

Implemented and Tested Conception Proposal of Adaptation Model for Adaptive Hypermedia

<https://doi.org/10.3991/ijet.v14i02.8230>

Mehdi Tmimi^(✉), Mohamed Benslimane, Mohammed Berrada, Kamar Ouazzani
Sidi Mohamed Ben Abdellah University, Fez, Morocco
mehdi.tmimi@usmba.ac.ma

Abstract—This article is a continuation of our work concerning the development of the adaptive hypermedia; and precisely the adaptation model, which is one of the three fundamental models that constitute the main architecture of adaptive learning systems.

The adaptation model describes the set of rules of construction and presentation responsible for the construction of the content to be delivered to the learner. Although there is a lot of adaptation model proposed by reference models like (MUNICH and ALEM), these models are on a high level of abstraction and also were never been implemented (lack of concrete implementation).

So, the main purpose of this article is to present our own adaptation model that has the capacity to be implemented and that responds to all the remarks and shortcomings of the existing models while showing implementations and tests (technically) that we have developed.

Keywords—Learner model, Adaptive hypermedia, E-learning, Domain model, state Adaptation model, Adaptation rules, Learning Style, Cognitive ability, Emotional

1 Introduction

This paper joins our work done on Adaptive hypermedia systems and particularly the context of adaptation, these systems are defined as a set of nodes and links that allows one to navigate through the hypermedia structure and that dynamically adapts various visible aspects of the system to individual user's needs, preferences, and knowledge[1].

These systems have a minimal architecture, which consists of a domain model, a user model and an adaptation model [2].

The domain model describes how the information content of the application is structured [3]. Generally, in terms of mechanisms by which the links and nodes are related and navigated, the nodes are treated as general data containers [4].

The user model, learner model in adaptive hypermedia systems,[5]allows the system to maintain a profound knowledge of each learner through the extraction of relevant characteristics that can better describe the person, measure his performance,

motivation, and cognitive capacities, identify his level of knowledge and define his goals, emotions and learning style [6].

The adaptation model represents the mechanism responsible to adapt content, links, representation and structure.

During our previous work [7], we developed a new conception of the learner model mainly based on six facets (personal information, knowledge and skills, historic, psychological profile, cognitive ability, emotional state). And as a next step, we have studied the two remaining models and noted that most of the proposed adaptation models are never implemented (lack of concrete implementation) due to the fact that during the design of these models the technical and programming implementation was not taken into consideration.

Also, despite the fact that each type of adaptation needs to have a specific way of presentation and process of implementation, they are treated with the same way on the conceptual level. This is clearly seen in the Munich Reference Model for adaptive hypermedia applications [4].

Some works neglect even a certain type of adaptation, we quote the work of (Debra, 2008) [7], which takes into account only two types of adaptation: adaptation of content and adaptation of links, however, there are two other types of adaptation: adaptation of the presentation and adaptation of the structure.

And finally, we will focus on the current trend of using open models that have the ability to learn and achieve new outcomes that are not anticipated in advance.

After the study of these different works we thought to change the vision of the work, and instead of developing the domain model then moving to the model of adaptation and finally finishing by detailing the mechanics of adaptation and implementation (technically speaking), we studied simultaneously the adaptation model and its different techniques of implementation and then we intend to move our focus to the domain model which will be designed to describe the whole structure of the elements that both our adaptation model and learner model need. (This way we will have a domain model that answers exactly the needs).

So in this article, we will focus on the developing of our adaptation model and its implementation processes, which is also known as adaptive agent, while separating the different types of adaptation at the conceptual level and on the implementation level, then finally we will test our established model in a web platform while presenting the different algorithms and technologies we have used.

2 Problematic

An adaptive hypermedia system is a set of nodes and links that allow a user to navigate through the structure of the hyperspace and dynamically adapt the different visual aspects of the hypermedia to the needs of the learner.

As the previous definition mentions, there are several aspects of which adaptation and customization can be done.

- **Adaptation of the links:** allows either to restrict the possibilities of navigation in the hypermedia or to propose to the learner new links of concepts.

- **Adaptation of the content:** adjust the content of the pages by providing the most appropriate content to the learner.
- **Adaptation of the structure:** allows displaying the same page in several structures and different forms.
- **Adaptation of the presentation:** allows displaying the same content of the page in different customized styles (size text, color and font.)

All of these different types of adaptation depend on both the domain model and the learner model and they are generally described by a set of rules. These rules describe how pages are constructed and represented to learners. Indeed, many reference models (like Munich [4] and Alem [5]) have proposed a conception of the adaptation model based on the concept of the rules and they will be the main topic of this section.

In both works, we found several remarks and reproaches that make the models proposed incomplete or really difficult to be implemented.

So as we have already mentioned, there are 4 types of adaptation where each one has its own characteristics and its own level of intervention that must be expressed in the adaptation model.

Unfortunately, the two models (Munich and ALEM) have just formally integrated the 4 types at the same level and in the same way without indicating how to envisage adaptation by means of its 4 types. In Fig. 1, we clearly notice that all the four adaptation types are expressed by a 'class' inheriting from the 'Class: Rule'.

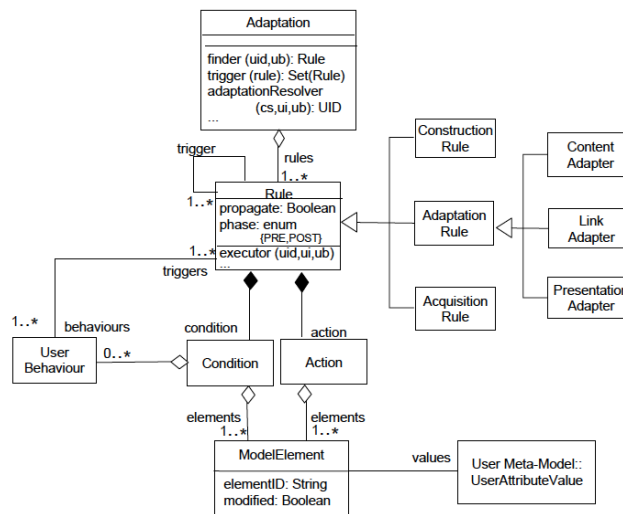


Fig. 1. Adaptation model of the Munich reference model [4]

We have also noticed that the context of adaptation is not fully supported only by the adaptation model, in the sense that adaptation techniques (which are the processes of implementation of an adaptation method) are provided by the domain model.

In fact, both works (Munich and ALEM) consider that the domain model must describe not only the structure of the concepts (knowledge, course, etc.) but also describe the structure of the pages, the components and the styles of presentation.

Finally, ALEM has integrated the possibility of defining new adaptation rules (specific rules) that we named ‘open mode’, unlike Munich where the rules are predefined (generic rule) that we named ‘closed mode’.

The adaptation model is in ‘open mode’ when it supports the creation of new rules that are not defined in advance, this mode allows the adaptation model to evolve and meet the needs of any type of learner, unlike ‘closed mode’ where the designer must set all the rules in advance, which makes the combination number of these rules calculable.

Nowadays, it has become essential that systems must be designed in open mode and especially adaptation models to follow the evolution and diversity of learners' preferences.

3 Reflection and Ideas of Conception

During the early stages of our thinking, we found difficulties in designing an adaptation model that responds to all of the previous detailed remarks because our way of thinking was global and very abstract, indeed when we talk about adaptation we must first answer the following question: what are we going to adapt?

In adaptive hypermedia systems, the different models that compose it collaborate to generate content in the form of a structured, well-presented page and with adequate content that responds to the needs of the learner.

So the adaptation model to be designed must have the capacity to represent the different types of adaptation and also the capacity to contain all the information needed to implement the adaptation. Unlike the existing adaptation models, we will separate from the domain model all that is a structure of page, structure and presentation of components. Therefore, our vision is: the domain model will remain abstract by describing course structures, knowledge, and concepts, etc. And it is up to the model of adaptation to provide the structure and presentation of the page.

This separation will be useful also in the case of importing courses (SCORM for example) from another platform and integrating it in the domain model then use the adaptation model to generate the page and also to adapt it.

In the rest of this chapter, we will detail each type of adaptation while explaining our different ideas of design that we have taken into consideration all the remarks and criticisms discussed previously.

Starting with the adaptation of structure, this type of adaptation specifies how the nodes (component) building the page are connected and ordered. These elements (components) form a tree with an indeterminate depth where each component can have several other components connected to it called “children” and can have at most only one parent; this type of tree can be modeled by a reflexive relationship or by the use of the composite design pattern that will be detailed later.

In addition, the child elements must be ordered. For example in a page we have a section (representing the parent component) that contains a title and a paragraph (representing the children), if in this case we do not have information that indicates the order of these components (title and paragraph) the system can generate an inconsistent section which starts with the paragraph then the title, thus the integration of a property indicating the order is obligatory.

Another concept that needs to be taken into account is that the adaptation model must retain both the original structure of the page designed by the tutors and the structures personalized by the learners.

Regarding the adaptation of the presentation, we think that each component has its own display style expressed by a set of property-value (example: text color = red). However, it is necessary to differentiate between the default presentation styles provided by the tutors and the styles of personalized presentation specific to each learner.

Concerning the two remaining types of adaptation: content and link. We consider that each component can be either a link or a title, an image, a video and can have content and a set of attributes. Let us take an example of a link whose name is 'Introduction' and which refers to the section 'Introduction', this link will be a link component, where the content is 'Introduction' and have an attribute: "address" which has the value "section Introduction".

Finally and as what we mentioned earlier, the content and attribute values will be provided by the domain model, because it's there where all content is structured, so it is sufficient that the adaptation model supports the fact that each component has a content and a set of attributes and develop a process that will choose the appropriate content from the domain model and assign it to the component.

4 Proposition

Following the ideas of conceptions discussed in the previous section, we have elaborated the following adaptation model using UML2 class diagram (Fig. 2) that will be discussed in detail throughout this chapter. Also we used different colors to distinguish the classes used for adaptation processes and the interfaces used as providers of content and data from the learner model and domain model.

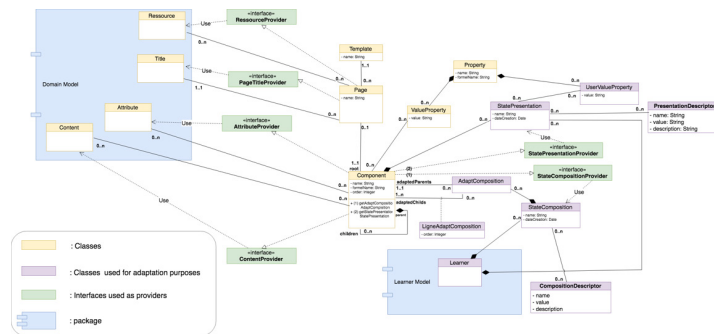


Fig. 2. The class diagram of our proposed adaptation model

In our conception, we have taken into account that there is a set of pages grouped in templates that will be defined according to the recommendations of adaptive teaching taxonomy for each learning style; the following is the combination of the four dimensions according to the learning dimension (LSD):

LSD = {Sensitive (S) / Intuitive (I), Visual (Vi) / Verbal (Ve), Active (A) / Reflexive (R), Sequential (Seq) / Global (G)}

The number of combination is $16 = 2^4$, so for each combination, the tutors must design pages according to the recommendations of each dimension and choose the appropriate electronic media (Fig. 3)

Let's take a concrete example of one of the combinations:

LSDn = (S, Vi, A, S), the pages to be designed for this type of learner must respect the following recommendations:

- Practical content, problem solving, laboratory and experiments → S: Sensitive Dimension.
- Serious games and simulation, presentation → Vi: Visual Dimension.
- Discussion Panel (group work), role-playing → A: Active Dimension.
- Content must be shown in steps (chapters) →Seq: Sequential Dimension.

			Learning styles							
			Sensitive	Intuitive	Visual	Verbal	Active	Reflexive	Sequential	Global
Electronic media	Audio	Audio Recording				X			X	
		Audioconference				X			X	
	Collaboration	Forums	X		X		X			X
		Online learning communities			X					X
		Weblog or blog	X				X			X
		Wikis	X		X		X			X
	Communication	Chat (Messenger)					X			X
		e-mail					X			X
	Diagrams	Animations	X		X					
		Graphics	X		X					
		Pictures	X		X					
		Simulations			X					
	Read	Digital magazines						X	X	
		Digital newspapers						X		
		eBooks			X			X	X	
		Hypertext (web pages)			X			X	X	
		Slideshows			X			X	X	
	Search	Internet research		X			X	X	X	
	Tutoring	Course Legacy System		X						
		Student Response System						X		
		Tutorial systems		X				X		
		WebQuest		X				X		
	Video	Podcast				X				
		Recorded live events			X	X				
Videoconference				X	X					
Videos				X	X					
Web seminars (broadcasts)										

Fig. 3. Adaptive taxonomy: LS dimensions and electronic media Relationship [9]

The complete description of these recommendations of adaptive teaching taxonomy of learning style can be found in the work of Ana Lidia Franzoni and Saïd Assar [9].

Some others recommendations can be taken into consideration like the cognitive capacity of the learner, his gender, etc. So in summary and as shown in Fig. 4, our adaptation model supports multiple templates dedicated to a specific type of learners that include multiple pages where each is dedicated to a type of content to display (course, exercise, introduction).

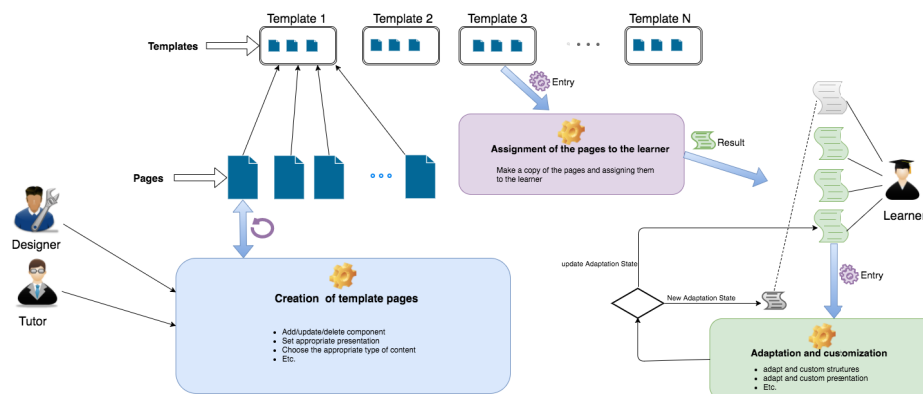


Fig. 4. Our adaptation model in action

Each page is connected to a component (Class Component, Fig. 2) that represents the root (first parent element of the page), and from this root, the system can find all the other components that make up the composition (parent-children) expressed by the reflexive relation.

To express this reflexive relationship, we implemented in our design two versions of the composite design pattern, which compose objects into tree structures [10]:

- Standard version where we represent the original structure of the components
- Advanced version for the custom structures.

When the learner decides for the first time to visualize a page, a copy of the original structure will be made by means of the advanced version of the composition. This copy will be identified by an instance of the class "State Composition".

So when the structure of the page is modified (manually or automatically) for this learner, the system updates one of the copies instead of the original one.

The learner can have the same page in several custom structures that one of them is chosen by the system during the display. This process is illustrated in the class diagram (Fig. 2) by the fact that each component "Class Component" can have several adapted and personalized compositions "Class Adapt Composition" where each one is identified by a state of composition "Class State Composition" that is associated with the learner. Each state has several descriptors and detectors that describe it (these descriptors and detectors are the set of information collected during the creation or modification of this personalized structure: the emotional state of the learner, period of the day.)

In case of conflict, the system can simply choose the most recent state using the date of creation property. We can consider also that these states of composition represent rules and all the detectors and descriptors represent the conditions of application of these rules.

In summary, when the learner wants to visualize a page, the system will generate its structure starting with the root by browsing through all its children provided by the most appropriate adapted composition, and before continuing the path for each child component, a sort of these child components is done using the order property defined in the associative class 'Adapt Line Composition'.

We also want to mention that for the browsing of the whole custom structure, a recursive algorithm is necessary since the structure is in the form of a tree with indeterminate depth.

These two types of composites that we designed allow each learner to have several structures of the pages specific to his/her needs, as well as each modification of one of the personalized structures will have no influence nor on the original structure nor on the others custom structures, without mentioning that open mode is implicitly supported as there is no limit or constraint in creating new custom structures.

A set of algorithm is necessary for the use of these two types of composition that we conceived:

- Recursive conditioned algorithm.
- Insertion in a sorted list.
- Attach and detach a component of the structure.
- Sort algorithm

Regarding the adaptation of presentation, we assigned to each component a set of couplet "property: value" presented by the two classes: "Value Property" and "Property".

This assignment represents the default styles created when developing pages, however, our proposed model allows learners to have custom styles, and this mechanism is realized in the following way:

Each component has several state of presentation "Class State Presentation", which represents states of its display. Each state is associated with a set of "value - property" couplets specific to a learner presented by the two classes: "User Value Property" and "Property".

Similar to the state of composition explained previously, each state of presentation has several descriptors and detectors that describe it and during the generation of the page, the system will choose for each component the state of presentation most adequate.

Our model supports the addition of new property values (open mode) either by tutors or by learners. This addition is used to enrich the database of styles. Let take an example of a paragraph type component; this component has several couplets "property-value" which describes its style of display. Among these couplets, we have for example blue as a color of the text "text color = blue". The system allows the learner to customize this color by proposing a set of text color values from our model, or to use the open mode by allowing the learner to introduce a new value that will be added

first to the model using the Class 'User Value Property' and then attributed to the current state of presentation.

Henceforth this new value will be part of the future propositions of the color of the text.

In summary, the learner can have several display state of components that can be customized independently by selecting one of the existing values of the display properties or by adding a new one, we also mention that in our model, we did not impose restrictions on display property types in the sense that there are some properties that can influence the structure of the page, and here we mention the Flexbox property which is a Box Layout Module that allows and makes it easier to design flexible responsive layout structure without having to use floats or positioning[11].

Regarding the content, we assigned to each component a name property that represents its type, attributes, and content. Let's take an example of an image, this component will have 'image' as value of name property, 'image path' as the value of the source attribute and blank content.

Often the name of the component is not understandable by the learners, especially in a web environment (HTML). For example if we want a component to represent a link, we must assign 'a' as a value for name property, sadly not all learners will understand that 'a' means a link, so we added a property named:[formal Name] to the class 'Component' which will represent a meaningful and comprehensible name for learners and also hide the complexity and the environment of the development from the learners, the same concept is taken up in the names of the 'Property' which does not often have a name understood by the learners. Also in the web environment, we recommend the use of new names introduced by HTML5 that in the near future, computers can start to take advantage of these new smart tags [12].

Finally, we would like to mention that the content is dissociated from the components, in the sense that during the generation of the page, the system chooses the structure and the appropriate presentation to the learner then with the help of a set of providers expressed by interfaces, the system will select the most suitable content from the domain model and assign it to the components while respecting the form of the generated page.

5 Implementation and Test

Since the adaptation model proposed above is abstract and does not depend on an implementation platform, we have chosen to implement it in a Java EE Web Application.

Also, we have chosen to support in the presentation layer a thin client that requires only a browser, present on all machines, which simplifies the work by eliminating the need to stream and install client software on learners' machines [13].

In Fig. 5, we show the architecture of our system, which is composed of three sides:

- **Client Side:** Represented by the web browser and considered as the window of interaction between the learner and our system.

- **Server Side:** It is the main and core part of our system and it is divided into three layers (Presentation, Business and ORM) where each of them is responsible for providing specific services.
- **Database Server:** All transactions made on our models by the system are persisted and stored there.

For better understanding of how our system work, we illustrate in Fig. 5 two scenarios:

- The first scenario presents what happens in the background when the learner decides to visualize content.
- The second scenario concerns the process of adding a new adaptation rule when the learner triggers it.

We also used different colors and numbering for each step of the scenario that we explained in the key of the figure for easier reading and understanding.

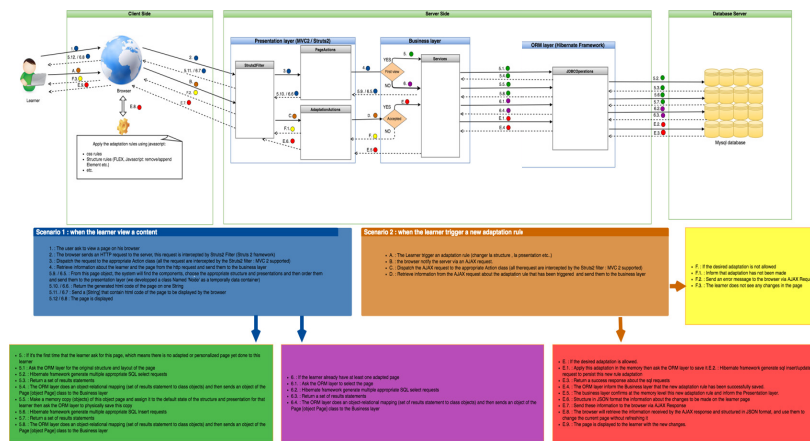


Fig. 5. Global architecture of our web application implementing our proposed adaptation model

Finally, in collaboration with our proposed adaptation model, we developed a set of tools (Fig. 6 and Fig. 7) and processes that proved necessary for the use of our model, then we put them to the test by taking an example of a C++ programming language course that we built its structure and presentation with these tools and then we assigned it this students while adapting the layout, structure and presentation of the elements.

5.1 Tools for creating template pages:

Since the page has to be built respecting the structure expressed in our conception of the model, the tutors must have a creation tool that takes care of all the complexity.

In Fig. 6, we show this web tool that we have developed allowing the creation of pages in real time; the following is a short list of the functionality supported by our tool:

- Add, update or remove component.
- Drag and drop component.
- Add, remove and update a presentation property of component.
- Add, remove and update an attribute value of component.
- Add [formal name] to component.
- Preview of the course on a new window

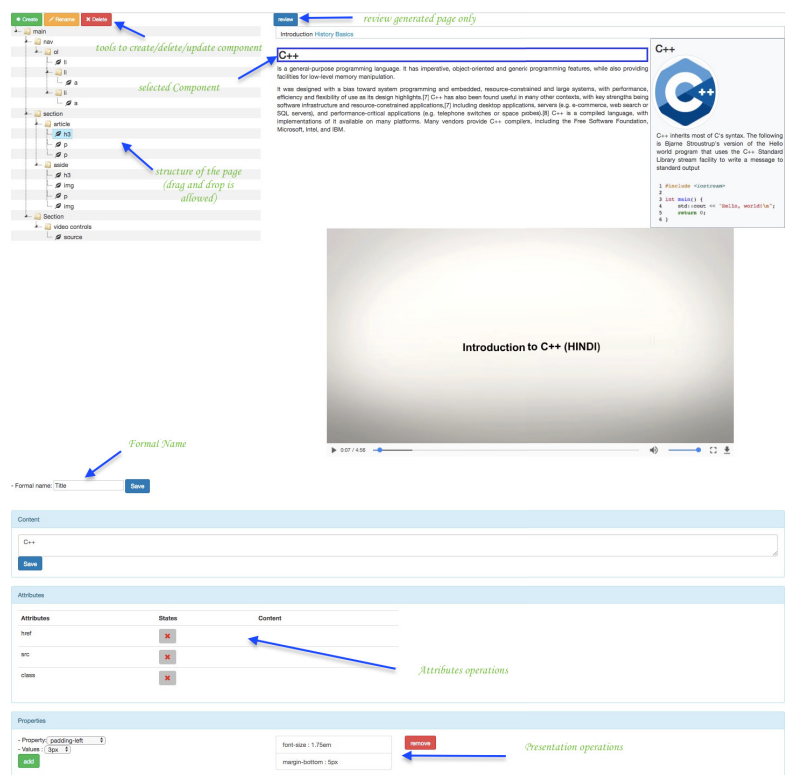


Fig. 6. Screenshot of our web app that represents the tools to create template pages

5.2 Assignment of the page to the learner:

After creating the pages that are grouped in templates where each one is dedicated to a learner type, an assignment of the template to a learner is made by making a copy of the pages and attributing them to the learner, (this concept is well previously explained)

5.3 Adaptation and customization:

In Fig. 7, the system generates the page to the learner while offering the possibility to change the structure and the presentation.

We notice that the order of the components of the page has changed; this new structure (the video is on the top of the page) is only applicable to this learner.

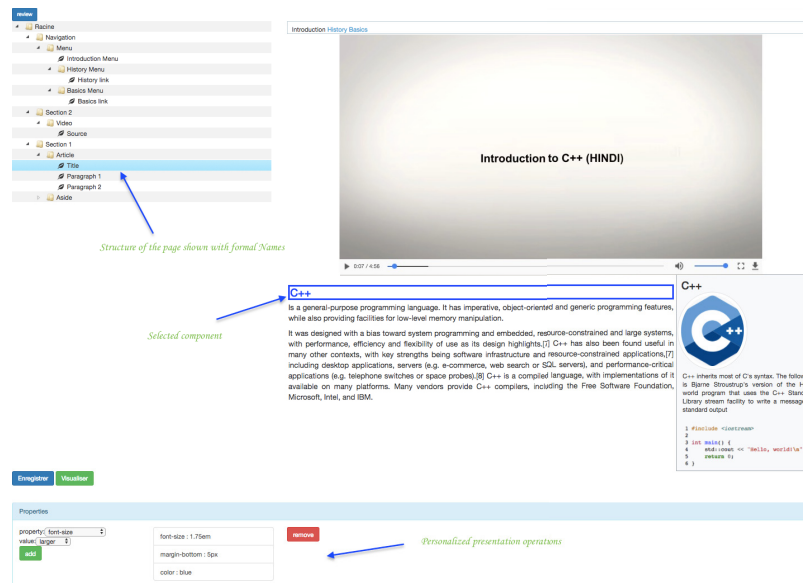


Fig. 7. Screenshot of our web app that represents a custom and adapted page of a learner

And finally, a set of technologies and algorithm have been used, here is a summary:

- HTML5, CSS3, JavaScript / jQuery, AJAX and JSON
- JAVA technologies
- Design pattern: JEE, Composite, Bridge
- Tools: Jstree.js, Bootstrap
- Frameworks: Hibernate, Struts2
- Algorithms: sorting, recursive algorithm, Inserting in a sorted list ...

6 Conclusion and Perspective

The adaptation model that we have designed responds to all the remarks and criticisms cited in the problematic, and has the ability to conceptually represent any page with the different types of adaptation.

Certainly, all the adaptations made and tested previously are made manually but it does not prevent that they can be done automatically. What is important is that our model supports them.

Our model is designed in open mode, i.e. we too as being the designers of the model we cannot imagine how the adapted and personalized pages can look like, in the sense that each learner can introduce new rules (no limit, no constraint), also over time we can retrieve from our model a set of new rules of adaptation that can become a reference if ever was approved by several learners.

Finally, after the success of the tests and implementations that we realized on our model and in addition to our learner model already proposed [7], we are counting on a future work to develop the domain model and also to design an intermediary model that will exploit the learning content generated from the three models (learner model, adaptation model, domain model) to generate an adequate learning content for the target platform (desktop, mobile, web).

7 References

- [1] Brusilovsky P., 1996. Adaptive Hypermedia: An attempt to analyze and generalize. Proceedings of First International Conference on Multimedia, Hypermedia and Virtual Reality 1994. Brusilovsky P. & Streitz N. (Eds.) LNCS 1077, Springer Verlag, 288-304.
- [2] Fayrouz Soualah Alila, 2015. CAM Learn*: une architecture de système de recommandation sémantique sensible au contexte: application au domaine du m-learning. Informatique. Université de Bourgogne, 2015. Français. <NNT: 2015DIJOS032.
- [3] Wu, H., Houben, G.J., De Bra, P., 2000. "Supporting User Adaptation in Adaptive Hypermedia Applications", Proceedings InfWet2000. Rotterdam, the Netherlands.
- [4] N. Koch and M. Wirsing, 2002. "The Munich Reference Model for adaptive hypermedia applications". In: De Bra P., Brusilovsky P., Conejo R. (eds) Adaptive Hypermedia and Adaptive Web-Based Systems. AH 2002. Lecture Notes in Computer Science, vol 2347. Springer, Berlin, Heidelberg https://doi.org/10.1007/3-540-47952-X_23
- [5] Mohammed Tadlaoui, Azeddine Chikh, Karim Bouamrane, 2013. "Alem: A reference model for educational adaptive web applications". Intelligent and Adaptive Educational-Learning Systems, Springer, pp. 25-48, 2013. https://doi.org/10.1007/978-3-642-30171-1_2
- [6] L. Dounas, O. El Beqqali, 2014. "Learner model in adaptive E-learning". Workshop International on New Wireless Technologies and Distributed Systems. WITS.
- [7] M. Tmimi, M. Benslimane, M. Berrada, K. Ouazzani, 2017. "A Proposed Conception of the Learner Model for Adaptive Hypermedia". International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 16008-16016
- [8] De Bra, P., 2008. "Adaptive hypermedia". Adelsberger, H.H., Kinshuk, P., Pawlowski, J.M., Sampson, D. (eds.) Handbook on Information Technologies for Education and Training, 2nd edn. pp. 29-46. Springer, Heidelberg (2008).
- [9] Franzoni, A. L., & Assar, S., 2009. "Student Learning Styles Adaptation Method Based on Teaching Strategies and Electronic Media". Educational Technology & Society, 12 (4), 15-29.
- [10] EBook, Design Patterns Explained Simply: "Composite Design Pattern" Available at <https://sourcemaking.com/design_patterns/composite> [Accessed 13 January 2018]

- [11] The world's largest web developer site, 2017. "CSS Flexible Box Layout Module" Available at https://w3schools.com/css/css3_flexbox.asp [Accessed 13 January 2018]
- [12] Open class rooms, 2017. "Learn how to create your website with HTML5 and CSS3" Available at <https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/structurez-sa-page> [Accessed 13 January 2018]
- [13] Arnaud LABARBE, 2015. "Development of a WEB application, client server vs Angular JS.js". High School of Management of Geneva (HEG-GE). Carouge, le 12 October 2015.

8 Authors

Mehdi Tmimi is from Department of Electrical and Computer Engineering from Sidi Mohamed Ben Abdellah University, Fez, Morocco.

Mohamed Benslimane is from Sidi Mohamed Ben Abdellah University, Fez, Morocco.

Mohammed Berrada is from Sidi Mohamed Ben Abdellah University, Fez, Morocco. His expertise lies in Artificial Intelligence and Software Engineering.

Kamar Ouzzani is from Sidi Mohamed Ben Abdellah University, Fez, Morocco. Kamar's expertise lies in Material Chemistry and Environmental Chemistry.

Article submitted 13 January 2018. Resubmitted 14 May 2018. Final acceptance 05 July 2018. Final version published as submitted by the authors.