Cascading XSL filters for content selection in multilingual document generation

Guillermo BARRUTIETA
Mondragon Unibertsitatea
Loramendi, 4
Arrasate, Spain, 20500
gbarrutieta@eps.muni.es

Joseba ABAITUA
Universidad de Deusto
Avenida de las Universidades, 24
Bilbao, Spain, 48007
abaitua@fil.deusto.es

JosuKa DÍAZ
Universidad de Deusto
Avenida de las Universidades, 24
Bilbao, Spain, 48007
josuka@eside.deusto.es

Abstract

Content selection is a key factor of any successful document generation system. This paper shows how a content selection algorithm has been implemented using an efficient combination of XML/XSL technology and the framework of RST for discourse modeling. The system generates multilingual documents adapted to user profiles in a learning environment for the web. This CourseViewGenerator applies simplified RST schemes to the elaboration of a master document in XML from which content segments are chosen to suit the user's needs. The personalisation of the document is achieved through the application of a sequence of filtering levels of text selection based on the user aspects given as input. These cascading filters are implemented in XSL.

Introduction

It is widely accepted that content selection plays a crucial role in text generation (Reiter and Dale 2000). This process is normally seen as a goal-directed activity in which text segments are fit into the discourse structure of the text so as to convey a coherent communicative goal (Grosz and Sidner 1986). Content planning techniques, such as textual schemas (McKeown 1985) or plan operators (Moore and Paris 1993), have been successfully used as models of text generation. There are cases, though, in which these techniques may face some limitations, for example, when the structure of the discourse is difficult to anticipate (Mellish et al. 1998). Nevertheless, when a set of well-defined communicative goals exists, complex goals can be broken down into sequences of utterances and generation becomes an efficient "top-down" process (Marcu 1997).

This paper shows a macro level content selection algorithm that applies user profiles to constrain and discriminate the contents of a text, whose discourse structure is represented using a simplified version of Rhetorical Structure Theory (Mann and Thompson 1988). The algorithm has been implemented using XML/XSL-based technology in a multilingual document generation system for educational purposes. The main objective of this CourseViewGenerator system (Barrutieta, 2001 and Barrutieta et al., 2001) is to automatically produce multilingual learning documents that suit the student's needs at each particular stage of the learning process. Figure 1 shows the overall architecture of the system.

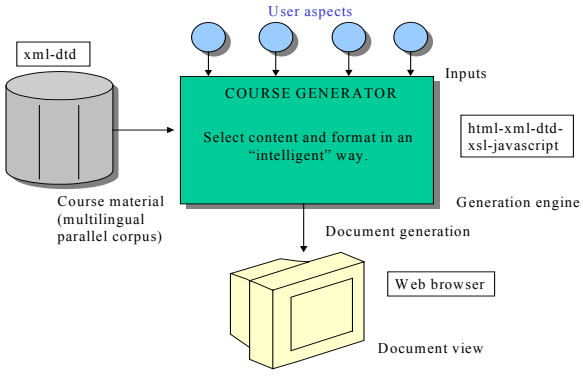

Figure 1: General scheme of the multilingual document generation system

We will begin by explaining the different parts of the system before addressing in more detail the content selection algorithm itself. The system starts by constructing a master document of the kind Hirst et al. (1997) proposed. This master document consists in a full-fledged text with references to all necessary multimedia elements (figures, tables, pictures, links, etc.). In our case, this master document takes the shape of a simple text file with all relevant information tagged in XML. Tags carry information of the logical composition of the text as well as metadata information about

its discourse structure. The text is seen as raw data, and tags encapsulate these raw data as metadata. The structure of the discourse is represented using a simplified version of RST. RST is simplified in the sense that the granularity of discourse segments does not transcend the boundaries of the sentence.

Table 1. illustrates this *gross-grained* version of RST in which discourse relations are represented as XML tags.

```
<RST>
 <RST-S>
  <PREPARATION>
   <S> What is knowledge management? </S>
  </PREPARATION>
 </RST-S>
 <RST-N>
  <S>
   Knowledge, in a business context, is the organizational
   memory, which people know collectively and individually
  </S>
  <S>
   Management is the judicious use of means to accomplish
   an end
  </S>
  <S>
   Knowledge management is the combination of those
    concepts, KM = knowledge + management
  </S>
 </RST-N>
</RST>
```

```
<RST>
 <RST-S>
  <PREPARATION>
   <S> ¿Qué es gestión del conocimiento? </S>
  </PREPARATION>
 </RST-S>
 <RST-N>
  <S>
   Conocimiento, en el contexto de los negocios, es la memoria
   de la organización, lo que la gente sabe colectiva
   e individualmente
  </S>
  <S>
   Gestión es el uso juicioso de recursos para alcanzar un fin
  </S>
  <S>
   Gestión del conocimiento es la combinación de esos dos
   conceptos, GC = gestión + conocimiento
  </S>
 </RST-N>
</RST>
```

```
<RST>
 <RST-S>
  <PREPARATION>
   <S> Zer da ezagutzaren kudeaketa? </S>
  </PREPARATION>
 </RST-S>
 <RST-N>
  <S>
   Kudeaketa, negozioetan, erakundearen memoria
   da, jendeak bakarka eta taldeka dakiena
  </S>
  <S>
   Kudeaketak erabideen erabilera zuzena du helburu
  </S>
  <S>
   Ezagutzaren kudeaketa bi kontzeptu hauen nahasketa da,
   EK = ezagutza + kudeaketa
  </S>
 </RST-N>
</RST>
```

Table 1: Gross-grained RST in XML

As any other standard RST discourse tree, this simplified RST contains a nucleus for each text paragraph, and one or several satellites linked by a discourse relation to the nucleus within the same paragraph. The nucleus is an absolutely essential segment of the text, as it carries the main message that the author wants to convey. Satellites can be replaced or erased without changing the overall message and play an important supporting role for the nucleus.

In our system, satellites are selected or discarded depending on the reader's profile. The reader's profile is defined through a set of *user aspects*. These take the form of multi-value parameters that were sketched after a number of surveys were conducted among teachers, students and other experts from the educational context. As a result of these surveys a user model was proposed (Barrutieta et al, 2002). Table 2 illustrates a simplified version of the model.

Specific User Aspects	**Discrete values**
Subject	Language processors
Moment in time	Before the course / Period 1 / Period 2 / … / After the course (review)
Languages	EN/ ES/ EU
General User Aspects	**Discrete values**
Level of expertise	Null / Basic / Medium / High
Reason to read	To get an idea / To get deep into it
Background	Not related to the subject / Related to the subject
Opinion or motivation	Against / Without an opinion or motivation / In favour
Time available	A little bit of time / Quite some time / Enough time

Table 2: User model

Based on this user model, we will now discuss the **content selection algorithm** (henceforth CSA). The CSA determines which segments of the discourse are going to be used in order to make explicit the set of parameters that conform with the user's profile. In principle, nuclei will always be chosen (as they convey the main message of the text); satellites, however, will be selected depending on their relation to the nucleus and the user aspects that are activated at the time of generation.

The selection algorithm works in three consecutive phases: parallel selection, horizontal filtering and vertical filtering. Vertical filtering is the most important phase of the three as it is here that the parts of the discourse tree are selected or discarded.

1 CSA - Parallel selection - Phase 1

In the phase of parallel selection two of the three specific user aspects are taken into account: subject and languages. These aspects identify the relevant XML master document in the chosen language (as illustrated in figure 2.). There is one master document for each subject covered by the system, and these documents contain parallel aligned versions of the texts in each language (English, Spanish and Basque, in our case).

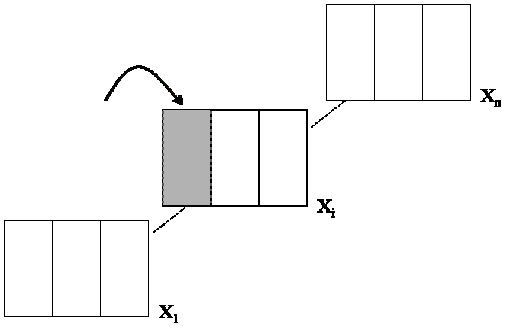

Figure 2: CSA – Parallel selection

As a result of this first filtering phase, the appropriate language division of the master document is selected. This text division is the input for subsequent filtering phases in which the particular segments of the document will be discriminated.

2 CSA - Horizontal filtering - Phase 2

The horizontal filtering phase concerns the third remaining user aspect that is moment in time, which is used to suit the generated text to the particular moment of the learning plan. This aspect cuts horizontally the parallel selection of the previous section.

The master document is structured in accordance with a set of course scheduling parameters. Each day and learning unit within the day is correlated with corresponding set of learning entities in the XML master document. In this way, the generated document can be targeted for learning unit 1 of day 1, or any other day or unit. The XML master file also contains some informative elements that the reader may need to know even before the course starts or after it has finished. These will be generated also as a result of some specific user aspects that are activated. Figure 3 shows a graphical representation of horizontal filtering.

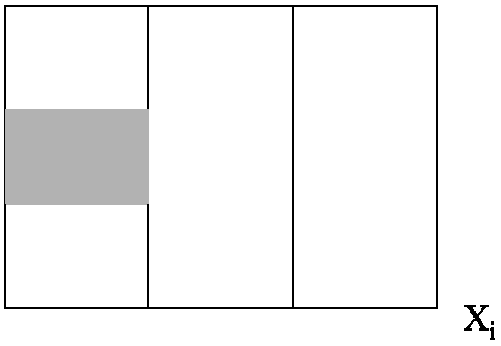

Figure 3: CSA – Horizontal filtering

3 CSA - Vertical filtering - Phase 3

The final phase of vertical filtering comprises the five user aspects of level expertise, reason to read, professional background, opinion or motivation and time available. These five aspects will be relevant to discriminate those parts of the discourse tree which have been previously selected and filtered.

Nuclei will be always maintained because they are, by definition, irreplaceable segments of the text and convey the main message. Satellites are segments of the text that will be subject to the algorithm's process of selection. Figure 4. shows graphically this filtering phase.

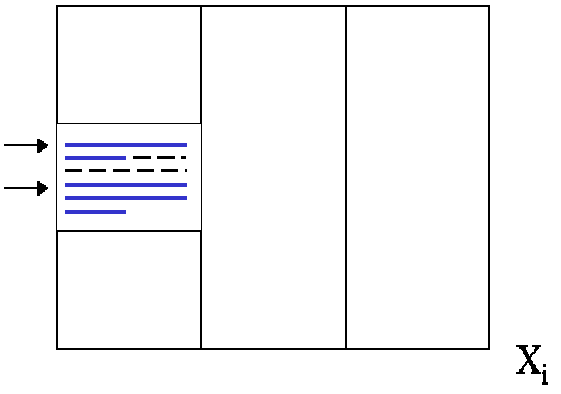

Figure 4: Vertical filtering

The set of discrimination rules applied in this first version of the content selection algorithm is described below. These rules apply in subsequent checking levels of filtering, and therefore have a cascading effect. It is known that RST covers an indefinite number of relation-satellites (Knott, 1995) which have been classified by Hovy & Maier (1997), but we will only mention the set of relation-satellites used in the master document taken as example.

3.1 Vertical filter – Level of expertise

If level_expertise = "null" or level_expertise = "basic" Then no relation-satellite is discarded;

If level_expertise = "medium" or level_expertise = "high" Then discard example, exercise, background and preparation relation-satellites;

Rationale for the rule: Any user with a null or basic level of expertise on the selected subject will need all the information available to understand the text. Alternatively, a user with a medium or high level of expertise will not require examples, exercises, background, preparation and similar relation-satellites.

3.2 Vertical filter – Reason to read

If reason_to_read = "to get an idea" Then discard exercise and elaboration (all the types of elaboration: textual elaboration, link elaboration and image elaboration) relation-satellites;

If reason_to_read = "to get deep into it"Then no relation-satellite is discarded;

Rationale: Any user wishing to broaden his knowledge in the selected subject will need additional information. Conversely, a user with the intention of just getting an idea does not need any exercise, elaboration, or similar relation-satellites, which often require a more active role on the part of the user.

3.3 Vertical filter – Professional background

If job_studies = "not related subject" Then no relation-satellite is discarded;

If job_studies = "related subject" Then discard background and preparation relation-satellites;

Rationale: Any user whose professional background is not related to the subject will need all the additional supporting text to understand its meaning. Conversely, if the user is related to the selected subject, we may assume that background, preparation and similar relation-satellites will be unnecessary.

3.4 Vertical filter – Opinion or motivation

If opinion_motivation = "against" or opinion_motivation = "without an opinion or motivation" Then no relation-satellite is discarded;

If opinion_motivation = "in favour" Then discard motivate, antithesis, concession and justify relation-satellite;

Rationale: A motivated or favourable user will not require additional motivation and, therefore, the motivate, antithesis, concession, justification, and similar relation-satellites will be disregarded, since they play a role in changing the opinion of the user to be in favour of the course material.

3.5 Vertical filter – Time available

If time_available = "a little bit of time" Then discard all the relation-satellites;

If time_available = "quite some time" Then discard exercise relation-satellite;

If time_available = "enough time" Then no relation-satellite is discarded;

Rationale: Time availability is a crucial user aspect. If the user is in a rush or has little time, the system has to provide only the most elementary information. In such case only nuclei will be generated. If the user has a bit more time, but not much, exercises are not offered, since they are usually quite time consuming and they require an active participation of the user. Finally, if the user has plenty of time, all the additional information is delivered.

3.6 Final comments on vertical filters

Cascading filters apply to the relation-satellites that are still active after the previous phases in the generation process. When a vertical filter 3 tries to get rid of a relation-satellite already abandoned at a previous phase (2 or 1), there will be nothing to act upon, but

this circumstance will produce no consequence, since the CSA continues the filtering process on the remaining text. Thus, the order in which the vertical filters are applied is not relevant.

After the filtering process has been successfully completed, there is still a final presentation task. A good presentation is, in our opinion, one that will provide the student with an optimal version of the document to read, understand and fruitfully assimilate its content.

4 Implementation

The javascript code manages the user aspects (one of the inputs of the algorithm) and the application of the casdading filters (the CSA).

Depending on the user aspects given by the user, the variables sXSL1 to sXSL5 take the value of the filter to be applied for each user aspect (level of expertise, reason to read, background, opinion or motivation and time available).

The sResult variable contains the XML file whose content will be varying after each filter is applied. Table 3 shows the code that executes a filter.

```
objData.loadXML(sResult);
objStyle.load(sXSL1);
sResult=objData.transformNode(objStyle);
```

Table 3: Javascript implementation

XSL filters pass on (or not) one element to the following vertical filter depending on the rules described before. Table 4 shows how this is done with the relation-satellite BACKGROUND.

```
<xsl:template
      match="BACKGROUND">
  <xsl:copy>
    <xsl:apply-templates/>
  </xsl:copy>
</xsl:template>
```

Table 4: XSL implementation

5 Experimentation

The objective of the experiment is to validate the hypothesis expressed in the filtering rules and the actual filtering mechanism of the CSA.

Several ideas are taken into consideration in this respect, but we are aware that users (students, professor and other scholars) are the final judges. Their assessment of the system will depend on whether the generated document meet (or fail to do so) their information requirements, providing them with just the right type and amount of information.

Conclusions

In the tests conducted so far, the *CourseViewGenerator* is functioning correctly. One of the features that is worth considering is the scalability of the filtering mechanism. We anticipate two types of expansions to the system: (1) Increasing the size of the corpus, including more subjects and master documents, and (2) augmenting the user model by adding user aspects or by adding more parameters to the existing user aspects.

The first type of expansion will not require any alteration of the CSA as long as the added document tokens conform to the existing DTD and our RST model. In order to increase the size of the corpus, it will be necessary to annotate XML discourse-tree metadata manually. This is a complex and time-consuming task (as has been noted by Carlson and Marcu, 2001). Future research activities should focus on helping automate the annotation process, for example using cue phrases à la Knott (Knott 1995; Alonso and Castellón, 2001).

The second type of expansion requires only the elaboration of additional XSL filters. Adding new values to existing user aspects requires only the modification of the corresponding XSL filter. Any of these last two operations can be incorporated easily. Therefore, adding a new user aspect or a new discrete value does not increase in any substantial way the complexity of the system.

Acknowledgements

This research was partly supported by the Basque Government (*XML-Bi: multilingual document flow management procedures using XML/TEI-P3*, PI1999-72 project).

References

Alonso, L. and Castellón, I. (2001) Towards a delimitation of discursive segment for Natural Language Processing applications. First International Workshop on Semantics, Pragmatics and Rhetoric. Donostia (Spain), pp. 45-52.

Barrutieta, G. (2001) Generador inteligente de documentos de formación. Virtual Educa 2001, Madrid (Spain), pp. 256-261.

Barrutieta, G., Abaitua, J. and Díaz, J. (2001) Gross-grained RST through XML metadata for multilingual document generation. MT Summit VIII. Santiago de Compostela (Spain), pp. 39-42.

Barrutieta, G., Abaitua, J. and Díaz, J. (2002) User modelling and content selection for multilingual document generation. Unpublished but currently been evaluated for publication.

Carlson, L.and Marcu, D. (2001) Discourse tagging manual. Technical report ISI-TR-545. ISI Marina del Rey (USA).

Grosz, B. and Sidner, C. (1986) Attention, intentions and the structure of discourse", Computational Linguistics, 12:175-204.

Hirst, G., DiMarco, C., Hovy E. & Parsons K. (1997) Authoring and Generating Heath-Education Documents That Are Tailored to the Needs of the Individual Patient. Proceedings of the Sixth International Conference. UM97. Vienna (NY-USA), pp. 107-118.

Hovy, E. & Maier, E. (1997) Parsimonious or profligate: how many and which discourse structure relations? <http://citeseer.nj.nec.com/hovy97parsimonious.html>

Knott, A. (1995) A Data-Driven Methodology for Motivating a Set of Coherence Relations, Ph.D. thesis, University of Edinburgh, Edinburgh (UK).

Mann, W.C., and Thompson, S.A. (1988) Rhetorical Structure Theory: A theory of text organization. Tech. Rep. RS-87-190. Information Sciences Institute. Los Angeles, CA.

Marcu, D. (1997) From local to global coherence: a bottom-up approach to text planning, in Proceedings of AAAI-97, American Association for Artificial Intelligence, pp.629-635.

McKeown, K. (1985) Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text, Cambridge University Press.

Mellish, C., M. O'Donnell, J. Oberlander and A. Knott (1998) An architecture for opportunistic text generation. Proceedings of the Ninth International Workshop on Natural Language Generation, Niagara-on-the-Lake, Ontario, Canada, pp. 28-37.

Moore, J. and Paris, C. (1993) Planning texts for advisory dialogues: capturing intentional and rhetorical information, Computational Linguistics, 19.

Reiter, E. and Dale, R. (2000) Building applied natural language generation systems. Cambridge University Press (UK).