WEB CONTENTS AND STRUCTURAL ADAPTIVITY BY KNOWLEDGE TREE
The Herculaneum Excavation Hypermedia

Silvia Rossi[1], Vincenzo Scognamiglio[2] and Ernesto Burattini[1]
[1]*Dip. di Scienze Fisiche, University of Naples "Federico II", Via Cinthia, I80126 Naples, Italy*
[2]*Centro di Ateneo per i Servizi Informativi, University of Naples "Federico II"*

Keywords: Web Personalization, User Modeling, Metamodeling.

Abstract: The aim of this work is the design and the implementation of a prototype for supporting a user through the navigation in a web-site. We exploit the potentiality of having a representation of an hypermedia based on the concept of an abstract semantic description of the knowledge tree, in order to use some adaptivity processes, typical of many hypermedia systems, without the limitation of having a fixed set of resources. Our prototype was implemented for the archaeological Herculaneum Excavations, but the novelty of this work is an attempt to combine methodologies coming from Adaptive Hypermedia, Recommendation Systems and Semantic Web in order to have a context-independent approach for Hypermedia on the Web. What we propose is an architecture with modules that can be reused and adapted in many different contexts. In order to have a context independent behavior the system was implemented with a structural adaptivity itself. This means that changing the tree the system adapts itself to the modifications and dynamically regenerates the code of its components.

1 INTRODUCTION

Nowadays several sites offer personalized portals that can be customized by the user and may even adapt to his interest automatically. In this framework there are many applications that include personal guides for navigation or ambient devices, portals, e-commerce Web-sites (e.g. Amazon, Expedia), or recommender sites (e.g MovieLens).

Usually, in web applications, one can identify two different classes of adaptivity. In the first class, called Adaptive Hypermedia, the adaptivity process is generated *ad hoc* for a specific system. Such systems have precise information on a fixed structure of resources, expressed as the content of each resource and the relationships between them (Goldstein, 1979). These solutions are hardly generalizable for re-use in different applications. A reason why this adaptive functionality is not re-usable is related to the so–called *open corpus problem* in adaptive hypermedia (Brusilovsky, 2004). Adaptive Hypermedia Systems are realized by experts that decide how to connect all the information. That makes difficult to add new material and to connect it.

In the second class, that uses Web Mining techniques, the adaptivity is created in order to operate in a network of Web resources. The are many approaches for creating adaptive web sites (Perkowitz and Etzioni, 2001), but the only adaptive systems that work successfully are the so called Web Recommendation Systems (Brusilovsky, 2004). Those systems may exploit adaptation starting from the word-level content of a resource (Content-based filtering) or recording information of other users with similar interests (collaborative filtering (Rashid et al., 2002)). However the adaptivity process in those systems consists in a list of recommended links that may be ordered according to some ratings. Moreover, an important aspect of Semantic Web (Berners-Lee et al., 2001) is to obtain more and more weight for building effective personalization in web applications. A modern system, in order to show flexibility and a context independent behavior, needs to have a semantic description of resources. Such description may not be bound to each single resource but can be expressed in a more abstract way.

The aim of this work is the design and the implementation of a prototype for supporting a user through

Rossi S., Scognamiglio V. and Burattini E. (2007).
WEB CONTENTS AND STRUCTURAL ADAPTIVITY BY KNOWLEDGE TREE - The Herculaneum Excavation Hypermedia.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Web Interfaces and Applications*, pages 270-275
DOI: 10.5220/0001283402700275

the navigation in a web-site. The adaptivity process is obtained without asking any information to the user that has to interact with the system as a common web-site. Our prototype was implemented for the archaeological Herculaneum Excavations[1] (Sec.2), but the novelty of this work is the attempt to combine methodologies coming from Adaptive Hypermedia, Recommendation Systems and Semantic Web in order to have a context-independent approach that manages Hypermedia on the web. WyW06 – (Whatever) Web You Want – was born as general purpose adaptive system based on user modelling that can be used as an information system on a huge set of data, but in his evolution it becomes different from a classic adaptive system, getting much powerful functionalities typical of recommendation systems. In Sec.2.1 we will introduce our way to represent the knowledge of a hypermedia, based on semantic description of the knowledge tree, while the representation of resources and of user models will be described in Sec.2.2 and Sec.2.3. In Sec.2.4 and Sec.3 we will introduce the adaptivity process in WyW06 and we will describe how the system works.

2 THE HERCULANEUM EXCAVATIONS HYPERMEDIA

The Herculaneum Excavations Hypermedia comes from a stand-alone system by our research group (Burattini et al., 1999). Herculaneum, as Pompeii, was destroyed in A.D. 79 the 24th August by heavy shower of lava from the Vesuvius volcano. The knowledge base contains many information including historical data and mythological tales, excavation reports, description of recovered buildings and objects, painting and so on. This information comes under the form of texts, videos, images, contemporary and old maps. The knowledge was appropriately linked – following the experts' suggestions – and the results of this analysis have been summarized in a tree where single chunks of knowledge are identified (see Sec.2.1). A root node represents one of the main points of view from which the knowledge on Herculaneum Excavations may be explored and subnodes show the ways in which such knowledge may expanded. This first implementation has shown that, even when a great amount of information is managed, the system can be used only from a narrow class of domain expert users since it is unable to adapt its information to different kinds of users. The current system, realized as a web site, is intended for users whose profiles can be different, such as archaeologists, historians, onlookers or just tourists.

[1]http://www.ercolano.unina.it

2.1 A Semantic Representation of the Knowledge Tree

One of the main reasons for the not re-usability of many adaptive system is the so-called open corpus problem. First of all, we have to remark that each adaptive hypermedia, as any hypermedia system, has to make some assumptions about documents and their relationships in a document space. Many of the adaptive applications work on a fixed set of documents that are defined and linked at design time. The process of adding a resource in the system always requires the agency of domain experts and makes more complex the maintenance of the system.

What we propose is an architecture that is context independent and has modules that can be reused and adapted. Our system has no *a priori* knowledge of the web contents, but is able to manage a high level representation defined on this set. The only requirement we impose is that the resources are expressed using some semantic annotation (see Sec.2.2) and that exists a high level representation in terms of a knowledge tree. Web pages or, more generally, any hypermedia content, are generated dynamically starting from those two representations. When the system has to organize the data in order to provide a chunk of information, or to create a web page, it collects the appropriate contents from the database.

In order to clarify this point we have to make a distinction among the concept of *resource class* and the concept of *resource instance*. A content element in the database represents a resource instance. For example, in the case of the Herculaneum Excavation, a resource instance may be the "excavation report" of a particular building or the "description of a fresco". Another example of a resource instance in a different context, such as travel recommendation systems, may be the information of a specific hotel. Such instance gets the semantic description of its class.

A resource class specifies a set of resource instances that have similar semantic properties (see Fig. 1(d)). All the resource classes are semantically interconnected in order to represent the knowledge tree. For example, in the case of the Herculaneum Excavation, the abstract class *building* is connected to the class *graffiti* in the sense that a building may have graffiti. Another example concerning travel recommendation system may be the abstract class *town* connected to the class *hotels*, to the class *tours*, and so on.

At this point we claim that this type of abstract knowledge representation can be easily used in many

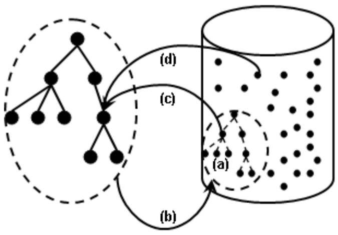

Figure 1: (a) Virtual relationships generated by a knowledge tree that is inserted in the database;
(b) To an instantiated tree corresponds a subset of resources;
(c) To a single resource corresponds a node in one or more instantiated tree;
(d) To more resources in the database may correspond the same node in the knowledge tree.

web applications. For example any recommendation system may be enhanced by the use of this semantic structure. Those systems, in fact, already have, in an implicit way, a semantic representation of how the global knowledge has to be organized and proposed. The knowledge tree defines in an explicit way also the resources and so the contents that we want to personalize and adapt to the user (see Fig.1). For instance some of the nodes that are the leaves of the tree represent a more specific and technical content that can be hidden for a user that is not an expert in the field. Starting from the description of the knowledge tree the system can make a search on the database and can make different instances of the tree using the available resources. In this way we can create virtual relationships among the resources in the database that are not fixed *a priori*. Any modification of the knowledge tree will automatically imply a modification of the virtual links among the resources and the process of adding a new resource does not need the intervention of domain expert to connect it to the others. In fact when a user requests a web page the system will dynamically instantiate the correspondent virtual tree and recognize all the new resources. The resource tree for the Herculaneum Excavations is represented in Fig.2 where each node represents a resource class.

2.2 Representation of the Resources

With the help of a group of architects, historians and archaeologists we detected the main properties that may characterize any information content of our hypermedia. Those features are related to the types of informative content that a resource may have and to the interests that a resource may arise. For example, a feature is the *historical content* - i.e. how much a resource class contains contents coming from historical data. A resource class is a vector $\overrightarrow{r} = (w_i, ..., w_n)$ in the space $\mathcal{R}^n$, where n is the number of characteristics or features (w) and the orthogonal base is represented by the vectors that has a single feature with value 100% and the other features with value 0%. The vector model of a resource class represents how much every specific characteristic is present in the resource (see Fig.3). Differently from the common usage in data mining the vector, that represents a resource, does not contain the frequency of occurrence of some specific words within the text, but represents how much a specific text can be classified according to some typologies of texts. This can be interpreted also as an analysis of the semantics of the keywords within the text.

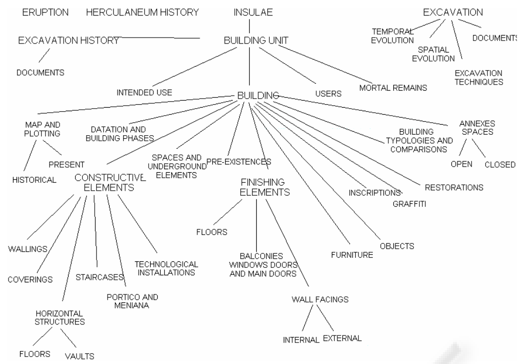

Figure 2: The knowledge tree for the Herculaneum Excavations Hypermedia.

Once the vectors for the resource classes are defined, any resource instance will inherit the model from its correspondent class; this means that all the instances of a class have the same semantic properties. A resource is an atomic and content-closed informative kernel. Each hypermedia page is viewed as specific subset of resources (see Sec.3).


```
<?xml version="1.0"?> <rdf:RDF xmlns:rdf=
"http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.ercolano.unina.it/modelli#">
<rdf:Description rdf:about=
"http://www.ercolano.unina.it/modelli/ModelloRisorsa">
<models:context>Archeological Cultural Heritage</context>
<models:nameOfResource>History of Excavations - archive
reports</nameOfResource>
<models:topographical>100</topographical>
<models:typological>80</typological>
<models:planimetric>0</planimetric>
<models:informative>0</informative>
<models:historical>100</historical>
<models:decorative>0</decorative>
...
<models:technological>0</technological>
</rdf:Description> </rdf:RDF>
```


Figure 3: An example of a vector class in RDF.

In this work we are not dealing with the problem of resources' classification. Resource models are created by domain-knowledge experts and the addition of a node in knowledge tree implies the necessity of defining a new resource model. Resource models are static through the user interaction but can be easily changed by administrator that only has to update the resource model. The system automatically reloads each redefinition when new user session starts. Also the creation of new hypermedia pages, as containers of resources, is obtained dynamically by the system just using the relationships described in the knowledge-tree.

2.3 User Model

In many classic adaptive hypermedia systems the classification of the user is made through the use of stereotypes (Kobsa, 1993), or by clustering a user with a group of people that has shown the same behavior during past interactions. In the first case, stereotypes must be incorporated in the system in the designing phase and, during the execution, the system will have to recognize the correct stereotype (Rich, 1999). In the second case the algorithms for clustering and the adaptivity process – i.e. the recommendation process – are exploit just recording the already visited web pages or the rankings made by previous users and they need a large amount of data in order to be effective.

Taking the advantage of the semantic descriptions of resources and of the knowledge tree, we decided to implement a user model that takes into account only the current behavior of the user starting from the specification of the behaviors of ”ideal” users. What is different from the classical use of stereotypes is that we do not try to classify a user in a specific class but we start from the assumption that a real use may exhibit a behavior that is a combination of ideal classes. An ideal users' class is the representation of a set of users, like a prototype, whose behaviors and interests are formally defined. It is fundamental that each ideal class of users does not have any behavior that is in common with another ideal class, i.e. the vectors representing ideal classes are an orthogonal base.

A user model is a vector $\overrightarrow{u}$ in the space of ideal user classes. When a user session start, the user model vector components are set to zero. During the user interaction, the choices made by the user lead the system to change the values of the user model. The user model vectors in WyW06 are represented as couples attribute–value and such values represent the percentage of similarity of the user model to an ideal user class (see Fig.4). This is to say that the user model $\overrightarrow{u_i}$ of the user i is a linear combination of ideal user's models ($\overrightarrow{\underline{u}_i}$), such as $\overrightarrow{u_i} = \alpha\overrightarrow{\underline{u}_1}, \beta\overrightarrow{\underline{u}_2}, ..., \gamma\overrightarrow{\underline{u}_m}$, where Greek letters represent percentages.

The content-independency of the system, obtained for resource classes, is kept also in the management of the user model. In WyW06 this vector represents a user model in an explicit way, starting from the definition of ideal classes. The modification or the adding of a new ideal class implies an automatic adaptation of the system to the new configuration, without the need of implementing new code since it is generated by the system itself.

```
<?xml version="1.0"?> <rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#
"xmlns:cd="http://www.ercolano.unina.it/modelli#">
<rdf:Description rdf:about=
"http://www.ercolano.unina.it/modelli/Modelloutente">
<models:context>Archeological Cultural Heritage</context>
<models:archaeologist>90</archaeologist>
<models:architect>25</architect>
<models:historian>62</historian>
...
<models:tourist>70</tourist>
</rdf:Description> </rdf:RDF>
```

Figure 4: An example of a user model vector in RDF.

2.4 Extension and Association Rules: Adaptivity

Up to now, we described the user model and the resource model as two distinct concepts, without mentioning how to go from one representation to the other. The user model changes according to the choices that the user made browsing the resources, and the adaptivity process on resources depends on the current user model. In Sec.2.2 we proposed the user model as a composition of orthogonal ideal user classes. Associated with each ideal class there is the specification of the ”interests” of an ideal user. This specification is made by using *association rules*. In the other way the updating of the user model is made though the application of *extension rules* – e.g. rules that specify how to modify the user model after a specific event.

Usually these rules are specified using statistical methodologies on real data. In our case an association rule is represented by the definition of a resource class ($\overrightarrow{r_i}$) considered optimal for an ideal user ($\overrightarrow{\underline{u}_i}$). Such class is the one that has the typological content more similar to the user interests. Moreover, the resources, selected to be representative for the ideal user models, are linearly independent, in order to respect the fundamental assumption that the ideal users represent an orthogonal base (see Sec.2.3). To verify the consistency of the user model representation, the system, whenever there is a change in the ideal users' specifications, calculates scalar products among the resources associate to the ideal users and verifies that they are orthogonal.

When a user requires a new hypermedia page, the system takes the user model, computed during the interaction up to that moment, and calculates the ideal resource for the current user ($\overrightarrow{\underline{r_i}}$). Then it compares a distance ($diff$) respect to each real resource ($\overrightarrow{r_j}$) in the selected page and, using a threshold for the obtained distance, makes decision about which content of the resource it has to show or hide. This process is summarized in the following algorithm. Moreover we want to remark that an ideal resource may not correspond to a real resource.

Algorithm 1 DIFFERENCE($\overrightarrow{u_i}, \overrightarrow{r_h}$)

▷ *user model* $\overrightarrow{u_i}$, *resource class* $\overrightarrow{r_h}$
▷ $\overrightarrow{r_h} = (w_{h1}, ..., w_{hn})$
▷ $\overrightarrow{u_i} = (\alpha\overrightarrow{\underline{u_1}}, \beta\overrightarrow{\underline{u_2}}, ..., \gamma\overrightarrow{\underline{u_m}})$

for $j = 1, j = n, j{+}{+}$
    $\underline{r}_{ij} = max(\alpha r_{1j}, \beta r_{2j}, ..., \gamma r_{mj})$
    ▷ $\underline{r}_{ij}$ *is the j component of the ideal resource* $\overrightarrow{\underline{r}}$ *for the user i.* $\underline{r}_{ij}$ *is the maximum value of all the j components of the resources* $\overrightarrow{r_m}$ *associated to the ideal classes of users* $\overrightarrow{\underline{u_m}}$ *and multiplied by the same factors as in the user model*
$diff = 0$
for all $\overrightarrow{r_k}$ $secondaryNodesOf(\overrightarrow{r_h})$
▷ *For all the secondary children of the node*
    for $j = 1, j = n, j{+}{+}$
    ▷ *For all the features of a node*
        if ($\underline{r}_{ij} > r_{kj}$)
        ▷ *The user is not satisfied*
            $diff = diff + 1$
    $distance = 100 - (\frac{diff}{n} * 100)$
    ▷ *Compute the percentage of fields that are not satisfied*
    if ($distance <= threshold$)
        show($\overrightarrow{r_k}$)
    else
        hide($\overrightarrow{r_k}$)

The metrics we use to compute the distance starts from the assumption that a feature of a real resource "satisfies" the user interests if his value is equal or greater than the value of the same feature in the ideal resource. This is to say that the real resource is appealing for the user at least for this specific feature. For example if one has an interest of 20% for the historical property, every real resource with the historical feature greater than 20% will satisfy his interest. For each real resource, within a web page, the system computes how many features do not satisfy the user interests. If this number is low the content will be shown otherwise it will be hidden. In future work we are planning an evaluation of this metrics. It is important to highlight that the system lets the user free to show the content of resources considered not interesting for him and to hide the content of resource automatically showed.

Finally, as we said, associated with each resource there is an *extension rule* that specifies how to changes the user model when the resource is activated. This rule refers to the nodes of knowledge–tree but not to the resource instances. The adaptivity process in WyW06 is achieved just by analyzing the user's interactions with the system environment without any explicit questions about preferences.

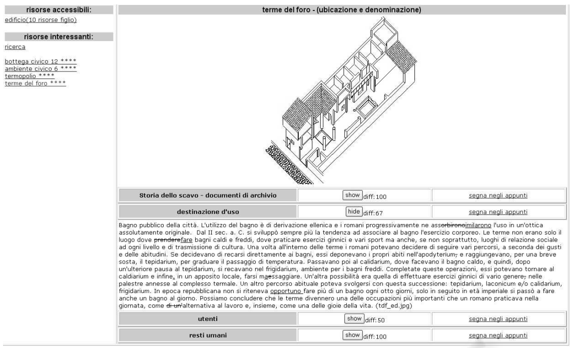

Figure 5: A snapshot of the prototype of the Herculaneum Excavation web hypermedia.

3 SYSTEM AT WORK

WyW06 introduced a new and potentially powerful layer of adaptation: the structural-adaptation. This structural adaptation refers to the ability of a system to create or regenerate some of his modules, dynamically and autonomously, when some specific parameters are changed.

To translate the tree in a hypermedia with his relative hyper–linked structure, WyW06 makes a fundamental and dynamical distinction between main nodes and secondary nodes. How to make this choice is relatively important, anyway, the current choice is to evaluate the number of child of a specific node. If it is smaller than a defined threshold then this node is classified as secondary, a main node otherwise. The addition of new instances of a resource does not require neither a complex intervention of system managers – the system automatically integrates the new resource – nor the intervention of knowledge-domain expert – the semantic of any new element refers to his specific class model. The addition of a new node in the tree is not absolutely an issue for the system that regenerates parts of code of his modules and the hypertext structures without other interventions.

At the beginning of a session each main resource will appear as a web page with his main resource content and the list of resources that are children of the main. If a main resource is a child of another, it will appear as a link to another web page in the left box (see Fig.5). If a secondary resource is child of the main resource, it will appear as a expandable text[2]

[2]With the term expandable text we mean a subsection of

(see Fig.5) that user can expand in every moment. When user expands or hides a secondary resource, its associated extension rule is applied to the user model. When the user model is sufficiently defined for the system criteria and a user accesses to a main resource, the system automatically expands the content of secondary resources that are considered interesting for him. The system also provides to the user the possibility of saving resources as personal notes. This action also implies the application of extension rules to the user model.

For obtain the multilevel adaptation we implemented the system using different language on server-side and client-side. We used JavaScript combined with Cascade Style Sheets (CSS) to realize the dynamic update of the fields of the user-model, resulting by the application of extension rules, and the user's choice about viewing/hiding resources on client-side. For the server-side, we used the PHP language to obtain the dynamic generation by the server of the contents–selection for the web pages. The PHP code has the important role of applying the association rules in order to obtain the user-model transformation. The structural adaptation is made representing web site structure, and relative resources, by knowledge-tree. The real resources, the rules and the knowledge-tree are stored in a database set up in MySQL environment managed dynamically by PHP code.

4 DISCUSSION

WyW06 was born as general purpose adaptive system based on user model. Like classic adaptive system WyW06 has the aim of supporting individual user in finding, selecting and managing content offered by web sites. In his evolution WyW06 becomes quickly more than a classic adaptive system, getting much functionality typical of recommendation systems, melting the advantages of both those typologies of systems. WyW06 is based on the point of view that each single web page is like a collection of resources. A resource is an atomic informative kernel semantically linked to other resources by a specific tree of relationships. The use of a tree to specify relationships between resources is a peculiar characterization of adaptive hypermedia systems (Burke, 2002). In the adaptive hypermedia a common problem is the adding of a node-resource in the system. As we said in section 2.1 to overcome this obstacle, WyW06 introduced a new and potentially powerful layer of adaptation: the structural-adaptation. The structural adaptation refers to the ability of system to create or regenerate some of his modules, dynamically and autonomously, when some specific parameters are changed. Respect to classical definitions in adaptive systems WyW06 is a user model based system. The adaptive behavior of the system aims to satisfy automatically the desires of the user but, it is important to highlight that, lets the user free to access to the resources that are not near to the self-selection threshold. The use of such techniques leads the system easily to become a recommendation system. In fact the system evaluates the secondary resources that are of more interest respect to the actual state of the user model and propose them to the user. Summarizing we can say that WyW06 has a behavior comparable to an adaptive hypermedia, some peculiarity of content-based recommendation system, some characteristics of inferential reasoning by using rules schemas, some fundamental notion for Semantic Web and something innovative respect all, that is the structural adaptive layer.

the web page that can be hidden or shown by a click on a button.

REFERENCES

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific Am.*, pages 34–43.

Brusilovsky, P. (2004). Adaptive navigation support: From adaptive hypermedia to the adaptive web and beyond. *Psychnology*, 2(1):7–23.

Burattini, E., Gaudino, F., and Serino, L. (1999). Hypermedia knowledge acquisition and a bdi agent for navigation assistance. a case study: Herculaneum excavations. In *Europ. Conf. on Cognitive Science*, pages 437–440.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.

Goldstein, I. P. (1979). The genetic graph: A representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies*, 11:51–77.

Kobsa, A. (1993). User modeling: Recent work, prospects and hazards. In *Adaptive User Interfaces: Principles and Practice*, pages 111–128. North-Holland, Amsterdam.

Perkowitz, M. and Etzioni, O. (2001). Adaptive web sites: Concept and case study. *Artificial Intelligence*, 118(1–2):245–275.

Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., and Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. In *IUI '02: Proc. of the 7th int. conf. on Intelligent user interfaces*, pages 127–134, New York, NY, USA. ACM Press.

Rich, E. (1999). Users are individuals: individualizing user models. *International Journal of Human Computer Studies*, 51(2):323–338.