (yoga pose). To refer to the interrelation of the information domain with the conceptual domain, we use the term “knowledge domain”. We have adopted this term, instead of the more widely used “domain model”, because we believe that it better reflects the meaning of a body of knowledge that is offered to the user.

In the *presentation phase* the author selects a subset of the knowledge domain modelled in the previous phase. The incremental approach allows the author to create several partial presentations or views (knowledge sub-domains) of the complete knowledge domain. Thus, two presentations may be focused on different conceptual subdomains (distinct concepts), or may treat the same concepts with a different approach (distinct items). In the example above, two presentations could be created: one for the more physical part of the Hatha-Yoga discipline and another for the more spiritual part. Note that, unlike other AHS models, such as XHAM (Cannataro and Pugliese 2001), where the presentation layer prepares the pages to be displayed by the end user, in SEM-HP the presentation phase is a pre-phase. That is, this does not deal with the final presentation of the pages (which will be done in the learning phase), but with the presentation of the knowledge domain (selecting items, concepts and associations, and rearranging their visual placement in each knowledge sub-domain). Of course, it will affect the final presentation and, accordingly, introduce an intermediate level of adaptation.

In the *navigation phase* the author establishes how a generic user can navigate the information items available in a presentation. In this phase, the author decides whether an association between two concepts is navigable in both directions or only in one. In addition, the author can formally define order restrictions among the items. These relationships partially fix the way in which information can be navigated. Therefore, this phase will permit an adaptation that does not depend on any specific feature of the user but only on the order in which he/she moves.

Finally, in the *learning phase* the author solves aspects related to adaptation and user modelling, answering questions such as “to what?”, “what?”, “how?” and “when to adapt?”. As a result, the author establishes mechanisms needed for the system to be able to learn from the user and adjust its operation according to their features (experience, knowledge, preferences, interests and goals). In this phase, the adaptation engine will be configured with rules to:

- Establish knowledge restrictions among the items, which forbid access to some items for which the user does not have the knowledge needed (in our example, the author could prohibit access to the advanced pose of “halasana” before passing, in terms of knowledge, all the beginner “asanas”) and mark as irrelevant other items for which the user knowledge is too high, so reading them would not add anything new.
- Calculate the acquisition of knowledge of the user about the studied items.
- Calculate the user knowledge about the concepts from the user knowledge regarding the associated items.

These phases are iterative, that is, the author can go back to an earlier phase when required (the meta-system manages the changes). In addition, the three last phases are incremental, which permit different presentations to be created in the presentation phase, different navigation strategies for each presentation in the navigation phase, and different adaptation rules for each navigation in the learning phase.

5. SEM-HP: Architecture

The architecture proposed by SEM-HP is structured in layers, performing a double division, as can be seen in Figure 3. The *vertical division* considers four interrelated subsystems. Each subsystem stores the part of the model generated in the corresponding phase of development and provides the other subsystems and users with some functionality. These subsystems are called Memorization Subsystem, Presentation Subsystem, Navigation Subsystem and Learning Subsystem and are described in sections 5.1 to 5.4.

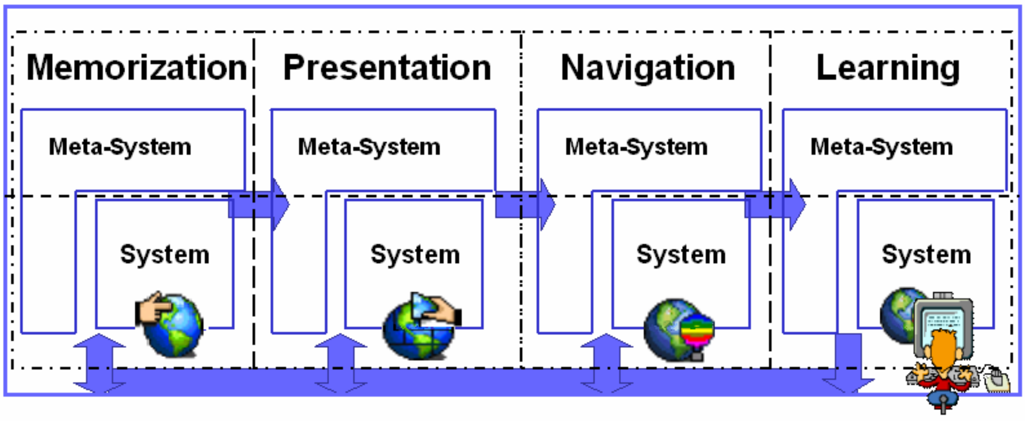


Figure 3. SEM-HP architecture (Medina-Medina *et al.* 2005).

The *horizontal division* distinguishes two layers within the aforementioned subsystems: *system* and *meta-system*. Each layer represents a different level of abstraction. The least abstract level (system) comprises the representation models defined by the author, while the most abstract level (meta-system) includes the evolutionary mechanisms that will enable the integrating and propagating of the changes to these models. We require the existence of an explicit meta-system because we conceive the software evolution (Felici 2003; Mens 2001) as a process that goes beyond user adaptation (argued in (Medina-Medina *et al.* 2002)).

In particular, we implement "evolution driven by the developer" (Torres and Parets 1999). Consequently, the author of the AHS must always initiate the evolution; although some modifications can be suggested by the system itself (e.g. delete an unnecessary conceptual relation), which is able to analyze its usage through the creation of transition matrixes (Medina-Medina *et al.* 2005). After that, the system ensures global consistency, performing additional changes in an automatic mode. In order to perform this iterative development, the evolutionary mechanisms included in the meta-system are: evolutionary actions, restrictions (pre and post requisites) and automatic change propagation. Thus, the author must interact with the meta-system, embedded in the authoring tool, to build and modify the AHS. To perform a change in a subsystem the author must choose the appropriate evolutionary action and run it. The available evolutionary actions are presented in the authoring tool in a friendly way (e.g. "Trash" icon in Figure 4 to remove elements) and internally are implemented as Java objects. In order to ensure the integrity of the system, the meta-system denies inconsistent changes. Therefore, an evolutionary action is only executed if it satisfies a set of restrictions imposed by the model (system restrictions) and by the author (author restrictions defined for a particular knowledge domain). For example, if the author tries to delete the concept "Asana" in Figure 4, the system does not allow the change because the concept "Loto" would be disconnected from the rest (connectivity is a system restriction). The system reports the error and informs the author that the change must be made in a valid way (the solution in the example could be deleting/adding others elements before or using a different presentation). Finally, modifying an element in one of the four subsystems may result in an update to other elements in the same subsystem (internal propagation) or even in other subsystems (external propagation). These two types of propagation are defined as a set of evolutionary actions and are carried out automatically, guaranteeing a coherent co-evolution of the entire AHS architecture.

With the intention of comparing SEM-HP from an evolutionary perspective to other existing works, we cite the hypertext system RICH (Reusable Intelligent Collaborative Hypertext) (Wang and Rada 1998) as being among the few approaches that formally deals with the maintenance of hypermedia. RICH uses a semantic-net-based structured-hypertext model which preserves structural and relational consistency. The RICH system may define and enforce a set of rules to maintain the integrity of the semantic net and, in a similar way to SEM-HP, provide support for creating multi-hierarchies through the reuse of existing content. However, RICH only contemplates the domain and presentation models, while in our proposal the changes are also managed in the navigation and user adaptation models.

5.1 Memorization Subsystem

The Memorization Subsystem stores, structures and maintains the conceptual and information domains of the AHS. Semantic links are established between concepts and extended to items, cataloguing the information items (also called learning objects) in the conceptual domain. This annotation allows the underlying semantics to be handled automatically and, consequently, the navigation and the adaptation processes are improved (Dolog *et al.* 2004). The representation model used for describing the knowledge domain is a conceptual structure. The *conceptual structure* is a semantic network that includes: 1) a set of concepts and conceptual relations, and 2) a set of items and functional relations, denoting the role that the information in the item performs with regard to the associated concept.

The conceptual structure included in the Memorization Subsystem is called the Conceptual Structure of Memorization (CS_M). Figure 4 shows a very simple CS_M in which the knowledge domain is “Hatha-Yoga”. Here, concepts are drawn as ellipses (e.g. “Asana”), conceptual relations as arrows (e.g. “Is_a”), items as rectangles (e.g. I11) and functional relations as lines (e.g. “definition”).

The evolutionary approach allows the author to update the CS_M at any time. To this end, a set of evolutionary actions have been established allowing the creation, modification and deletion of a concept, an item, a functional relation or a conceptual relation. For example, if the author deletes the concept “Ying Yang”, the meta-system automatically deletes the conceptual relation “balance” and the functional relations “definition” (between I8 with “Ying Yang”) and “example” (between I10 with “Ying Yang”). Items I8 and I10 are still available in a central repository if the author considers them necessary. In addition, the meta-system executes an automatic propagation mechanism in the other subsystems.

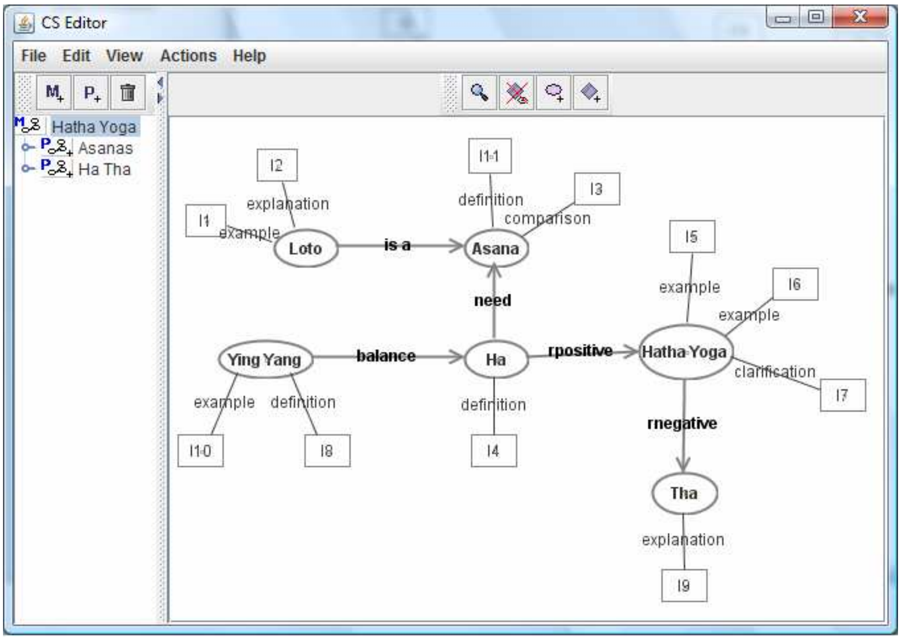


Figure 4. Example of CS_M.

As explained, in SEM-HP, links are stored separately from the final documents. This separation is typical in Open Hypermedia Systems (OHS) such as the FOHM model (Fundamental Open Hypermedia Model) (Bailey *et al.* 2007), and allows different sets of links to be dynamically applied to the content. To this end, both SEM-HP and FOHM internally treat the links as objects of first class. In FOHM, each concept has a set of references (items in SEM-HP), each of which describes it from a different perspective. These might differ semantically (e.g. an item describing the abstract concept of a chair and another showing a picture of a salon chair) or syntactically (e.g.

the same description in different languages). In SEM-HP, the most suitable perspectives are chosen using knowledge rules and guided routes (section 5.5), while in FOHM this is done by using context objects containing metadata about each reference.

The Memorization Subsystem is comparable to the domain model of LAOS (Saksena and Cristea 2006), as in LAOS the relationships among concepts are expressed by means of a conceptual map (formally specified by means of algebraic operators). However, with respect to information items, in our model these are directly associated with the concepts on the semantic net, while in LAOS they are stored as attributes. In fact, a difference between SEM-HP and most existing AHS models, which maintain (in one way or another) a conceptual map, or at least a hypermedia map, is that the maps in these models are hidden, while in our case, the internal representation of the knowledge domain (conceptual structure) is graphically displayed during user navigation (section 5.5). This is an important difference because it allows the end user to know the semantics that underlies the accessed information.

5.2 Presentation Subsystem

In the presentation phase, the author creates a set of possible views of the CS_M, which are stored in the Presentation Subsystem. These partial views (Conceptual Structures of Presentation, CS_P) divide the AHS's knowledge domain in sub-domains, but it is not a strict division because the presentations may overlap. This pursues a double objective: firstly, to reduce the problems arising from browsing structures that are too large, and secondly, to provide the users with a diversity of navigation structures that focus on different knowledge sub-domains, so that they can choose, at each moment, the desired sub-domain. Figure 5 shows two possible presentations created in JSEM-HP from the

Figure 4 (existing CS_P are listed in the left frame, and the chosen CS_P is displayed in the right frame).

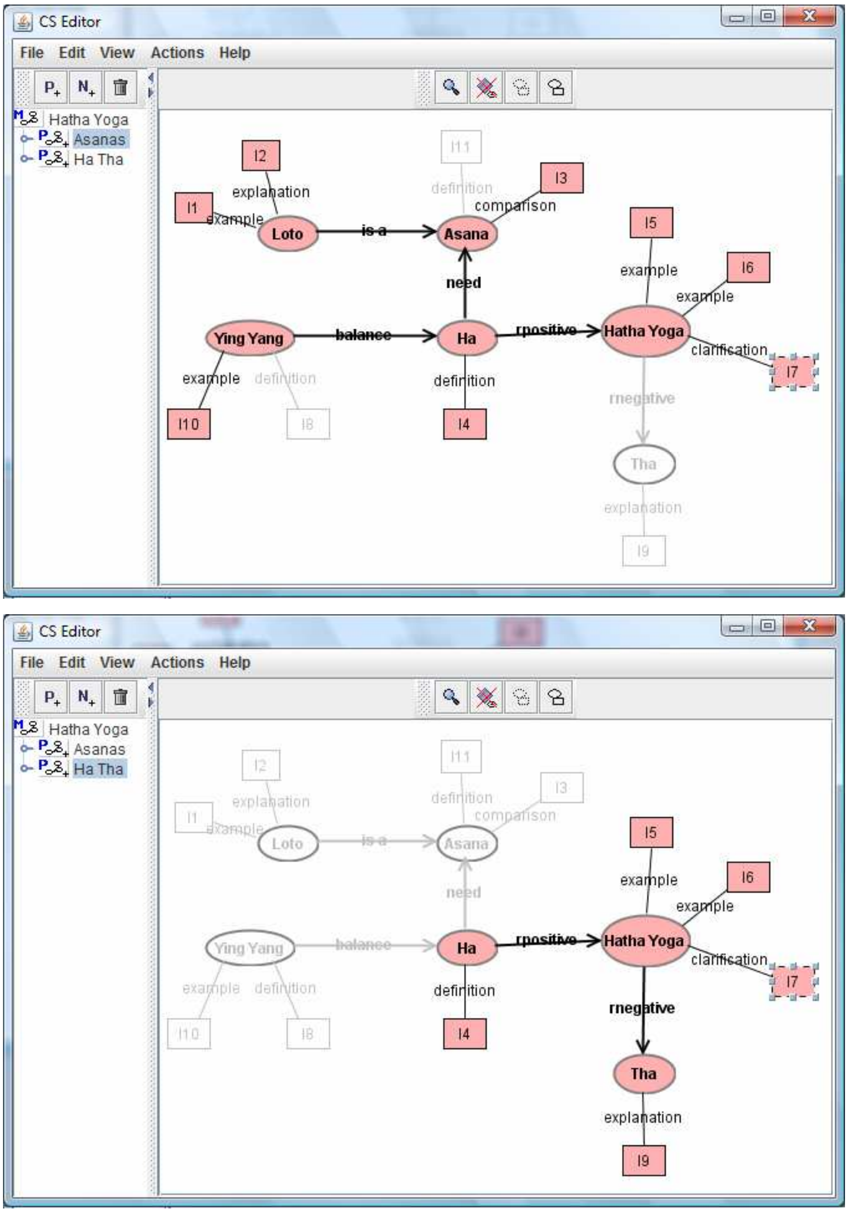


Figure 5. Two different CS_P from the same CS_M.

During the creation and modification of a CS_P, the author shows or hides elements (items, concepts or relationships) of the complete conceptual structure, triggering the evolutionary actions associated with the Presentation Subsystem. The correct application of these changes is controlled by the meta-system (the selected subset must satisfy the same properties as the complete conceptual structure). In addition, the meta-system updates each CS_P with the modifications performed in the CS_M. For instance, if I4 is deleted in the CS_M of Figure 4, this item is removed from the two CS_P of Figure 5.

Some other AHS models also use mechanisms to reduce the size of the domain model. In the case of LAOS, restrictions are established at an intermediate layer called “Goal and Constraints Layer”, which maintains goals, to give a focused presentation, and constrains, to limit the space of the search (Saksena and Cristea 2006). In a similar way to SEM-HP, concepts from the domain model are selected, but in LAOS links that do not exist in the domain model can be created and the labels of some links can be changed, while in SEM-HP, the presentation model only implies a selection process which remains faithful to the initial model (changes in navigation are dealt with separately in the Navigation Subsystem). In the case of RICH (Wang *et al.* 1998), a fish’s eye view is implemented to reduce the complexity of the graph. To this end, the skeleton of the structured hypertext is mapped into a virtual spanning tree. In SEM-HP, the complete knowledge domain may be shown as a bird’s eye view by hiding the information domain (items and functional relations) as shown in Figure 8 (navigation by concepts). And, each CS_P is a knowledge sub-domain that details a fish’s eye view showing all the concepts, items and associations included in that sub-domain.

5.3 Navigation Subsystem

The Navigation Subsystem allows the author to decide the order in which generic users can navigate the structures available from the Presentation Subsystem. To do so, two types of restrictions are defined: navigability and order restrictions. The *navigability restrictions* determine how a conceptual association is navigable. By default, the conceptual relations will be navigated from the origin concept to the destination concept. However, if the author so wishes, the navigability can be extended in both directions. Order restrictions are defined by *order rules* (Ro) and establish a partial order for navigation in each CS_P. This order is based on: the last item accessed by the user, the concept associated with it, and the concepts which it is possible to obtain from that current concept (following a conceptual relation). By default, order rules establish that the user can go to an item, if there is a conceptual relation that can be followed from the last visited item. Therefore, if the last visit in the upper screenshot of Figure 5 was to I4, then the user can visit any item associated with the concept “Asana” (I3) or with the concept “Hatha Yoga” (I5, I6, I7). In addition, the author can add to the order rules the requirement that it be visited before other items not directly connected with it.

For each structure created in the presentation phase, the author can define several ways of browsing it. Thus, a Conceptual Structure of Navigation (CS_N) is well identified by the CS_P from which it is defined and the set of navigability restrictions and order rules imposed on it. The meta-system creates the CS_N by default from each CS_P and provides the author with evolutionary actions to modify the default CS_N or to create new ones.

5.4 Learning Subsystem

The Learning Subsystem is responsible for modelling the users and adapting the structure and operation of the AHS to their personal features. In order to perform this task, the Learning Subsystem applies diverse adaptive techniques, which follow a forward reasoning based on a set of predefined author rules that analyse and update the contents of the UM. Each of these elements is briefly described in the following sections.

5.4.1 User model

The UM stores data about the users, their environment and their usage of the system. This information can be grouped in five categories, which are outlined in Table 1 and detailed in (Medina-Medina *et al.* 2011).

Table 1. SEM-HP UM.

Category	**Attribute**	**Use**
Personal Data	Name, age, gender, occupation, etc.	Personalising the user-system interaction
Experience	Experience in the subject and experience in navigation	Choosing the navigation structure that best adjusts to the user experience
Knowledge	Number of visits to each item	Updating user knowledge about items and concepts
		Inferring the user navigation experience (if it was not explicitly established)
		Performing navigation restricted by conceptual relationship
	Knowledge state: Degree of knowledge about each item and concept	Performing navigation restricted by knowledge
		Inferring the user subject experience (if it was not explicitly established)
Interests	Conceptual sub-domain	Choosing the navigation structure that best adjusts to the selected conceptual sub-domain
	Interesting items	Positively annotating the interesting items on the navigation structure (when they become accessible)
	Knowledge goals	Constructing a guided route to reach the degree of knowledge desired by the user about each item included in the goal
Preferences	Preferences about items: language, author, date, etc.	Generating guided routes whose items fit the preferences specified by the user
	Length of the guided route	Generating shorter guided routes (if the user prefers a shorter route, the preferences about items will have a lower weight when building the route)
	Structure of conceptual summaries	Organising the summary of each selected concept in the navigation by concepts, according to the structure preferred by the user (e.g., first items containing "definitions", then items tagged as "example", etc.)

Most attributes do not require explicit initialisation (all except the personal data), although almost all of them support it. For example, users can indicate if they have prior knowledge of the domain of the AHS, specifying their degree of knowledge about some items and concepts with an approximate percentage or a semantic label (five values are permitted: “null”, “low”, “medium”, “high” and “total”). Nevertheless, if nothing is specified, the degree of knowledge is initialised at “null”. For the subdomain of interest and preferences on items, initialization is not mandatory but strongly recommended.

5.4.2 Update rules

Update rules (Ru), also called “item knowledge update rules”, increase the degree of user knowledge of the items. The default update rules (generated by the meta-system) imply that the user fully knows an item after visiting it (“total” knowledge). For very simple items, this may be a correct assumption. However, when the item is complex, the author may associate a comprehension test and set the obtained result as update. In addition, the author can modify the update rules as needed (using the evolutionary actions). For instance, the update rule of I5 (Figure 5, upper) could increase knowledge about it by two degrees (from “null” to “medium” the first time, from “medium” to “total” the second time) and fix knowledge of I6 to “medium” (after the first reading of I5). For now, no model of retention is taken into account, but users can modify the state of knowledge inferred in their UM if they detect that they have forgotten something (the UM is open and allows both explicit and implicit updating, although the author could disable the explicit update for some users).

The structure of the update rules has a condition-action format, as in the case of AHAM (De Bra *et al.* 2002). However, in AHAM they are generic rules, because they do not assume specific attributes for concepts or specific relationships between them. However, if we instantiate the attributes "knowledge" and "access" for each concept in AHAM it is possible to compare both update models with the following differences: 1) SEM-HP never performs a negative automatic update (atomic concept in AHAM), 2) in SEM-HP, update rules are not executed recursively (in order to avoid cycles), and 3) in SEM-HP, update rules are applied only to items while the knowledge about concepts (abstract concepts in AHAM) is calculated by using weight rules (section 5.4.3).

In SEM-HP, a dynamic object attached to each item is responsible for executing its update rule when the item is accessed while the user fulfils the knowledge requisites. Similarly, in FOHM, each data object (item in SEM-HP) has a behaviour object attached to update the client's UM when the file referred to is displayed (Bailey *et al.* 2007).

5.4.3 Weight rules

The weight rules (Rw), also called "concept knowledge update rules", determine the knowledge about the concepts. Each weight rule calculates the user's knowledge of a concept according to the user's knowledge of each associated item. The meta-system has mechanisms to generate a set of default weight rules, which establish the same weight for all the items. But it also allows the author to modify the weights. For example, giving a weight of 1/3 to I1 and a weight of 2/3 to I2 (Figure 5, upper) if I2 is considered more important to learn the concept "Loto".

5.4.4 Knowledge rules

The knowledge rules (Rk), also called “knowledge restriction rules”, determine which items the user can visit (accessible items) and which items the user cannot visit at a particular time (inaccessible items); and establish which of the accessible items contain relevant information (recommended items). Hence, each knowledge rule for a particular item defines the minimum knowledge that the user should have about a determined set of items in order to access it (accessibility restrictions), and/or the maximum knowledge that the user should have about other specific items to highlight the visit of it (idoneity restrictions). For example, the author could establish that I3 (comparison of “Asana”) (Figure 5, upper) should be accessed only if the user has a "medium" degree of knowledge about I10 (example of “Ying Yang”) and I6 (example of “Hatha Yoga”).

Continuing our comparison, knowledge rules have some points in common with the adaptation rules in AHAM. However, besides the difference cited above, in SEM-HP the pedagogical requirements needed to visit the information are specified outside the domain model. This allows the author to create several knowledge rules for the same item, so that users can follow the learning path that they prefer (from among all the learning paths defined by the author).

The meta-system has mechanisms to generate a set of default knowledge rules by transforming the order restrictions into pedagogic restrictions. That is, if the reading of an item is required in an order rule, a “total” knowledge of the item is required in the corresponding knowledge rule.

5.5 Conceptual Structure of Learning and multi-modal navigation

Again, several Conceptual Structures of Learning (CS_L) may exist for each CS_N (Figure 6). Each CS_L establishes a set of weight rules for the concepts in the CS_M, and includes a set of order, navigability, knowledge and update rules for the items in the associated CS_P. The meta-system can create a default CS_L from each CS_N, and provide the author with evolutionary actions to modify it or to create a new one. Thus, the authors' task can be as simple or complex as they wish. For example, it may be desirable that once the author creates the CS_M the meta-system automatically generates a single presentation, and using both of them (CS_M and CS_P), it builds the UM's structure and all rules by default. On the other hand, the author may define several presentations, several navigations for each presentation and several learning structures for each navigation structure. Anyhow, the CS_L must be labelled in order to allow a personalised choice of the navigation structure. The required labels, which will subsequently be compared with the data stored in the UM are: 1) the list of conceptual sub-domains captured in the associated CS_P and at what percentage, and 2) the ranges of experience (in the subject and in hypermedia navigation) it is advisable to have for the CS_L.

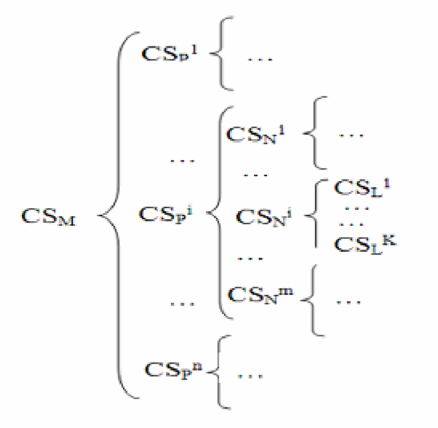


Figure 6. Tree of conceptual structures.

Thus, the adaptation process starts with the *selection of the CS_L that best fits the experience and interests of the user*. In our example (Figure 5), if the user is more interested in the asanas than in the Hata-Yoga philosophy, a CS_L based on the upper CS_P is chosen. Then, other adaptations will be applied to reduce the disorientation and lack of understanding that may arise during the navigation process. As numerous studies support the use of "overview diagrams" (Nielsen 1990) and graphs in particular (Zizi and Beaudouin 1995) to improve hypermedia navigation, our users browse the semantic network directly. As can be seen in Figure 7, the CS_P included in *the chosen CS_L is graphically visualised in the user interface*, which allows users to see the context of navigation at any time. Semantic tags are also shown, so that, the semantics defined by the author in the memorization phase emerge during the user navigation and enrich this process.

A full adaptation will make the task of browsing easier if the users want to make an exhaustive navigation. However, if they only want to query a specific data within an item (for example), the fact that the system forces them to achieve a set of previous prerequisites may be frustrating. For this reason, in SEM-HP, four navigation modes with different levels of adaptation are permitted (Table 2). The first three modes apply an essentially concept-oriented adaptation and the two first are restricted, that is, the users can only access the items for which they have satisfied the imposed conditions.

(1) N*avigation restricted by knowledge*. Restrictions are checked regarding the user knowledge state, disabling the items for which the user does not meet the access restrictions (I3 in Figure 7) and negatively annotating the items for which the idoneity restrictions are not satisfied.

(2) *Navigation restricted by conceptual relationship*. The conceptual relations mark the order to browse the information and restrictions are checked on the visits performed by the user.

(3) *Navigation by concepts*. This navigation mode is free, so, when the user clicks an information element, its content is always opened. The navigation structure only shows the conceptual domain (Figure 8) and, on selecting a concept, the user obtains a summary of it. The structure of the summaries is predefined by

the author and can be customised by the end users, indicating their preference in terms of roles, which results in the order of the sections (first the two “examples” (I5 and I6) and then the “clarification” (I7), in Figure 8).

(4) *Traditional navigation*. The users freely click items in a similar way to browsing the Web.

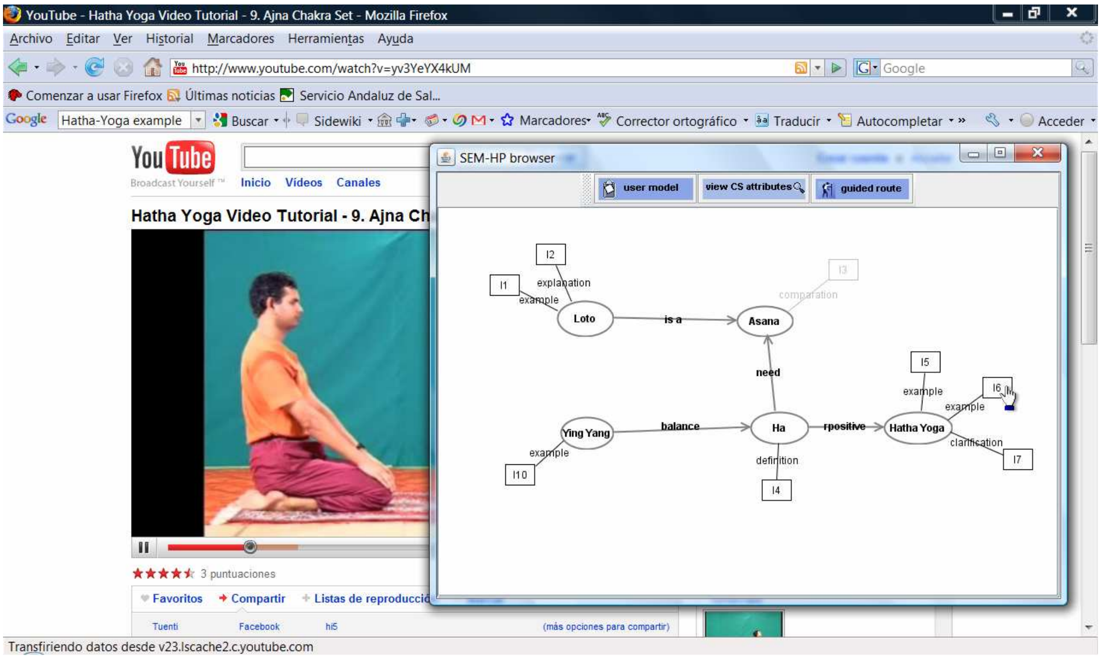
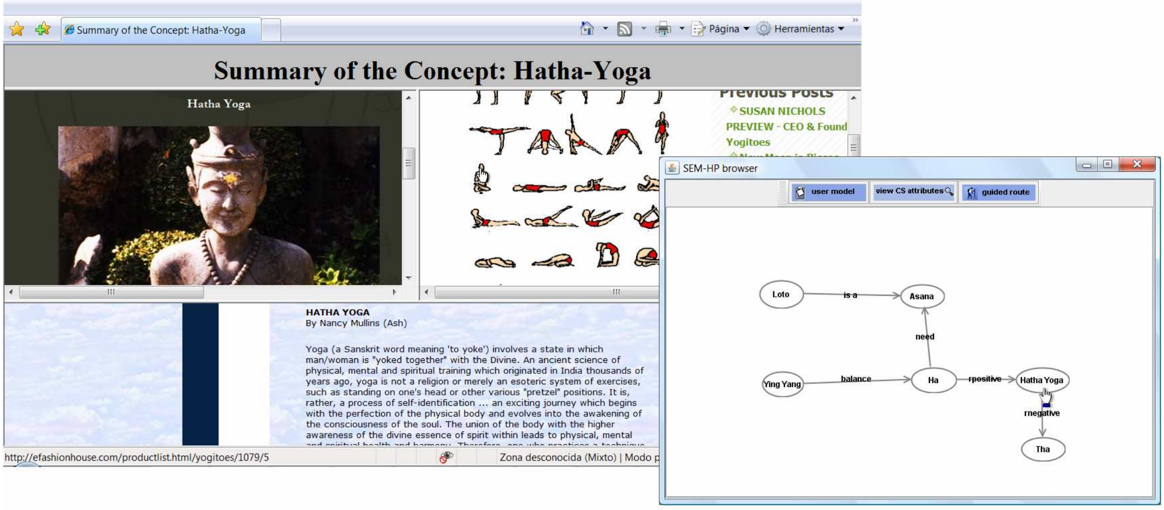


Figure 7. User interface for navigation restricted by knowledge.



HATHA YOGA
By Nancy Mullins (Ash)

Yoga (a Sanskrit word meaning 'to yoke') involves a state in which man/woman is "yoked together" with the Divine. An ancient science of physical, mental and spiritual training which originated in India thousands of years ago, yoga is not a religion or merely an esoteric system of exercises, such as standing on one's head or other various "pretzel" positions. It is, rather, a process of self-identification ... an exciting journey which begins with the perfection of the physical body and evolves into the awakening of the consciousness of the soul. The union of the body with the higher awareness of the divine essence of spirit within leads to physical, mental

Figure 8. User interface for navigation by concepts.

Table 2. SEM-HP navigation modes.

Navigation mode	**Navigation control**	**Navigation element**	**When to use it**	**Recommended knowledge**
Restricted by knowledge	Knowledge rules	Item	Study in depth a knowledge sub-domain	[null – medium]
Restricted by conceptual relationship	Order rules and Navigability restrictions	Item	Browse a knowledge sub-domain in a coherent order	[low – medium]
By concepts	None	Concept	Obtain a bird's eye view	[medium – high]
			Obtain all the information on a specific concept	
Traditional	None	Item	Take a look at a knowledge sub-domain	[medium – total]
			Read specific items	

Depending on the knowledge, intentions, and learning style (Kolb *et al.* 2001) of each user, the *best navigation mode* may be one or another. In addition, the author may prohibit some types of navigation. For example, for an educative AHS, the author may only permit navigation restricted by knowledge if the users have little experience and are expected to do very deep learning. Regardless of the chosen mode, the system updates the *user's knowledge state* according to the update and knowledge rules obtained from the CS_L in use, and shows this knowledge state using a colour code to paint the nodes in the navigation graph: the background of the item or concept ranges from white to black, from the lowest to the highest range of the user's knowledge of it. In each mode of navigation, the user may request a *guided route*, which is constructed taking into account: 1) the current knowledge state of the user, 2) the knowledge state that the user wants to achieve, 3) the features (language, date, etc.) of the available items and 4) the preferences indicated by the user regarding the items and the length of the route. To generate the route, the system uses a search tree (Medina-Medina *et al.* 2005) constructed from the knowledge and update rules defined in the current CS_L. In the search tree, the nodes represent knowledge states, the arcs represent visits to items and the weight of the arc takes into account user preferences.

6. JSEM-HP: Two case studies

To evaluate the effectiveness and efficiency of the SEM-HP model, we have conducted several pilot experiences with the JSEM-HP tool. The general methodology that we applied was divided in two stages: collaborative creation of an educational AHS, and its exploration and exploitation by a group of students. This section presents the results of both stages. In the first case (section 6.1), we focus on the valuation performed by teachers and the assessment instruments we used were informal interviews. In the second case (section 6.2), we focus on the judgment and learning effectiveness of the students, which we measured by means of opinion questionnaires and content tests. In both cases, the study groups were small, so despite the results obtained in the pilot evaluations have been positive, it must be confirmed with further experiences.

6.1 Case study 1: Author perspective

JSEM-HP was used to create an educational AHS whose knowledge domain is Object Oriented Programming (OOP) (Medina-Medina *et al.* 2011). The AHS was designed using the JSEM-HP authoring tool by teachers at the University of Granada, to provide students with a global view of this programming paradigm. Specifically, three teachers of the subject OOP participated in the experiment. Two of them had a high knowledge of the tool (they are authors of this paper) and they instructed the third teacher to use the authoring tool.

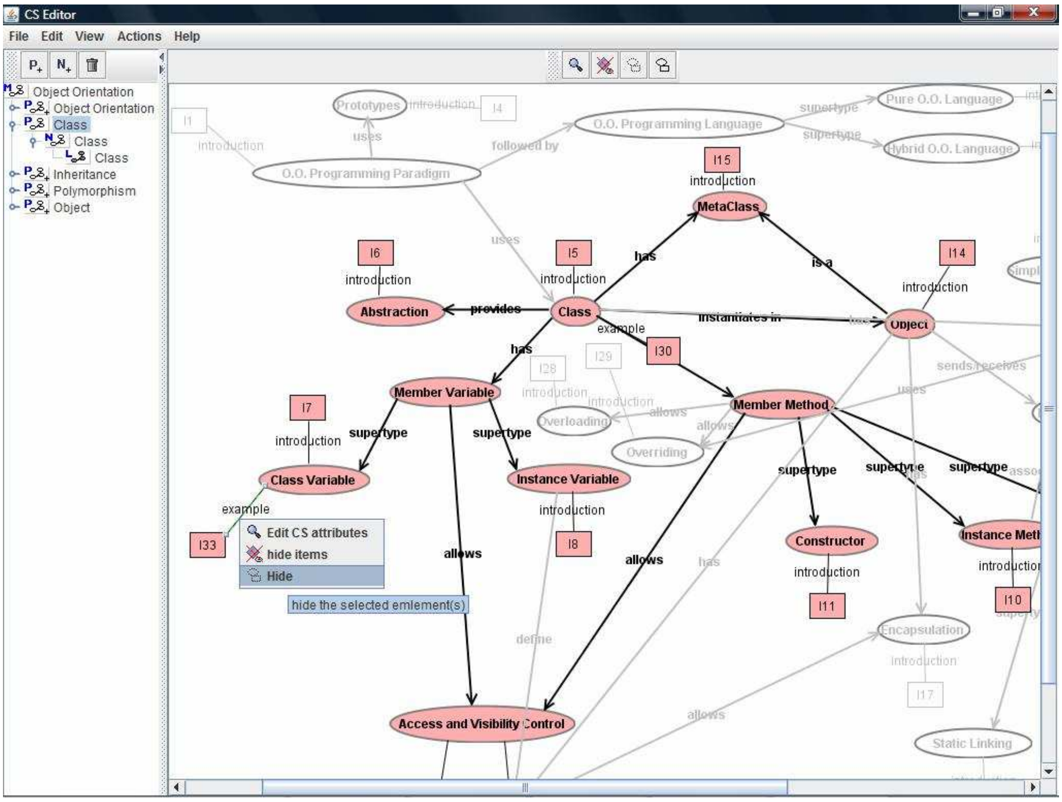


Figure 9. Evolving CS_P^1.

The CS_M of the AHS was comprised of 30 concepts related to object orientation. Each concept was linked to one document containing a brief introduction. In order to avoid comprehension problems, the knowledge domain was divided into four presentations: sub-domain of “Class” (CS_P^1) (Figure 9), sub-domain of “Object” (CS_P^2), sub-domain of “Inheritance” (CS_P^3), and sub-domain of “Polymorphism” (CS_P^4). Each presentation had around 10 concepts, which were obvious at first glance. On each CS_P a unique learning structure was created (CS_L^1, CS_L^2, CS_L^3 and CS_L^4, respectively). Due to the simplicity of the items, the update rules were mostly those generated by default, and the teachers focused on defining the knowledge rules needed to express the educational prerequisites.

The development of the AHS was an evolutionary process. Figure 9 captures the moment at which an author modifies the CS_P^1, hiding the functional relation between the concept “Class Variable” and the item I33. To do so, the meta-system executes the evolutionary action “Hide Functional Association”, which is detailed in Table 3.

Table 3. HideFA(<“Class Variable”, “example”, I33>, CS_P^1).

System prerequisites	None
Author prerequisites	Prevent the erasure of the association if it is the only one with the concept in CS_P^1
Effect	The functional relation "example" is hidden in CS_P^1
Internal propagation	The item I33 is hidden in CS_P^1
External propagation	**In CS_N^1 based in CS_P^1**: Ro(I33) is cancelled before(I33) is cancelled in all the Ro where it appears **In CS_L^1 based in CS_N^1:** Ru(I33) is cancelled Update(I33) is cancelled in all the Ru where it appears All the Rk(I33) are cancelled KnowledgeRestriction(I33) is cancelled in all the Rk where it appears

During the execution, the evolutionary action HideAF is a dynamic object instantiated with the functional association <“Class Variable”, “example”, I33> and the presentation CS_P^1 where the association will be eliminated. The meta-system requests of the action itself the set of prerequisites that must be evaluated before applying its effect. Once consistency is guaranteed, the meta-system runs the effect of the evolutionary action (the functional relationship is hidden). Then, the meta-system requests from the evolutionary action the set of other evolutionary actions that are necessary to propagate the change. In this case, propagation actions affect the Presentation Subsystem (item I33), the Navigation Subsystem (order rules where I33 appears) and the Learning Subsystem (update and knowledge rules where I33 is involved). The meta-system acts recursively with the obtained evolutionary actions in order to keep the whole system consistent. A transactional system permits the return to a previous state (if necessary) and maintains the structural history to facilitate the traceability of change.

After the experience, the teachers rated the editor of conceptual structures as highly positive, as well as the evolutionary mechanisms to ensure the global integrity of the AHS. However, the teachers requested the user interface that is used to work with the knowledge and update rules, currently based on logical predicates, to be replaced by a more graphical version.

6.2 Case study 2: Students perspective

JSEM-HP has also been used to create an AHS whose knowledge domain is JSEM-HP itself. The AHS has been utilised during a session of the course “Hypermedia Systems”, which forms part of the “Master in Software Development” conducted at the University of Granada. The goal was that students would learn the basic architecture of JSEM-HP and see it in operation. With this aim, at the beginning of the session, the teacher gave a 30-minute tutorial, using only the hypermedia system constructed with JSEM-HP. This AHS consisted of five presentations: Overview, Memorization Subsystem, Presentation Subsystem, Navigation Subsystem and Learning Subsystem. All presentations were explained by the teacher except the last, which was navigated individually by the students for 20 minutes. This presentation (Figure 10) consists of 10 concepts and 15 items of varying complexity and format.

Students were randomly divided into two groups. A group of six students (G1) freely navigated the presentation, while the other group, consisting of nine students (G2), browsed the information using the restricted by knowledge navigation mode. Figure 10 shows the initial state of the navigation structure (CS_L^5) for this second group. As shown, 10 items were disabled according to the knowledge rules. For instance, visiting the “example” of "User Model" (I3) was not allowed until the student had a knowledge degree greater than “medium” of I2 (“description” of “User Model”) and I4 (“description” of “Knowledge State”).

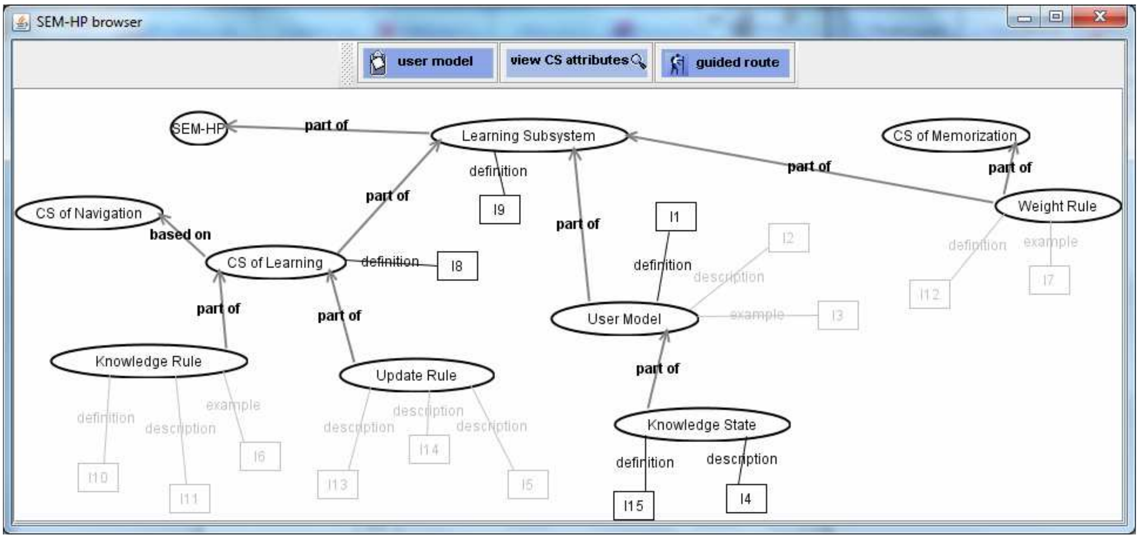


Figure 10. Browsing CS_L[5].

After the experience, the students completed an opinion questionnaire (results in Figures 11.1 to 11.5) and did a little test of understanding (results in Table 4). The opinion questionnaire was divided into four blocks: usability of the tool, browsing experience, user adaptation and comparison with the Web. Students rated each question using an ordinal Likert scale: (1: “nothing”, 2: “little”, 3: “some”, 4: “a lot” and 5: “quite a lot”). The third block (user adaptation) was not answered by the students in G1 who navigated without adaptation.

Regarding the analysis of the results (expanded in http://bios.ugr.es/jsem-hp), we consider “positively relevant” those questions that were valued with 4 or 5 by at least 80% of the students and/or whose average value is equal to or greater than 4. Similarly, we consider “negatively relevant” those questions answered with 1 or 2 by at least 80% of students and/or whose average value is equal to or less than 2. It must be emphasised that no question was negatively relevant.

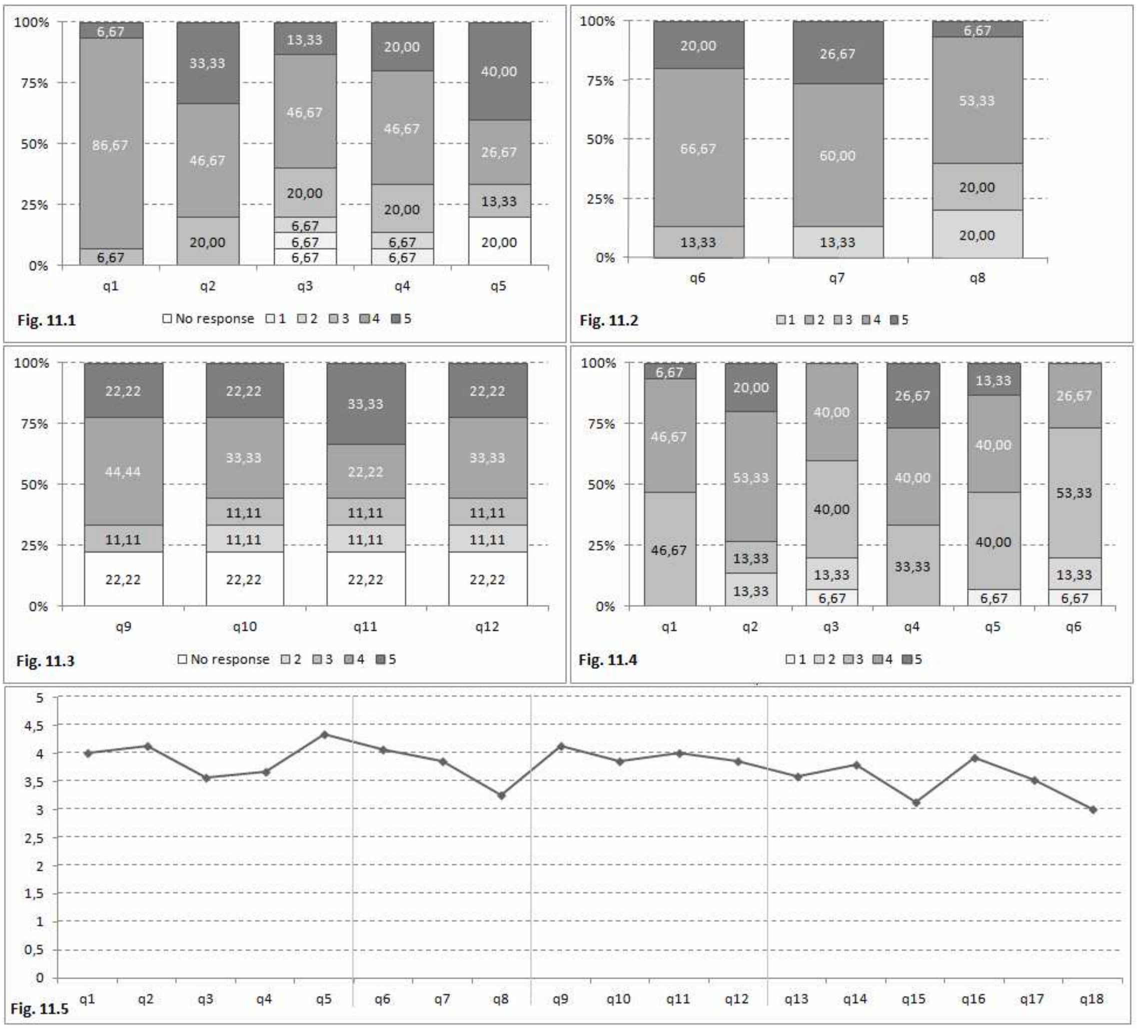


Figure 11. Pilot evaluation of JSEM-HP: Opinion of the students.

The first block’s questions asked in terms of usability about: display of the conceptual map (q1), navigation on the map (q2), existence of different presentations (q3), use of an open user model (q4) and annotation of the state of knowledge on the navigation map (q5). The analysis determined to be positively relevant the questions: q1, q2 and q5, which allows us to deduce that the explicit conceptual structure used in JSEM-HP is an important navigation support, and that students appreciate graphically seeing how their knowledge progresses.

The second block’s questions regarded: general satisfaction during the navigation (q6), absence of sensations of disorientation (q7) and absence of cognitive overload (q8). The positively relevant questions were q6 and q7. Consequently, it is reasonable to think that the browsing experience in JSEM-HP is successful because, among other reasons, it does not produce disorientation.

The third block evaluated the impact of adaptation in reducing disorientation (q9) and reducing cognitive overload (q10), maintaining the flexibility of the hypermedia navigation (q11) without producing a new disorientation due to the adaptations performed (q12). Q9 and q11 were identified as positively relevant, and, therefore, we can presume that adaptation in JSEM-HP reduces disorientation without causing a significant loss of flexibility.

The fourth block compared JSEM-HP with the Web in the following aspects: efficiency (q13), ease of use (q14), speed (q15), ease of learning (q16) and interestingness (q17). Students were also directly asked if they liked the navigation in JSEM-HP more than the navigation in the Web (q18). In this block, no question is remarkable according to the criteria established. In this case, the means are all between 3 and 4. However, we believe that these results are acceptable as we are comparing a new navigation system to the one they usually work with.

Table 4. Pilot evaluation of JSEM-HP: Student learning.

		Traditional (G1 group)	**Restricted by knowledge (G2 group)**
Percentage of correct answers per question	t1	83%	89%
	t2	100%	89%
	t3	83%	89%
	t4	83%	100%
	t5	50%	78%
Average score (out of 5)		4	4,44

Finally, the comprehension test consisted of five multiple choice questions about JSEM-HP (t1 to t5). As noted (Table 4), students in G2 obtained a higher average score than those in G1, 4.4 versus 4 (out of 5). The hit rate was also higher in G2 in all questions except t2. The difference is greater in the last question, t5, which was the most difficult. This leads us to estimate that the adaptation performed by JSEM-HP contributes to the enhancement of the learning process (although the results can not be considered statistically significant).

7. Concluding remarks and further work

In this paper we have argued for the importance of the authoring tools and the evolutionary process in AHS. Consequently, we have presented SEM-HP, a systemic and semantic model that permits the development of adaptive and evolutionary hypermedia systems, supported by three fundamental pillars:

- Systematic development process to obtain a quality hypermedia.
- Double-layer architecture to structure the AHS in four subsystems with well-defined competencies, each of which have a meta-system that manages the representation structures defined.
- Authoring tool to easily create and evolve the AHS.

We have advocated the importance of providing alternative navigation modes, with different levels of adaptation, and have outlined the four navigation modes existing in SEM-HP: traditional (not guided), by concepts, restricted by conceptual relationship and restricted by knowledge (more guided). In all these cases, we have opted to make the knowledge network explicit while the users browse. In addition, the benefits of this and other aspects of the proposal have been analyzed in two case studies carried out with the JSEM-HP tool, in which teachers and students of the “Grade in Computer Science” and of the “Master in Software Development” of the University of Granada positively judged the experience in terms of teaching and learning, respectively.

Our work plans for the immediate future are structured in two directions, which must converge:

- Improving the development of JSEM-HP in order to perform a wider assessment (involving more students and teachers, performing more complete assessment tests and introducing control questions to reduce the risk of cognitive biases).
- Extending the SEM-HP's Learning Subsystem with new adaptive techniques and a more generic UM, according to the challenges defined in (Knutov et al. 2009). Specifically, some improvements we have in mind are to enrich our UM with the physical and technology environment of the user (Cannataro and Pugliese 2001; Vildjiounaite and Kallio 2007), and to study the integration of a layer of storytelling (Brooks 1997) to improve the user interaction and expand the use of the tool to new areas, such as hypermedia novels (Heiden and Ostovar 2006).

Acknowledgements: This research is supported by the --- and the Vice-Rector's Office for Scientific Policy and Research of ---, and the Research Program of the ---.

References

Ainsworth, S. E., Major, N., Grimshaw, S. K., Hayes, M., Underwood, J. D., Williams, B. and Wood, D. J., 2003. REDEEM: Simple intelligent tutoring systems from usable tools. *In:* Murray, Ainsworth and Blessing, eds. *Authoring Tools for Advanced Technology Learning Environments*. Netherlands: Kluwer Academic Publishers, 205-232.

Armani, J., 2005. VIDET: A visual authoring tool for adaptive websites tailored to non-programmer teachers. *Journal Educational Technology & Society*, 8(3), 36-52.

Bailey, C., Hall, W., Millard, D. E. and Weal, M. J., 2007. Adaptive hypermedia through contextualized open hypermedia structures. *Journal Transactions on Information Systems*, 25(4), 16:1-36.

Beck, K., 2010. The inevitability of evolution. *IEEE Software*, 27(4), 26.

Brooks, K.M., 1997. Do story agents use rocking chairs? The theory and implementation of one model for computational narrative. *In: Fourth ACM International Conference on Multimedia*, 317 - 328.

Cannataro, M. and Pugliese, A., 2001. XAHM: An xml-based adaptive hypermedia model and its implementation. *In: Revised Papers from the International Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity*, 252-263.

Carro, R., M., Pulido, E. and Rodriguez, P., 2001. TANGOW: A model for internet-based learning. *International Journal of Continuing Engineering Education and Life Long Learning*, 1, 25-34.

Chen, F., Yang, H., Zhou, H., Qiao, B. and Deng, H., 2008. Web-based system evolution in model driven architecture. *In:10th International Symposium on Web Site Evolution,* 69-72.

Conlan, O., Lewis, D., Higel, S., O'sullivan, D. and Wade, V., 2003. Applying adaptive hypermedia techniques to semantic web service composition. *In: International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH),* 53-62.

Cristea, A., 2005. Authoring of adaptive hypermedia. *Journal Educational Technology & Society*, 8(3), 6-8.

De Bra, P., Aerts, A., Smits, D. and Stash, N., 2002. AHA! meets AHAM. *In: 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems,* 381-384.

De Bra, P., Aroyo, L. and Cristea, A., 2004. Adaptive web-based educational hypermedia. *In: 3rd International Workshop on Web Dynamics*, 387–410.

De Bra, P., Smits, D. and Stash, N., 2006. The design of AHA*!. In: 17th Conference on Hypertext and Hypermedia,* 133-134.

De Bra, P., Smits, D., van der Sluijs, K., Cristea, A. and Hendrix, M., 2010. GRAPPLE: Personalization and adaptation in learning management systems. *In: Conference on Educational Multimedia, Hypermedia and Telecommunications*, 3029-3038.

Di Lucca, G.A., Fasolino, A.R. and Tramontana, P., 2004. Reverse engineering web applications: The WARE approach. *Journal of Software Maintenance and Evolution, Research and Practice*, 16, 71-101.

Dolog, P., Henze, N., Nejdl, W. and Sintek, M., 2004. The personal reader: Personalizing and enriching learning resources using semantic web technologies. *In: 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH)*, 85–94.

Draheim, D., Lutteroth, C., Weber, G., 2005. A source code independent reverse engineering tool for dynamic web sites. *In: European Conference on Software Maintenance and Reengineering,* 168-177.

Felici, M., 2003. Taxonomy of evolution and dependability. *In: 2nd International Workshop on Unanticipated Software Evolution*, 95-104.

Frasincar, F. and Houben, G., 2002. Hypermedia presentation adaptation on the semantic Web. *In: 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH),* 133-142.

Furuta, R. and Stotts, P. D., 1990. The Trellis hypertext reference model. *In: NIST Hypertext Standardization Workshop*, 83-93.

García-Cabrera, L., Rodríguez, M. J. and Parets, J., 2002. Evolving hypermedia systems: A layered software architecture. J*ournal of Software Maintenance and Evolution: Research and Practice*, 14 (5), 389-405.

Garrido, A., Rossi, G. and Distante, D., 2011. Refactoring for usability in web applications. *IEEE Software*, 28(3), 60-67.

Garzotto, F., Paolini, P. and Schwabe, D., 1993. HDM: A model-based approach to hypermedia application design. *ACM Transactions on Information Systems*, 11(1), l-23.

Ginige, A. and Murugesan, S., 2001. Web engineering: An introduction. *IEEE Multimedia*, 8(1), 14-18.

Graf, S., 2006. Book review: Adaptable and adaptive hypermedia systems. *Educational Technology & Society*, 9(1), 361-364.

Halasz, F. and Schwartz, M., 1994.The Dexter hypertext reference model. *Communications of the ACM*, 37(2), 30-39.

Heiden, W. and Ostovar, A., 2006. Structuring hypermedia novels. Technologies for Interactive Digital Storytelling and Entertainment. *In: Lecture Notes on Computer Science*, 4326, 98–103.

Isakowitz, T., Stohr, E. A. and Balasubramanian, P., 1995. RMM: A methodology for structured hypermedia design. *Communications of the ACM*, 38(8), 34-44.

Katsamani, M. and Retalis, S., 2011. Making Learning Designs in Layers: The CADMOS Approach. *In:* IADIS Multi Conference on Computer Science and Information Systems, 305-312.

Knutov, E., De Bra, P. and Pechenizkiy, M., 2009. AH 12 years later: A comprehensive survey of adaptive hypermedia methods and techniques. New Review of Hypermedia and Multimedia, 15(1), 5-38.

Knutov, E., De Bra, P. and Pechenizkiy, M., 2010. Generic adaptation framework: A process-oriented perspective. Journal of Digital Information, 12(1).

Koch, N. and Wirsing, M., 2002. The Munich reference model for adaptive hypermedia applications. 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. *In: Lecture Notes in Computer Science*, 2347, 213–222.

Kolb D., Boyatzis, R.E. and Mainemelis, C., 2001. Experiential learning theory: previous research and new directions. *In: Perspectives on Thinking, Learning and Cognitive Styles*, 1, 227-247.

Kupka, T., Zajaczek, J. E. W., Behrends, M., Walter, G. F. and Matthies, H. K., 2004. Schoolbook: An authoring tool and content management system. *In: Web Education Conference,* 169-171.

Marchetto, A. and Ricca, F., 2009. From objects to services: Toward a stepwise migration approach for java applications. *International Journal on Software Tools for Technology Transfer*, 11(6), 427-440.

Medina-Medina, N., García-Cabrera, L., Torres-Carbonell, J.J. and Parets-Llorca, J., 2002. Evolution in adaptative hypermedia systems. *In: 1st International Workshop on Principles of Software Evolution*, 34–38.

Medina-Medina, N., Molina-Ortiz, F. and García-Cabrera, L., 2005. Diversity of structures and adaptive methods on an evolutionary hypermedia system. *Journal IEE Proc.-Software*, 152 (3), 119-126.

Medina-Medina, N., Molina-Ortiz, F. and García-Cabrera, L., 2011. Adaptation and user modeling in hypermedia learning environments using the sem-hp model and the jsem-hp tool. *Journal of Knowledge and Information Systems*, 29 (3), 629-656.

Mehboob, Z., Zowghi, D. and Lowe, D., 2009. An approach for comparison of architecture level change impact analysis methods and their relevance in web systems evolution. *In: Software Engineering Conference,*162-172.

Mens, T., 2001. Transformational software evolution by assertions. *In: 5th European Conference on Software Maintenance and Reengineering, Workshop on FFSE*, 67-74.

Mesbah, A. and van Deursen, A., 2007. Migrating multi-page web applications to single-page Ajax interfaces. *In: 11th European Conference on Software Maintenance and Reengineering,* 181-190.

Molina-Ortiz, F., García-Cabrera, L. and Medina-Medina, N., 2009. Applying software evolution theory to hypermedia systems. *International Journal of Web Engeneering Technologies*, 5 (1), 69-87.

Nelson, T. H., 2001. ZigZad (tech Briefing). *In: 12th ACM conference on Hypertext and Hypermedia*, 261-262.

Nielsen, J., 1990. The art of navigating through hypertext. *Communications of the ACM*, 33(3), 297-310.

Ruiz-Gonzalez, D., Koch, N., Kroiss, C., Romero, J.R. and Vallecillo, A., 2009. Viewpoint synchronization of UWE models. *In: 5th International Workshop on Model-Driven Web Engineering,* 46-60.

Saksena, M. and Cristea, A., 2006. Towards more efficient generic semantic authoring for adaptive hypermedia. *In: International Workshop on Adaptivity, Personalization & the Semantic Web (APS)*, 11-20.

Sato, D., Miyashita, H., Takagi, H., and Asakawa, C., 2007. Automatic accessibility transcoding for flash content. *In: 9th International ACM SIGACCESS Conference on Computers and Accessibility,* 35-42.

Schwabe, D. and Rossi, G., 1995. The object-oriented hypermedia design model. *Communications of the ACM*, 38(8), 45-46.

Torres, J. and Parets, J. A., 1999. Formalization of the evolution of software systems. *7th International Workshop on Computer Aided Systems Theory*, 269-272.

Vildjiounaite, E. and Kallio, S. A., 2007. Layered approach to context-dependent user modelling. *In: 29th European Conference on IR Research*, 749–752.

Wang, W. and Rada, R., 1998. Structured hypertext with domain semantics. *ACM Transactions on Information Systems*, 16(4), 372-412.

Zizi, M. and Beaudouin-Lafon, M., 1995. Hypermedia exploration with interactive dynamic maps. *Journal Human-Computer Studies*, 43(20), 441-64.