An approach for automatic generation of adaptive hypermedia in education with multilingual knowledge discovery techniques

Enrique Alfonseca *, Pilar Rodríguez, Diana Pérez

Department of Computer Science, Universidad Autónoma de Madrid, 29049 Madrid, Spain

Abstract

This work describes a framework that combines techniques from Adaptive Hypermedia and Natural Language processing in order to create, in a fully automated way, on-line information systems from linear texts in electronic format, such as textbooks. The process is divided into two steps: an off-line processing step, which analyses the source text, and an on-line step, which executes when a user connects to the system with a web browser, moment at which the contents and hyperlinks are generated. The framework has been implemented as the WELKIN system, which has been used to build three adaptive on-line information sites in a quick and easy way. Some controlled experiments have been performed with real users aimed to provide positive feedback on the implementation of the system.

Keywords: Architectures for educational technology system; Authoring tools and methods; Multimedia/hypermedia systems

0360-1315/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2005.10.007
1. Introduction

The World Wide Web has made hypermedia a widely used mean for conveying information. However, the quantity of data available on the Internet grows steadily, and the large amounts of information can pose potential problems to the web surfer (Wu & de Bra, 2002): firstly, static web sites offer the same information to all kinds of users, independently of their interests. In this situation, it can be difficult for students to find relevant data with which to increase their knowledge about some topic, or they may be forced to spend some time browsing uninteresting information before finding it. Secondly, web pages can also produce comprehension problems to the students, because the author of the pages is making implicit assumptions about their previous knowledge.

The previous problems have motivated research in many different areas. On the one hand, there appeared Adaptive Hypermedia (AH) applications that try to provide the students with the information they need, and to guide them in finding their way from document to document. AH has been applied in Information Retrieval applications (Baeza-Yates & Ribeiro-Neto, 1999), that either search for or filter information inside huge repositories of documents or in the Internet, according to some user’s profile or query; adaptive hypermedia educational systems, that help the students to navigate across a course in function of the concepts that they have already learnt; or on-line information systems, that try to show the users the information that they need according to their interests and context (e.g. museum information systems).

On the other hand, there is a wholly different line of applications, thought to ameliorate the problem of the information overload, which stemmed from research on Natural Language Processing (NLP). Here, we may cite Information Extraction applications, that obtain structured information from textual documents; Question Answering systems, that look for the answer of a question written by a user in a collection of documents, or Text Summarisation applications, that condense the relevant information found in a larger textual source. In some cases, Information Retrieval (IR) applications use tools from the field of computational linguistics, such as stemmers, shallow parsers, language identification or multilingual resources. Therefore, we can also include IR in this group of applications.

This work centres on automatically building hypermedia sites that are adapted to the needs of the particular students who want to gain additional knowledge apart from that studied in a course. The following sections further specify the task addressed: firstly, Section 2 describes the motivations for attempting the work. Next, Section 3 describes the requisites with which it must comply, and the general architecture designed. Section 4 describes the different textual resources that were used for constructing some example adaptive sites, and the evaluation performed with those sites. Finally, Section 5 contains a discussion on the conclusions and contributions of this work.

2. Problem addressed

AH appeared as a mean to overcome the one size fits all paradigm, providing the framework needed to tailor web pages to the needs of the specific users. The adaptation is usually based
on a user model that encodes some of the user's characteristics, such as preferences and previous knowledge, and a device model, which stores the characteristics of the device used, such as screen size or network connection. Popular techniques for AH include adaptive navigation support, consisting in adapting the link structure of the web site so that the user is guided toward interesting information; and adaptive presentation, consisting in adapting the contents of the web pages to the user's needs, e.g. by hiding irrelevant paragraphs and highlighting the fragments that the system believes will be more relevant (Brusilovsky, 2001; de Bra, Brusilovsky, & Houben, 1999).

These techniques, while simplifying the labour of a web visitor, imply, on the other hand, a large increase in the amount of work for the hypermedia author. Now, it is not enough to write the contents of a web site and to connect the different pages together with links, but it is also necessary to define the characteristics of the user that are to be modelled, and the rules that will determine how to present the contents to different users and how to guide them through the site. One of the answers to this need has been the appearance of authoring tools for adaptive web sites (Brusilovsky, Eklund, & Schwarz, 1998; Cristea & Aroyo, 2002; Freire & Rodríguez, 2004; Murray, Shen, Piemonte, Condit, & Thibedeau, 2000; Sanrach & Grandbastien, 2000). They reduce the design work by providing a framework in which several characteristics are more or less fixed, such as the user profiles and the adaptation mechanism, but still the designer has to work hard in order to explore and take advantage of all the possibilities for adaptation.

For this situation, it would be useful to have an automatic procedure to select information from separate sources, put it all together, according the student's profile, and provide an internal structure with separate sections and hyperlinks between these sections. The research performed intends to mimic in some degree the human actions when looking for relevant information that satisfies some particular need:

1. Information can be collected from domain-specific sources, such as books and articles about the topic in which the user is interested, and which are provided by the user. Expectedly, there will be large amounts of relevant information in these sources, as they are carefully selected by the user.
2. The selected data can also be extended with information taken automatically from open general-purpose corpora, such as the World Wide Web.
3. This information can be shown to the user in a structured way, with tools that help to reduce the information selected while keeping the most relevant pieces, and which help in navigating through that information.

In order to build such a tool, it has been necessary to investigate ways in which information about the user's interest can be represented and acquired, ways in which information can be selected, from different kinds of sources, according to the user's needs, and ways in which the information, once tailored to some student's need, can be structured and organised so the student can access it.

2.1. Goals and expected results

The design of adaptive hypermedia sites can often be divided into two steps. Firstly, in an off-line step, all the information that will appear in the hypermedia site is collected and structured. It includes the following tasks:
• **Identification of the relevant topics and the future sections of the site.** If the author is already an expert, then this step might not pose much complication; otherwise, it may involve a strong interaction with domain experts. This task is common with non-adaptive hypermedia design.

• **Writing the contents of those sections.** The author has to write textual units describing the several topics. In contrast with traditional web construction, it may be necessary to write different versions of the same topics, such as equivalent texts in different languages, with different lengths, or intended for people with different cultural background.

• **Generating the Knowledge Base,** if it is necessary for the application, indicating which are the relationships that hold between the textual units. For example, some text might contain a description of something more specific than other text, and in that case it may be necessary to state that the second text has to be already visited before allowing the user to read the first one.

Secondly, there is an on-line step which functions when the visitors access the adaptive site to find information. For this step, it is necessary to perform the following work:

• **Adapting the contents** according to the user’s profile or environment. This may include showing different fragments of texts (e.g. written in different languages or with different levels of detail), changing the media (e.g. providing either a text or a image containing the same information), and several other techniques.

• **Adapting the structure of the site** in order to guide the user in the search for information. This includes hiding or showing links if they are considered irrelevant or interesting, respectively; creating new links on-the-fly; annotating the links with the kind of information to which they lead; etc.

The final goal of the work performed is a complete automation of both the off-line and the on-line steps, so that the job of the designer is reduced to a supervision of the system. Using a set of texts that is supposed to contain the contents of the future hypermedia site, the system has to identify the relevant sections, select the text included in each section, structure the different pages as a hypergraph, and provide adaptation mechanisms for different users.

There are two different results of this work. Firstly, the description of a theoretical framework that combines ideas from different areas in order to fulfil the goal of automatically generating the web sites. Secondly, the implementation of the architecture as a system which has been called **Welkin.** The system has been used to build three different adaptive web sites automatically. The evaluation has been performed both for usability (easiness of use) and for utility (usefulness) in a controlled experiment with prospective students.

### 3. Description of the system

#### 3.1. Operational requisites

The general objective of this work is the identification and presentation of relevant information to the users, addressing the need that stems from the information overload. The relevant data should be finally structured as a hypermedia site.
The input material consists of the following elements:

1. One or several texts, which are chosen by the lecturer, about a domain which is relevant to the interest of the course, such as an authoritative textbook.
2. A connection to the World Wide Web, from which additional information will be collected, if necessary.
3. A description of the student’s possible interests and goals.

The output material is a complete adaptive hypermedia web site, constructed from information found in the original documents, and in the World Wide Web. The operational requisites that this generated web site should satisfy are the following:

- It should provide simple ways for the students to describe their interests.
- It should supply relevant information.
- This information should be structured as a complete web site, with appropriate links for navigating.
- It should also provide means to adapt the information to different users, or even to the same user with different requirements, such as a higher or a lower degree of summarisation.

Concerning the domains of application, the generation of any possible kind of text from any domain is currently out of the scope of this work. Different fields of knowledge use different terminologies; the documents follow different layouts; they may include tables, figures, examples, programming code or pseudo-code, diagrams and many kinds of information each of which has to be processed with different tools. On the other hand, it is possible to define a general architecture in which different modules, able to process different kind of information, could be plugged in, and their output can be used by the hypermedia site generator in order to decide whether each piece of information is useful or not. For example, in computer science, the relevant terminology that has to be identified consists almost entirely of technical names; while in biology it consists of animals and plants, scientific names, proper names and even locations and dates. The architecture should, thus, be modular, and different modules have to communicate with each other by adding annotations either in the source documents or in dedicated databases.

3.2. Architecture: theoretical approach

The first step in designing an adaptive hypermedia site is the decision about the kind of information that it will contain. There are, quite often, linear texts, such as books, manuals or articles about the topic for which the web site will be built, and which can be used as source material. The first hypothesis that is taken is that the teacher will have them available. Next, it will be necessary to decide how the information from those linear texts will be arranged as a hypermedia site.

Furthermore, in every particular domain there is a set of words that are not used in common language (and which therefore do not appear in dictionaries), which describe domain-specific concepts. For example, in a text about history this restricted terminology includes the names of the
characters, locations, political parties and tendencies, countries, etc.; in a text about mathematics, it includes the names of theories, theorems and algebraic entities, amongst others; in a text about biology, it includes taxon and location names, biochemical compounds, etc. The second hypothesis taken is the following: a complete adaptive web site can be constructed with the information from linear text (e.g. a textbook) where all hypermedia documents are of the following three types:

- The sections already present in the source texts.
- Hypermedia pages containing information about frequent domain-specific terms.
- Index pages that list these term-specific pages in an ordered way.

For example, if we have a written report enumerating the research performed in a University department, which is structured according to some administrative guidelines, the following hyper-documents could be extracted from it:

- Web pages summarising each of the sections in the report: introduction, different kinds of research performed, funding, conclusions, etc.
- Different pages for each of the specific terms, such as scientific terms referring to fields of study, personal names of the researchers involved, locations where research stays have been performed, etc.
- Index pages, that include the index of the original report, and lists of faculty members and scientific areas.

The purpose of the modules in the off-line step is to gather, from the original sources, all the material necessary to be able to build the above-mentioned hypermedia pages.

With respect to the on-line step, given that hypermedia consists of documents and links between them, there are two kinds of methods that can be used to adapt a web site to different users. Concerning adaptive presentation, once the relevant sections have been identified, the next task consists in providing the contents to each of the sections. Index pages can be lists of links to sections describing terms that are related in some way. Secondly, the generation of the section bodies can be done in many ways. The original fragments taken from the linear text can be processed so as to select the information that most likely will be relevant for each user, by hiding fragments of information, by highlighting portions of the generated page, or by automatically generating a summary of the original text tailored to the user’s interests. For a discussion on automatic text summarisation, cf. Mani (2001). In order to generate the contents of the sections, two different sources have been used: the original domain-specific texts, from which the core of the information will be taken, and the World Wide Web, in order to extend the original information with extra data for the users who are interested in it. The adaptive navigation support includes all the techniques used to link these content pages to each other so as to facilitate the user find the information needed.

The internal structure of the proposed architecture, implemented as the Welkin system, is represented in Fig. 1. The input consists of three resources: one or several documents, a conceptual semantic network—we have used the Princeton WordNet (Miller, 1995)—and the World Wide Web (represented at the right-hand side of the figure). The following subsections describe each of the modules in more detail.
3.3. Off-line processing step

During the off-line processing, the domain-specific texts provided by the user are analysed with some linguistic tools, and then examined in order to extract the relevant terminology from them. This terminology will be used later, when the sections of the web site are planned and structured. WELKIN processes the documents with the following modules:

*Linguistic processing.* The texts provided by the user are analysed with several tools for linguistic processing, depending on the source language. They include a tokeniser, a sentence splitter, a stemmer, a part-of-speech tagger, several partial parsers for Noun Phrases and Verb Phrases, and a module that identifies text sections and chapters. Each of these modules adds annotations to the texts in the form of XML entities and attributes.

*Term extraction.* After the linguistic processing, there are several modules (with versions for English and Spanish) that locate several different entities. Some of these components are optional specialists for restricted kinds of terminology, such as dates, scientific names, or unknown domain-specific words. They use various technologies, including regular expressions and language identification technologies. The unknown domain-specific terms that have a high frequency of appearance in the documents are automatically considered relevant terms, and there will be special hypertext pages for describing them in the generated adaptive web site.

*Term classification.* After the terms have been identified, there is a module that classifies the unknown terms, automatically, inside WordNet, so some meaning can be inferred from their position in the network. The classification procedure, based on the contexts of appearance of the terms, is described in detail by Alfonseca and Manandhar (2002).

---

1 For English, the *wraetlic* tools, developed by the authors (available at [http://www.ii.uam.es/ealfon/eng/download.html](http://www.ii.uam.es/ealfon/eng/download.html)); for Spanish, the *tacat* parser (Atserias et al., 1998) combined with other tools by the authors.
The output of the off-line processing is the following:

- The original documents are returned with XML annotations, which include linguistic analyses of the texts; temporal expressions, and the relevant terms found.
- WordNet has been extended with new concepts, corresponding to the relevant terms that were identified by the Term Extraction module.
- A database is created about these documents, containing information about how to order chronologically the events told in the text; about all the places in the documents where each of the relevant terms appears; and an empty table in which the user profiles will be stored.

This modular architecture allows the author of the adaptive web site to plug in optional modules specialist in identifying some particular kind of terminology, such as dates, companies or scientific names. Every one of the described steps is completely automatic: if the specialist modules are available, it is possible for the teacher to generate the whole site just by executing a shell script, and there is no need for supervision. On the other hand, for experienced users and web site authors, the output of each processing module is written in XML files, databases and configuration files, and it is also possible to perform a manual revision of the module outputs and to correct them if a mistake (e.g. in the term classification step) is identified.

3.4. On-line processing step

The on-line processing is performed whenever a student enters the web site to read the information. Via the user profile manager, new students can register and create their profile; they can choose some of the options according to their needs, and specify their interests. Some parts of their profile will be created dynamically, using tests. After registering, it is possible to browse the generated pages. If they have time constraints, the documents will be presented to them summarised. It is also possible to ask the system to show summary pages with information from the World Wide Web, which has been automatically collected and filtered.

User profile manager. The design of the user profiles is discussed by Alfonseca and Rodríguez (2003c). When registering, new students are asked to complete a few forms in which the system constructs their profiles, with some of the following information:

1. The amount of information they are willing to read. They may indicate it in various ways, such as the total number of words that the generated web site must contain; a fixed compression rate to be performed to all the web pages; or the amount of time that they want to spend reading the whole site. In some cases, there will be no need of condensing the information, if a user has a large availability of time or asks for a very large number of words. On the contrary, in other occasions it will be necessary to reduce the contents of the site. If a user has indicated the amount of time he or she has available, it will be necessary to know his or her reading speed, measured in words per minute with a simple test.

2. Topic preferences, which can be selected from a set of pre-defined stereotypes, or generated dynamically if the students classify portions of texts as relevant or irrelevant as they browse the site.

Summary generator. When a student has time restrictions, then the adaptive hypermedia site provides the information in a condensed way, using a summarisation algorithm. However, the
user can always expand a summarised fragment in order to read it all, in case that a fragment is
found very interesting and the user does not want to miss any information about it. A sentence
extraction procedure based on genetic algorithms has been used to produce the summaries
(Alfonseca & Rodríguez, 2003b).

**User tracker.** Although the users specify, at registration time, their interests, it is possible that a
stereotype does not describe exactly the interests and the goals sought. Because of this, the user
can always return feedback to WELKIN about whether the decisions of highlighting some informa-
tion or eliminating other data were correctly taken. The aim of the user tracker component is to
update the user model of interests while the user browses the adaptive site, according to the indi-
cations received. The changes on the model take effect immediately, so the page generated right
after an update takes them into account.

**Internet retriever.** For all the domain-specific terms identified, the system performs a query on
the World Wide Web, to extend the original information about it. It has to be taken into account
that (a) most of the Information returned by search engines is probably irrelevant; and (b) it is
necessary to disambiguate the query terms in order to find accurate data. Therefore, for perform-
ing the search on the World Wide Web, context words are used to improve the quality of the
information found, as keywords, to guide the search using a standard search engine, and as filter
words, to retain only the pages in which the query terms appear surrounded by some of the words
which surround them in WordNet. The retrieved web pages, after they have been filtered with
some heuristics (Alfonseca & Rodríguez, 2003a), are put together in a single page that summarises
their information.

**User Interface.** The first purpose of the user interface is to allow new students to register into
the system, and to initialise their personal profiles. The on-line information sites support the
following functionalities:

- Concerning *adaptive content*, texts are provided summarised if there are time constraints. The
  information collected from the documents used in the off-line processing is expected to be
  highly reliable, because it was provided by the user; on the other hand, the information taken
  from the World Wide Web might not be correct. Therefore, the user will be able to identify the
  source of the different information. Finally, the user can always indicate the system whether the
  provided information is or is not interesting, in order to train it and to get better information
  afterwards.
- Concerning *adaptive navigation support*, the following links are automatically generated: links for
  navigating chronological information, to read the events chronologically ordered; links to
descriptions of semantically tagged new concepts (to read information about persons, locations
or artifacts identified in the original documents); links to read sections linearly, as they were writ-
ten in the original documents; and links to extra information collected from the World Wide Web.

Figs. 2 and 3 show two snapshots of the user interface.

### 3.5. Web sites built with the system

Although the architecture described here has been designed in a general-purpose way, for
implementation purposes some of the modules were built with a few additional restrictions.
The following are the restrictions which affect directly the kinds of texts that can be chosen to generate adaptive sites about them:

- The processed texts must be written in either English or Spanish, as all the tools for linguistic processing of the texts have been developed for these languages. Given the modularity of the architecture, it is possible to add tools for processing any other language.
- The procedures for classifying unknown terms in WordNet have been restricted to physical entities. These include animals, plants, people, artifacts, locations and bodies of water, but other kinds of terms such as abstractions or actions have not been studied yet. Therefore, the texts chosen to generate adaptive web sites with WELKIN should not contain a high amount of abstractions if the classification of unknown terms is to work.

In addition, given that the purpose of the system is to select a subset of a source text with the information that can be interesting for a particular user, for evaluation purposes, it will be applicable to documents that can be studied from many points of view. In this way, many kinds of interest profiles could be defined by the users, and they can evaluate the adequacy of the information selected by the system to their interests.
Three adaptive web sites have been developed from texts that satisfy these requirements. The first one, Darwin’s *The Voyages of the Beagle*, has been built in two versions, in English and Spanish. The other two, which have only been built for English, are Osler’s *The evolution of modern medicine*, and Hegel’s *Lectures on the history of philosophy*.

These texts are written correctly, a fact which facilitates the work of the syntactic analysers. As they deal with historical events, they contain relevant terms which are physical entities (e.g. people, locations, artifacts). Finally, they can be studied from different points of view, as they contain fragments about historical events, descriptions of the places where those events take place, or accounts of different sub-divisions of the disciplines. The generation of a whole site from each text (the off-line processing) has taken between 1 and 2 h in a Pentium III 900 MHz, so the sites can be put up-to-date with a certain frequency if the source text varies with time.

---

2 Obtained from the Gutenberg project, http://promo.net/pg/.
3 Obtained from the Gutenberg project.
4 Obtained from http://www.class.uidaho.edu/mickelsen/ToC/Hegel-Hist of Phil.htm.
3.6. Comparison to related work

We can identify two main families of systems that automatically generate hypermedia: those that translate linear text into hypertext, and those that incorporate adaptation to text or hypermedia systems.

Text to hypertext. Concerning text to hypertext conversion, Blustein (1994) distinguishes three different kinds of systems. Firstly, there are those that create a site from a highly structured text, such as dictionary definitions (Raymond & Tompa, 1988); or those that expect a set of layout annotations inside the linear text from which the structure of the generated hypermedia site can be guessed (Furua, Plaisant, & Shneiderman, 1989). With these procedures, the information that is taken into consideration consists of the sections of the documents, and the explicit references written in some markup language, such as *roff or LATEX, to figures, tables, footnotes, references and to other sections. One example is the LaTeX2HTML, a widely used command that changes a LaTeX document into a set of hypertext nodes with automatically generated hyperlinks between them, including direct guidance to the previous and the next section, and index pages. These approaches usually start by segmenting the text into smaller units. When there is no markup but the formatting guidelines of the document are known, it may be also possible to identify section titles and paragraph boundaries automatically.

Welkin has in common with these approaches that it is fully automatic; the only needed resource is the original linear texts, and a script performs all the processing with no user supervision. Welkin provides the following additional advantages:

- A linguistic processing of the texts allows the acquisition of some knowledge from them, such as the classification of high-frequency words in an ontology.
- There is no need of markup in the text, such as LATEX or SGML commands.
- The final output is generated dynamically for each kind of user. On the other hand, it is still possible to read the texts without any adaptation, by registering a user interested in everything and without compression, for whom all the information will be available, and nothing will be filtered out or summarised.

There are other methods for text segmentation that do not require markup (Hearst, 1993). These methods look for topical units, i.e., portions of the text that refer to the same topic. Welkin combines both approaches, as

- Section and paragraph boundaries are automatically found using layout clues, such as indentation, underlined sentences, sentences in uppercase and keywords (e.g. prologue, introduction or chapter).
- The topical units are identified at the topic filtering step, using a vector space model applied to the paragraphs, without the need of any markup in the text. With Welkin’s approach, furthermore, it is possible to classify a portion of the text as pertaining to several topics at the same time, something that occurs rather often in the texts, as there may be overlapping between several topics.

A second approach for hypertext generation makes use of AI techniques to transform text into hypertext. Two of the earliest systems that offer this approach are VISAR (Clitherow, Riecken, &
Muller, 1989) and TOPIC (Hahn & Reimer, 1988). The first one is used for maintaining a database of journal citations, and the second one represents the topical structure of a text as a hierarchical graph, with relationships between the portions on the text. Both use linguistic processing to some degree, and are usable only on specific domains, because they make extensive use of domain-specific knowledge bases, so they are not easily portable for different kinds of texts. Compared to these approaches, WELKIN also performs a linguistic processing of the texts and uses the lexical knowledge base WordNet, but most of the processing done is not domain-specific, so it has been successfully applied to texts on biology, history, medicine or philosophy, and it has proven able to handle unrestricted documents collected from World Wide Web.

Finally, there is a third approach that can also be considered a text-to-hypertext approach, and it is applied when there are several separate documents, and Information Retrieval techniques are used in order to group the related documents, and to add links between the ones that are related in some way. This is the approach used by WELKIN for the topic filtering, as vectors of words have been used to perform the topic filtering to calculate the adequacy of paragraphs and summary sentences to the user.

There are other works that combine the three approaches. Blustein (1999) transforms a set of journal papers into a hypertext structure identifying structural relations, linking terms with their definitions, and linking portions of texts which are similar, and proposes other kinds of links such as anaphorae. Ragetli (2001) also attempts at structuring a set of pages in a hypermedia structure, by creating a concepts hierarchy (manually) and next linking the documents to the concepts using vectorial models. In contrast, in WELKIN the internal structure of the documents is analysed, and the generated pages are obtained from portions of the source documents. Furthermore, some of the generated pages have text fragments that are not consecutive in the original text collection.

**Adaptive text to hypertext.** We may cite here as related work the systems that use text generation techniques for adaptive hypermedia generation. Oberlander, O’Donell, Mellish, and Knott (1998) and Milosavljevic, Dale, Green, Paris, and Williams (1998) describe two approaches for creating virtual museum guides and adaptive encyclopaedias, and these have the advantage that they use a complete Natural Language Generation component. Thence, the generated text is more flexible than extracting prewritten sentences, because the sentences themselves may be adapted to the user’s interests and previous knowledge.

On the other hand, using a full Natural Language Generation (NLG) approach has the problem that creating the Knowledge Base is typically very time-consuming. Oberlander et al., and Milosavljevic et al., used semiautomatic procedures for acquiring information from semi-structured data; for example, Milosavljevic et al. (1998) used a museum database where some of the information, such as the date or location of an item, was stored as separate fields in the entry; and only attempted some simple extraction techniques for the fields which contained unrestricted text. By using a summarisation technique, WELKIN loses some flexibility, but succeeds in generating complete sites from unrestricted documents.

The same applies to DiMarco, Hirst, and Hovy (1997), which describes a system for providing medical advice. The output shown to the user is produced with NLG techniques, using master documents which contain the information to be shown to users with different profiles. In order to acquire these master documents, the system uses extensively NLP tools such as a semi-automatic parser, a coreference solver and a rhetorical parser, but ultimately the documents are written by “a professional technical writer or Web-document designer”. Although full NLG allows the system to
adapt the output in a more sophisticated way to the user, our fully automated approach based on summarisation frees the web designer from the knowledge acquisition bottleneck.

4. Experiment and evaluation

The purpose of the on-line system is to be used by students who may possibly not have a background in computer science. Therefore, it is important to make sure that it is easy to use. On the other hand, it is also necessary to ascertain that the adaptive on-line web sites are really useful to the students, addressing the need of finding relevant information. For these reasons, the system has been evaluated both for usability and utility in a controlled experiment. The following three hypotheses have been tested:

**Hypothesis 1.** The system is easy to learn by novice users. This is specially needed so we know that students can get used to the system without much effort.

**Hypothesis 2.** The filtering done with respect to the user interests is useful in order to access the more relevant information at once.

The evaluation has been performed in the following way:

1. Twelve people from different backgrounds have volunteered to use the system. The sample size, although small, is similar to that used in controlled experiments for similar systems.
2. They have been taught to use the system. Next, they have filled a form indicating whether they felt at ease with it, in order to test the first hypothesis. All the questions in the forms required an answer in a scale from 1 to 5, which indicate a lower or a higher satisfaction with the system.
3. Next, half of the users have been asked to complete two tasks consisting in looking for special information inside the adaptive site, while the other half had to use a traditional text editor. This experiment was intended to measure the utility of the system, in order to discover whether users can be more productive looking for information if they use WELKIN. In this case, the utility is measured as the number of information items that each used has been able to find.

4.1. Evaluation procedure

Before the experiment, the users that participated in the controlled evaluation were asked to fill in a form in order to learn the following user characteristics: name, sex, age, previous experience using a web browser and adaptive hypermedia sites, and previous knowledge about Darwin’s and Hegel's texts. Twelve subjects volunteered for the experiments, with different backgrounds: there were one linguist, three electrical engineers, one industrial engineer, two mathematicians and five computer scientists. Seven of them were males and five of them females. They were taught how to use the system, and the profile of each person was collected. Table 1 shows their characteristics. In general, they all had much experience using a web browser (the standard deviation is very small), but only some of them knew what adaptive hypermedia is. Finally, a few of them knew about the texts used.

With respect to usability, the users were asked to play with the system until they felt confident that they understood all the functionality. In no case it took more than half an hour to experiment
with all the functions of the system. At the end, a small questionnaire was given with a few questions about usability and utility. Each question received an answer between 1 and 5, in order of increasing acceptability of the different options.

Next, in order to measure utility, the students were divided into two groups, with means as homogeneous as possible. They were also asked to perform the reading efficiency test, and special care was taken in that the two groups contained a similar distribution of this characteristic, as much as possible (the discrepancies in the standard deviation were unavoidable). Table 2 shows the distributions of the properties of the users from the two groups. Both have a mean reading efficiency of around 120 words per minute (the last row in the table).

The users were asked to complete two different tasks. Firstly, in 8 min, to collect all the names of animals found in *The Voyages of the Beagle*. Secondly, in 5 min, to collect all the biographies found in the *Lectures on the History of Philosophy*. The ones from the first group were able to use WELKIN, and could use the pre-defined stereotypes biology and biography, respectively. These profiles had been created by the teacher previously, by training the system with a few paragraphs about each topic. On the other hand, the users from the second group were given a plain text editor (emacs for the Linux users and Notepad for the Windows users) open with the text, and were told that they could use any of the functionalities of the editor.

Finally, the users were asked to answer a final question about their general satisfaction after using the system.

4.2. Evaluation results

The results for usability are shown in Table 3. In general, most of the options were welcomed by them, with a mean value between high and very high (greater than 4). A specially good result was

<table>
<thead>
<tr>
<th>Table 1</th>
<th>User features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question</td>
<td>Mean</td>
</tr>
<tr>
<td>Experience using a web browser</td>
<td>4.17</td>
</tr>
<tr>
<td>Experience using adaptive hypermedia</td>
<td>2.58</td>
</tr>
<tr>
<td>Previous knowledge about Darwin</td>
<td>2.08</td>
</tr>
<tr>
<td>Previous knowledge about Hegel</td>
<td>2.17</td>
</tr>
</tbody>
</table>

Answers from 1 to 5 (very low, low, medium, high and very high).

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Profiles of the users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question</td>
<td>Group 1</td>
</tr>
<tr>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>Experience using a web browser</td>
<td>4</td>
</tr>
<tr>
<td>Experience using adaptive hypermedia</td>
<td>2.83</td>
</tr>
<tr>
<td>Previous knowledge about Darwin</td>
<td>1.83</td>
</tr>
<tr>
<td>Previous knowledge about Hegel</td>
<td>1.83</td>
</tr>
<tr>
<td>Reading speed</td>
<td>169</td>
</tr>
<tr>
<td>Reading efficiency</td>
<td>121.64</td>
</tr>
</tbody>
</table>

Each question was evaluated from 1 to 5.
obtained by the possibility of creating the new stereotypes. Other system features that were praised are the annotation of links with colours and the possibility of expanding the summaries. The reason why the possibility of expanding the summaries received a very high score is probably linked to the fact that the coherence of the summaries received the lowest score of all the features: 3.42. One of the bigger problems of sentence extraction procedures is that the resulting summaries might not be coherent, because of pronouns and definite Noun Phrases whose antecedent is lost, dangling conjunctions, and other typical problems. This is probably the weakest point of the system.

Finally, the question of whether the navigation options make it easy to find information received the second lowest score, 3.63. Together with this question, some users provided useful comments, some of which are the following:

- The system should keep track of the pages that have already been visited by the users, and allow them to go back to them easily.
- Some users want better search procedures that allowed them to perform a search on the whole adaptive web site simultaneously. As every single page is generated on-the-fly according to the user profile, it is not a trivial task.

In addition, as some users pointed out, the speed of the system can be improved. The XML library used requires that whole files have to be read into memory and parsed in order to use a part of them, which makes it a bit slow. In recent experiments, using a machine with more memory, the files are kept in memory and the speed of the system has improved drastically.

In the utility task, the users from the first group were able to find an average of 19.17 animals, but there were big differences, ranging from the 10 to 34 animals. They also found an average of 10.6 biographies, with values ranging from 10 to 12. On the other hand, the users from the second group got a mean value of 15 animals, with individuals scores ranging from 9 to 23. Most of the users did not find any useful option of the editor, and limited themselves to reading the text and marking the animals they found. Concerning biographies, most of the users spent half of the time thinking what to do or browsing the text purposelessly, until they found by chance a biography, and next made a search through the document for the word born, which allowed them to find most

<table>
<thead>
<tr>
<th>Question</th>
<th>Mean</th>
<th>Std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easiness to register in the system</td>
<td>4.08</td>
<td>2.99</td>
</tr>
<tr>
<td>Did you find the texts interesting?</td>
<td>4.08</td>
<td>2.99</td>
</tr>
<tr>
<td>Was any stereotype similar to your interests?</td>
<td>3.91</td>
<td>2.98</td>
</tr>
<tr>
<td>Did you find useful the possibility of creating new stereotypes?</td>
<td>4.83</td>
<td>1.29</td>
</tr>
<tr>
<td>The length of the summaries is appropriate</td>
<td>4.18</td>
<td>1.91</td>
</tr>
<tr>
<td>The summaries are coherent and easy to read</td>
<td>3.42</td>
<td>2.63</td>
</tr>
<tr>
<td>The possibility of expanding the summaries is useful</td>
<td>4.67</td>
<td>2.58</td>
</tr>
<tr>
<td>The navigations options make it easy to find information</td>
<td>3.63</td>
<td>3.82</td>
</tr>
<tr>
<td>The annotations of links with colours is useful</td>
<td>4.42</td>
<td>2.22</td>
</tr>
<tr>
<td>The information collected from the WWW is useful</td>
<td>4.25</td>
<td>3.5</td>
</tr>
</tbody>
</table>

Each question was evaluated from 1 to 5.
of the biographies in the text (but not all). They found a mean of 7.6 biographies, with individual scores ranging from 6 to 9.

It can be observed that the number of results provided by each student is positively correlated to their reading efficiency: the faster a user reads, the more answers he or she finds. The mean results of the students with Welkin are all higher than the results of the other group, although they are not statistically significant with 95% confidence. It is also worth to note that the users that used Welkin had to read a smaller amount of text, as some of the time was spent by the system while generating the pages. That is also an important variable if we are to consider the fatigue of the users.

Finally, the last question, about their general satisfaction, obtained a mean value of 4.17, with a standard deviation of 1.29. Everyone answered either ‘high’ or ‘very high’.

5. Conclusions

The designed framework shows how different algorithms, components and techniques have been integrated to fulfil one single goal. Techniques borrowed from the field of Natural Language Processing have been applied to knowledge acquisition and text generation; challenges from the Semantic Web community have been addressed for the automatic collection of accurate information from the World Wide Web; and different adaptive hypermedia methods and techniques take advantage of the output of the previous modules in order to show tailored information to different users. The architecture integrates more than 15 components written in different programming languages (C, C++, flex, prolog, Java, shell script and Javascript), which interchange information via an XML encoding scheme, so that all the information generated by every module is used, in a way or other, by some other module.

A new architecture has been described for automatically transforming linear texts into adaptive hypermedia sites. Text understanding is performed in a shallow way, learning the semantics of unknown concepts by classifying them inside the lexical semantic network WordNet. By combining hypermedia generation with user modelling techniques, the text provided by the system is adapted to the user’s interests. New ways have been devised in which the interests of users can be represented, in a fine-grain way that allows the encoding of small variations in the user’s interests.

The system has been implemented and tested. It has been applied to different texts and topics, with good results. It has also been evaluated in a controlled experiment, where it was very well received by the students. The experiment suggests that productivity increases when performing certain text analysis tasks, and the students found the system usable and appealing.

Automatically constructing a hypermedia site can be done in 60–120 min in a Pentium III 900 MHz. If special stereotypes have to be defined for the site, then it may take between 1 or 2 h more because the texts have to be annotated in order to train the topic filtering module. This means that a complete hypermedia site can be obtained from a linear text quickly, and a whole collection of adaptive hypermedia sites can be available for use in just a few days. Compared to the manual work of creating a web site from an electronic text by hand, or using standard text and hypertext editors, it highly improves the productivity, apart from offering the term identification and classification results, the adaptation and summarisation facilities, and the various navigation options.
The modular architecture allows a module to be easily replaced by other, with no alteration to the remaining portions. In fact, some of the modules, such as the Term Classification component or some of the algorithms for linguistic processing, were changed as the algorithms evolved, and their replacement did not alter the overall functioning of the system. The Spanish language was added to the system with a manpower of only two week-persons (Alfonseca, Pérez, & Rodríguez, 2003). This will also make it very easy, in the future, to improve the modules independently.

References


