Dynamic Generation of Adaptive Web Catalogs*

Liliana Ardissono and Anna Goy

Dip. Informatica, Università di Torino, Corso Svizzera 185, I-10149 Torino, Italy
{liliana,goy}@di.unito.it

Abstract. This paper describes the techniques used to dynamically generate personalized Web catalog pages in a prototype toolkit for the creation of adaptive Web stores. We focus on the integration of personalization strategies for selecting the layout and content of the catalog pages, with Natural Language Generation techniques, used to dynamically produce the descriptions of products, tailored to the individual user.

1 Introduction

With the increasing popularity of e-commerce, the development of adaptive Web catalogs has become a central issue. Several commercial tools for the creation of on-line stores tailor the interaction to the customer by suggesting goods on the basis of her/his preferences; however, they adopt quite simple, if any, techniques for personalizing the presentation of items. Most Web catalogs fail to provide the information relevant to the customer's interests and, especially in technical sales domains, challenge her/him with very complex descriptions. Finally, the internal organization of the catalogs unavoidably reflects the designer's view of the sales domain, which can hardly correspond to the customer's one, making it difficult for the user to find the products (s)he needs. In this scenario, the quality of Web stores could be enhanced by dynamically generating personalized catalogs during the interaction with the customer.

The personalization of the presentation style concerns several aspects, including the customization of the page layout and of the media used in the presentation (e.g., see [11, 13]), the inclusion of targeted advertisements, and so forth. In this paper, we present the strategies exploited to dynamically generate personalized catalogs in SETA, a prototype toolkit for the creation of adaptive Web stores developed at the CS Department of the University of Torino [2]. In [3], we described the strategies used to handle the user models and personalize the selection of the information to be presented. Here, we focus on two main issues: a) enhancing the accessibility of Web catalogs by providing the user with information about their organization; b) generating rich presentations, where different types of information about products are conveyed in structured descriptions.

* This work extends the SETA system, developed in the "Cantieri Multimediali" initiative funded by Telecom Italia. In particular, we thank Cristina Barbero for her contribution to the design and implementation of the NLG module described in this paper. This paper has been published in **LNCS n. 1892, Springer Verlag.**

The Web store catalogs generated by SETA are organized as two-level hypertexts: the first level includes pages sketching the main characteristics of the product categories; the second level includes pages describing in detail the items available for each product category. Although the transparency of the catalog could be further improved by providing extra information, such as a map of the store, the separate presentation of product categories and individual items helps the user to recognize whether a product category is relevant to her/his needs as soon as (s)he visits the related page, without inspecting the available items.

SETA exploits template-based Natural Language Generation (NLG) to dynamically generate the descriptions of the product categories and their items: in the descriptions, the system merges different types of information about features and properties of the presented goods. This approach reduces the amount of pre-compiled information to be defined at configuration time: in fact, the NLG module used to generate the product descriptions retrieves the data about products from a single internal information source and supports the generation of text in different languages as well as the production of alternative descriptions of items, tailored to different user characteristics.

2 Adaptivity Issues

In the adaptive hypermedia research, different techniques have been used to develop systems tailoring the interaction style and the information they present to the user's individual characteristics. A major distinction has been made between the personalization of the link level, i.e., the navigation structure, and the personalization of the content level, i.e., the information to be presented [7]. Some researchers, like [8], have focused on the dynamic adaptation of the hypertextual structure to users with different backgrounds. Others, like [16], [17], [12], [9] and [10] have focused on the dynamic generation of text tailored to the user. Some recent applications are also focused on the generation of personalized presentations exploiting life-like characters [1].

Although, as mentioned in section 1, e-commerce has strong adaptivity demands, it constraints the potentialities of adaptive hypermedia in several ways. For instance, from the viewpoint of the customer, Web stores must be accessible via standard equipments, such as a personal computer, a (usually not very fast) Internet connection and a Web browser. Moreover, the time needed to browse the catalog and find the needed products must be as short as possible, and the run-time efficiency of the system is essential.

A further constraint arises in the development of Web store shells, which must satisfy the requirements of the store designer, possibly reducing the overhead in the configuration of a new Web store. There is a trade-off between developing systems characterized by powerful capabilities in the generation of pages, but typically requiring a very detailed description of the domain-dependent knowledge, and the need to configure such systems on new sales domains with a limited effort.

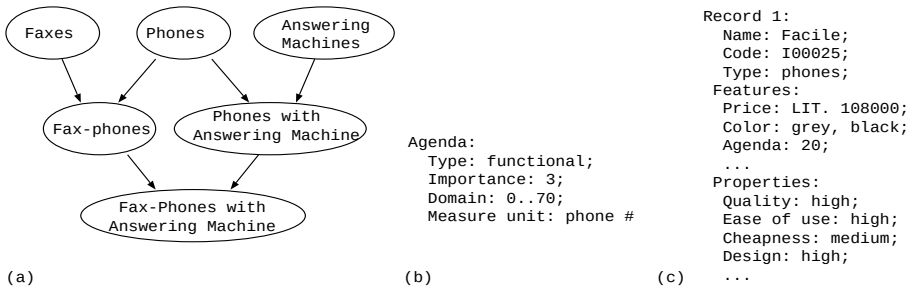

Fig. 1. (a) Portion of the Product Taxonomy. (b) Representation of a feature (c) Representation of an item.

The use of sophisticated page generation techniques, including for instance complex NLG engines, supports the development of extremely flexible systems, but can be problematic for two main reasons: first of all, these techniques may sensibly slow down such systems during the interaction with the customer. Second, the definition of very detailed domain ontologies and large linguistic resources, such as lexicons and grammars, challenges the store designers and threats the exploitation of shells in different domains.

In our work, we have focused on the selection of the type and quality of the presented information, because this aspect is essential to adapt Web catalogs to heterogeneous customers and help them to efficiently get the needed information. To this extent, a dynamic generation of the descriptions presented in the Web catalog is essential. However, to support the configurability of our system, we used template-based generation techniques, therefore avoiding the need for large lexicons, syntactic knowledge bases, and so on. Moreover, we have exploited relatively simple personalization and generation techniques, with a limited impact on aspects such as the system's accessibility, its speed during the interactions, and the overhead in the creation of new Web stores. For example, we have not exploited complex multimedia solutions (as, for instance, VR applications), or high-quality graphics that can dramatically slow down the system.

3 The Knowledge about Products

SETA retrieves the (domain-dependent) information about the product categories, their features and the relations among products from the Product Taxonomy. This is a declarative knowledge base which the system uses to reason about products and exploits to provide a rational organization of the hypertextual catalog. Moreover, the system gets the information about the items available for a product category from a Products Database (DB).

Representation of product categories. The Product Taxonomy (see Fig. 1.(a)) is a conceptual representation of products: its roots describe the general product categories (e.g., phones, faxes, etc.) and have subclasses representing

more complex products (e.g., fax-phones, fax-phones with answering machine, etc.); see [4] for details. Each node of the Product Taxonomy defines the features of a product category, specifying the range of values they can take.

This taxonomy is organized as a multiple-inheritance network to handle the description of features in an compact way. In the first version of SETA, this taxonomy was handcrafted, but we have recently developed a tool which creates it automatically, given the definition of the items in the Products DB.
The features of products are structured entities, characterized as follows:

- *Type*: the features concern different aspects of products and we have characterized them in an abstract way, by defining the following feature types:
  - *Functionalities* are basic facilities representing the purposes for which a product has been designed; e.g., phones support vocal communication, while faxes are designed to transmit documents.
  - *Technical features* concern technical details; e.g., the resolution of a fax.
  - *Functional features* include minor facilities offered by the product; e.g., the agendas offered by phones.
  - *Aesthetic features* concern aesthetic aspects, such as color and size.
  - *Generic features* include information not belonging to the previous types, such as the price.
- *Importance*: this slot specifies to which degree the feature represents a mandatory piece of information in the description of a product.[1]
- *Domain*: this slot represents the range of values that a feature can take.
- *Measure unit*: this slot characterizes the type of information which the feature values refer to.

For instance, Fig. 1.(b) shows the "agenda" which is an important functional feature and takes values in the [0..70] range: the values represent the maximum number of phone numbers which can be stored.

Representation of items. The items of a product category are described in the Products DB, which reports the values of their features and properties. As described in [4], the properties are qualitative evaluations of items and can take three scalar values: "low", "medium, "high". For instance, Fig. 1.(c) sketches the description of the "Facile" phone. This item costs Lit. 108.000 and is available in the gray or black colors; it has an agenda to store up to 20 phone numbers; it is a high-quality, very easy to use phone; it is not very expensive and has a very good design.

4 Structure of the Web Catalog

The Web catalog is a two-level hypertext and includes pages describing product categories and pages presenting the individual items of a category.

[1] The importance recalls the "salience" of features introduced in [14] and is used to describe the store designer's viewpoint on the relevance of a feature, in contrast to the possibly different customer's interests. For instance, a feature might be essential to the product description, extremely trendy, or other.

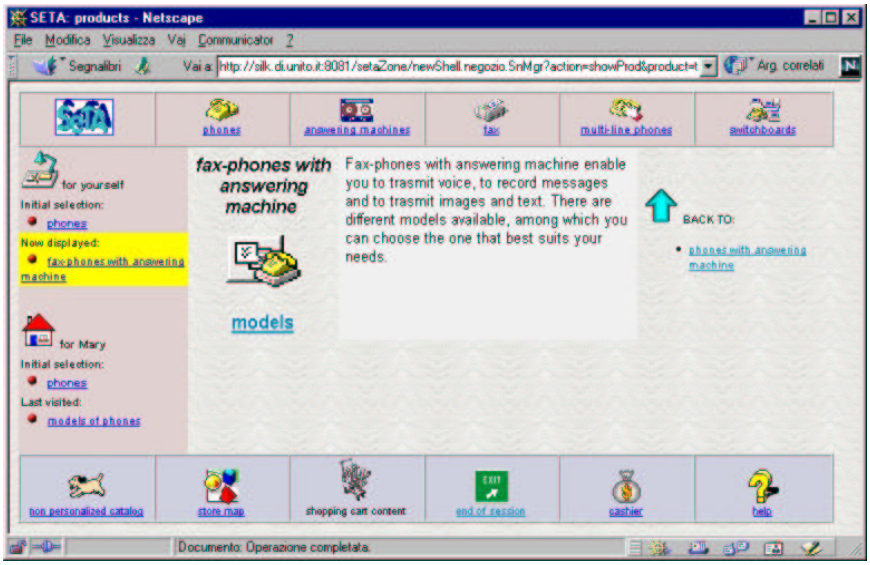

Fig. 2. A page describing the fax-phones with answering machine category.

Product pages. These pages represent the higher level of the hypertext and enable the user to browse the catalog without analyzing the individual items.

A product page is organized in areas displaying the contextual information (leftmost part of the page), the navigation buttons (upper and right parts), the control buttons (lower bar) and the description of the product category (central part). This description specifies the functionalities offered by the product items: e.g., fax-phones with answering machine enable the user to transmit voice, record messages and transmit images and text; see Fig. 2. The description is dynamically generated when the related page has to be displayed, by extracting the functionalities of the product category (defined in the Product Taxonomy as features of type "functionality") and applying a template-based NLG technique to produce the NL sentence; see section 5.

A product page also offers the links to access the pages describing other product categories related to the current one. The Product Taxonomy represents the skeleton of the hypertext: at the beginning of the interaction, the user enters the catalog by selecting some of the main product categories. Then, (s)he can visit pages presenting more complex products, by following hypertextual links corresponding to the possible paths in the taxonomy. Fig. 2 shows a page describing the "Fax-phones with answering machine" category, which the user has reached by visiting the page describing "phones" ("Initial selection" slot) and moving down to the more complex products: "phones with answering machine", "fax-phones with answering machine".

Pages describing items. Given a product page, the user can inspect the items belonging to the related category by following the "models" link, which enables

Fig. 3. A page describing a fax-phone with answering machine.

her/him to visit the pages in the lower level of the hypertext. These pages are similar to the product pages, except for the central area, which reports a detailed description of the items. Fig. 3 shows a portion of a page presenting the "Scriba Compact 401" item: the "next items" link allows the user to inspect the other available items, while the "BACK TO" link enables her/him to go back to the product page presenting the product category (higher level of the hypertext).

The description of an item specifies its properties and features. In particular, the definition of the features in the Products Taxonomy is exploited to select the information to be described, while the specific values of both properties and features are retrieved from the record of the item in the Products DB. In order to tailor these descriptions to the user, a Personalization Rule Manager sorts the features on the basis of their relevance to the user's interests and their intrinsic importance to the presentation of the item. Then, it decides how many features should be displayed in the page, on the basis of the user's receptivity. The features falling out of the limit are linked by means of the "more information" link. Finally, template-based NLG techniques are exploited to generate different linguistic descriptions, on the basis of the user's expertise level: simpler descriptions are produced for non-expert users, while more technical ones are provided to experts; see [3].

5 Dynamic Generation of Personalized Web Pages

Fig. 4 shows the architecture of the part of SETA devoted to the generation of the Web pages. The dotted boxes represent agents; the plain ones are sub-

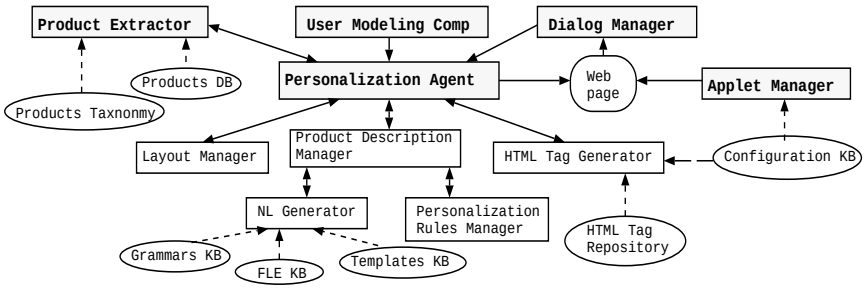

Fig. 4. Part of the SETA architecture devoted to generate the Web store pages.

modules of the Personalization Agent, which generates the Web store pages; the ovals represent knowledge bases and databases.

The Personalization Agent is invoked to produce a page by the Dialog Manager, which handles the logical interaction with the user, deciding at each step which type of page has to be produced next. When the Personalization Agent is invoked, it gets from the Product Extractor the data about products and items; moreover, it gets from the User Modeling Component (which handles dynamic user models [5]) the data about the user's characteristics; then, it generates the pages by applying its customization strategies [3].

The Web pages include standard HTML, produced by the Personalization Agent, and Java applets, provided by the Applet Manager. The Personalization Agent invokes specialized modules which apply different strategies to customize the layout and content of a page. Then, it generates the page by composing the results of their processing. In particular:

- it gets the page layout (background, fonts, etc.) from the Layout Manager;
- it gets the personalized description of the features and properties of products and items from the Product Description Manager;
- it provides these data to the HTML Tag Generator which generates the code for the requested page, by accessing a HTML Tag repository.

We focus on the activity of the Product Description Manager, which generates the NL descriptions of products and items: the selection of the information to be presented (i.e., the sorted feature list) is performed by the Personalization Rule Manager according to the rules described in [3]. Then, the generation of the linguistic description is performed by the NL Generator, which produces the sentences by exploiting the following knowledge sources:

- the Templates KB, storing the templates of the NL sentences;
- the Grammar KB, specifying the (language dependent) syntax of the expressions filling the slots of the templates: this grammar is used to select, for instance, the agreement rules for articles and the connectors;
- the Feature Linguistic Expressions (FLE) KB, mapping the internal codes of products, items, features, properties, and of their values, to linguistic expressions.

While the templates only depend on the selected language (e.g., Italian or English) and can be exploited in different domains, the other two knowledge sources depend on both the language and the sales domain; thus, they have to be configured by the store designer for the specific Web store instance.

The NL Generator exploits JTg2, a small and flexible text generation engine developed by DFKI and CELI, which, basically, produces (complex) strings receiving as input a (complex) object. This engine can be plugged in a NLG component thanks to a straightforward interface to external modules. In the following, we will not describe the engine itself,[2] but we will focus on the strategies we implemented for the generation of the product descriptions in SETA.

Generation of the descriptions of products (product pages).
The NL Generator produces these descriptions by exploiting generic templates and filling their slots with the linguistic expressions of the functionalities offered by the product categories. Such expressions are generated on the basis of the rules defined in the Grammar KB: each rule has an optional applicability test, specifying the contexts where the rule can be applied. These tests are used to represent context-dependent information: as we will see, they are essential to directly feed the NL Generator with the data to be described when the descriptions have to be generated.

For example, consider the first part of the description in Fig. 2: "Fax-phones with answering machine enable you to transmit voice, to record messages and to transmit images and text". The sentence is generated using the template:

< *product_name* > enable you to < *functionality_list* >

The NL Generator fills the first slot with the name of the product category, retrieved from the FLE KB: "Fax-phones with answering machine". For the Italian version, it also selects the appropriate article (the plural feminine "Le" or the masculine "I"), depending on the gender of the name.

Given the list of functionalities offered by the product category, the second slot of the template is filled by exploiting the grammar rules which determine the introduction of the appropriate commas and the coordinating particle "and" to link the last functionality:

funct(w:3) = ""	{*if there are no functionalities left*}
funct(w:2) = altFun	{*if we are describing the last functionality*}
funct(w:1) = altFun + cong + funct	{*if there are only two functionalities left*}
funct(w:0) = altFun + comma + funct	{*otherwise*}

In the rules, "funct" and "altFun" are non-terminal symbols, while "cong" and "comma" are terminal ones. Each rule has an associated weight, shown in brackets: rules with a higher weight are evaluated before the other ones. The applicability tests are shown in italics.

The NL expression of each individual functionality is generated by applying rules such as the following ones:

altFun(w:0) = memoF	{*if memoF is top of of functionality stack*}
altFun(w:0) = voiceF	{*if voiceF is at top of functionality stack*}

[2] For more information, see http://www.celi.it/english/tecnologia/tecLing.html.

where "memoF" and "voiceF" are non-terminal symbols, each one associated to a functionality. The NL Generator handles the list of functionalities supplied by the Product Extractor as a stack: it pops the element at the top of the stack and applies the related grammar rule, until the stack is empty. The applicability tests of these rules are used to trigger the rule associated to the top functionality and disable all the other ones. If the rule of a functionality F applies, the NL form is generated, by retrieving the linguistic expression for F from the FLE KB, and F is deleted from the stack.

Generation of the descriptions of items.
The descriptions of items include a presentation of their properties, a list of feature descriptions and an optional "more features" link. Since the properties represent a more general evaluation of the item, the paragraph presenting such properties is linked to the list of feature descriptions by means of the expression "In particular:". Currently, this is the only template expression at the discourse-level and all the rest of the generation process concerns the sentence-level.

The generation of the **property descriptions** is similar to that of the functionalities, but must include the property values. Each property is expressed by means of an adjective or a simple expression: e.g., the ("cheapness") "ease of use" is described by ("cheap") "easy to use". The ("low", "medium" or "high") property value associated to the item is added as a modifier of the adjective: for the English language, "high" is mapped to "very", "low" to "little" and "medium" to the null string.

The **feature descriptions** are generated by examining the customized feature list supplied by the Personalization Rule Manager: similar to the previous cases, the NL Generator handles the feature list as a stack, popping each feature and applying the related grammar rule to produce the NL description, until the stack is empty. Each description includes an optional template, the linguistic expression of the feature and of its values and, in some cases, the measure unit. A grammar rule is associated to each feature defined in the Product Taxonomy; e.g., the description of the "agenda" is generated by exploiting the rule:

agenda = templ + agDescr + agVal + agMeas {*if agenda is top of feature stack*}

which supports the generation of sentences like "It enables you to store up to 20 phone numbers", where "It enables you to" is the template, "store up to" is the linguistic description, "20" the value and "phone numbers" the measure unit.

Alternative rules can be introduced to support the generation of different (e.g., simple, or technical) feature descriptions, on the basis of the user's domain expertise [3]: the applicability tests of the rules will refer to the user's knowledge level. For instance, the following rules can be defined to support the generation of alternative descriptions of the "agenda" feature:

agDescr = agDescrL {*if the user expertise level is low*}
agDescr = agDescrM {*if the user expertise level is medium*}
agDescr = agDescrH {*if the user expertise level is high*}

where "agDescrL/M/H" define the feature "agenda" in natural language at different technicality levels (e.g., agDescrH = "store up to"; agDescrL = "directly

select up to"). This mechanism supports a high flexibility in the generation of sentences: e.g., alternative templates, tailored to different knowledge levels, could be defined as well, by expanding the "templ" part in different ways; for instance, "It enables you to", or "A device is offered to".

6 Evaluation of SETA

The current interface of SETA, its interaction style and its functionalities are the result of several revisions, which we have done thanks to the comments and suggestions of about 120 users involved in a subjective evaluation of our telecommunication prototype. The selected users belong to rather heterogeneous groups, including people with different age, backgrounds, job and education level.

During the experiments, we let the users free to browse the catalog, with the only requirement that, before closing the session, they had to select the item which they would have liked to purchase, out of all the items of a certain product category (specified before starting the experiment, e.g., phones or faxes). We also asked them to make comments on the type and amount of information about products provided by the system and on the quality of the linguistic descriptions.

One of the main results of this evaluation concerns the overall structure of the hypertextual catalog: in the first version of SETA, some users had problems in finding the navigation path which led them to the desired products. In order to improve the clarity of the catalog, we have structured the interface of the Web store as the two-level hypertext described in Section 4. This revision greatly improved the interaction with the users, because they could quickly search the catalog for the products offering the needed functionalities, skipping the descriptions of the items belonging to the other categories.

Another major finding concerned the description of items, which proved to be suited to users with very different levels of expertise, thanks to the adaptation of its linguistic style and to the presence of descriptions concerning both features and properties. All the users appreciated the schematic presentation of the features, based on the use of bullets, which clearly separate each feature from the other ones. Moreover, we noticed that the users' domain expertise influenced the way how they analyze the various items to select the preferred one. More specifically:

- expert users focus on the descriptions of the features offered by the presented item and rely on that information to evaluate the various items, looking at the description of the properties in rare cases;
- novice users strongly rely on the description of the properties of items and confirm that such information helps them in the selection of items.

The tests also confirmed that the use of buttons to ask for particularly technical or complex information is very useful to reduce the overhead on novice or little receptive users. In particular, these users appreciated the fact that the system focuses the presentation on the most relevant features, linking the other ones as "more information", or "technical information", so that the presented information is reduced, in their case, to the essential data.

7 Conclusions

We have described the techniques used in SETA to dynamically generate an adaptive hypertext representing a Web store. Such techniques are embedded in a shell supporting the construction of new Web stores [2]. The dynamic generation of the catalog, based on data retrieved from declarative knowledge sources (e.g., Product Taxonomy and Products DB) represents an improvement with respect to previous approaches, such as [18] and [6], where multiple versions of the catalog existed and the system selected the one to be displayed. In fact, we don't need to store redundant information about the content of pages. Moreover, the product descriptions are adapted to the user at the granularity level of their features.

The specification of the knowledge about products has a strong impact on the personalization of catalogs. For instance, a detailed representation of the domain concepts and of their features, such as the one adopted in [15] supports the exploitation of powerful techniques to describe and compare products to one another. However, it requires the definition of hierarchies for specifying not only the relations among products, but also those among the product features (e.g., to identify more general / specific features, etc.). In contrast, a flat representation of the knowledge about products, such as the one exploited in several information filtering systems [19], can be easily applied, but does not support the generation of structured presentations. In our work, we have adopted a knowledge-intensive approach to the description of the knowledge about products, in order to support effective personalization strategies. However, we have simplified the definition of the product features to enhance the configurability of the system.

In particular, the catalog structure of SETA is based on a taxonomic organization of products which provides the user with a rational navigation space. Furthermore, the classification of features in types supports the use of personalization rules, based on such types, for generating product descriptions in different domains, without requiring the introduction of complex knowledge bases. Although we defined feature types relevant to a technical sales domain, new types can be added with a modest effort. Finally, we have exploited flexible (template-based) NLG techniques, enabling the system to dynamically produce product descriptions, tailored to the individual users, in an efficient way.

Most of the applications using NLG to tailor documents to the reader require a module for dealing with discourse structure. As product descriptions used in an e-commerce interface must support a fast and efficient interaction with the user, we have not yet included a discourse planning component in the NLG module of our current prototype, e.g., like those described in [15], or in [10]. However, we think that a simple discourse planning component could improve the item descriptions to provide an explanation of the way how the item features support certain properties; for instance, in the case of phones, the ease of use relies on the availability of particular "fast access" keys. We are also improving the generation of the property descriptions themselves: in particular, we would like to filter out the properties irrelevant to the user and focus on the most important ones, on the basis of her/his preferences.

References

1. E. André and T. Rist. Presenting through performing: on the use of multiple lifelike characters in knowledge-based presentation systems. In *Proc. 2000 Int. Conf. on Intelligent User Interfaces (IUI'00)*, pages 1–8, New Orleans, Louisiana, 2000.
2. L. Ardissono, C. Barbero, A. Goy, and G. Petrone. An agent architecture for personalized web stores. In *Proc. 3rd Int. Conf. on Autonomous Agents (Agents '99)*, pages 182–189, Seattle, WA, 1999.
3. L. Ardissono and A. Goy. Tailoring the interaction with users in electronic shops. In *Proc. 7th Int. Conf. on User Modeling*, pages 35–44, Banff, Canada, 1999.
4. L. Ardissono, A. Goy, R. Meo, G. Petrone, L. Console, L. Lesmo, C. Simone, and P. Torasso. A configurable system for the construction of adaptive virtual stores. *World Wide Web*, 2(3):143–159, 1999.
5. L. Ardissono and P. Torasso. Dynamic user modeling in a web store shell. In *Proc. 14th Conf. ECAI*, to appear, Berlin, 2000.
6. C. Boyle and A.O. Encarnacion. Metadoc: An adaptive hypertext reading system. *User Modeling and User-Adapted Interaction*, 4(4):1–19, 1994.
7. P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129, 1996.
8. L. Calvi. Multifunctional (hyper)books: a cognitive perspective (on the user's side). In *Proc. workshop "Adaptive Systems and User Modeling on the World Wide Web"*, pages 23–30, Chia, Italy, 1997, http://www.cs.usask.ca/UM99/w4.shtml.
9. R. Dale, S.J. Green, M. Milosavljevic, and C. Paris. Dynamic document delivery: Generating natural language texts on demand. In *Proc. 9th Int. Conf. and Workshop on Database and Expert Systems Applications (DEXA '98)*, Vienna, 1998.
10. B.N. De Carolis. Introducing reactivity in adaptive hypertext generation. In *Proc. 13th Conf. ECAI*, Brighton, UK, 1998.
11. J. Fink, A. Kobsa, and A. Nill. Adaptable and adaptive information for all users, including disabled and elderly people. *New review of Hypermedia and Multimedia*, 4:163–188, 1998.
12. G. Hirst, C. DiMarco, E. Hovy, and K. Parsons. Authoring and generating health-education documents that are tailored to the needs of the individual patient. In *Proc. 6th Conf. on User Modeling*, pages 107–118, Chia, Italy, 1997, http://www.cs.usask.ca/UM99/w4.shtml.
13. T. Joerding. Intelligent multimedia presentations in the web: Fun without annoyance. In *Proc. of the 7th World Wide Web Conference (WWW7)*, Brisbane, Australia, 1998.
14. K.F. McCoy. Generating context-sensitive responses to object-related misconceptions. *Artificial Intelligence*, 41:157–195, 1989.
15. M. Milosavljevic. *The automatic generation of comparison in descriptions of entities*. PhD thesis, Macquarie University, 1999.
16. M. Milosavljevic, A. Tulloch, and R. Dale. Text generation in a dynamic hypertext environment. In *Proc. 19th Australasian Computer Science Conference (ACSC'96)*, pages 417–426, Melbourne, 1996.
17. A. Ndiaye and A. Jameson. Predictive role taking in dialog: global anticipation feedback based on transmutability. In *Proc. 5th Int. Conf. on User Modeling*, pages 137–144, Kailua-Kona, Hawaii, 1996.
18. H. Popp and D. Lödel. Fuzzy techniques and user modeling in sales assistants. *User Modeling and User-Adapted Interaction*, 6:349–370, 1996.
19. P. Resnick and H.R. Varian, editors. *Special Issue on Recommender Systems*, volume 40. Communications of the ACM, 1997.