TRANS · NÚM. 11 · 2007
ARTÍCULOS · 259-276

Current approaches to multilingual document management employ human translation, machine translation (MT) or computer-assisted translation (CAT) systems to produce versions of a single document in several languages. However, recent advances in natural language generation (NLG) technology suggest that it is possible to implement language-independent systems to produce source language-unbiased multilingual documents in a more efficient and cost-reducing way.

In this paper we introduce GenTur –an authoring tool for producing multilingual tourism contracts. Special attention will be paid to two basic elements of its implementation: on the one hand, the XGTLing interlingua for the discursive representation of contracts, and, on the other hand, the development of a system architecture that enables the aforementioned interlingua to generate tourism contracts by means of the GT–Mth generation algorythm.

KEYWORDS: Authoring tool, natural language generation, GenTur, XGTLing

A New Multilingual Authoring Tool of Semistructured Legal Documents

Los enfoques actuales de gestión de la documentación multilingüe hacen uso de la traducción humana, la traducción automática (TA) y la traducción asistida por ordenador (TAO) para producir versiones de un solo documento en varios idiomas. Sin embargo, los recientes avances en generación de lenguaje natural (GLN) indican que es posible implementar sistemas independientes del lenguaje a fin de producir documentos en varios idiomas, independientes de una lengua origen, de forma más eficiente y rentable.

En este artículo presentamos GenTur —una herramienta de ayuda a la redacción para producir contratos turísticos en varios idiomas. Se prestará especial atención a dos elementos básicos de su implementación: por un lado, la interlengua XGTLing usada para la representación discursiva de los contratos, y por otro lado, el desarrollo de una arquitectura que permita a la citada interlengua generar contratos turísticos por medio del algoritmo de generación GT-Mth.

PALABRAS CLAVE: Herramienta de ayuda a la redacción, generación de lenguaje natural, GenTur, XGTLing

José Luis Caro Herrero, Gloria Corpas Pastor, Antonio Jesús Trujillo Pérez, M.ª Rosario Bautista Zambrana, A. Guevara Plaza, A. Aguayo Maldonado
Universidad de Málaga

1. INTRODUCTION

Nowadays there is no doubt about the role that multilingual text generation plays in an ever-changing world where international relationships keep growing at a fast pace. In an era of globalisation, the world of business does not remain foreign to the current situation, as there is a pressing need to translate and/or produce documents in several languages simultaneously.

In the tourism industry, international agreements are daily signed between tourism companies (eg. hotel chains, transport companies, etc.) or between individual parties and tourism companies. This requires enormous translation efforts to ensure the appropriate comprehension of contracts, so that (1) the parties involved fully understand the contractual relationship they enter into, and (2) that all translated legal documents are fully comprehendible and conformant to the different national regulations involved.

Machine translation (MT) systems capable of performing the aforementioned tasks have existed from the very beginning of computer science. Nowadays Machine Translation can be considered a conglomerate of several research fields, where computing and linguistics play a leading role, among other auxiliary disciplines.

There are many R&D projects and joint research efforts within this multidisciplinary field. Many of these research activities are carried out under the auspices of official institutions, such as the European Union, where loads of documents have to be translated or produced into the several official languages everyday. Turicor (BFF2003-04616) is an information and communication technologies (ICT) R&D project applied to Translation, along the lines of the EU pluriannual programmes Human Language Technologies (1998-2002), e-Content (2001-2004) or e-Contentplus (2005-2008)[1]. The Turicor project aims at developing a prototype authoring tool for producing tourism contracts in four languages (Spanish, Italian, English and German) (Aguayo et al., 2004). This text generator is based on a language-independent content representation (interlingua) and an automatic generation module developed within the framework of a former R&D project on sales contracts within the domain of residential tourism (Ref. no. PB98-1399, Spanish Ministry of Education and Science).

The present paper[2] sets out to describe GenTur, a prototype authoring tool for tourism contracts, and it is divided into several parts. Section 2 offers a brief overview of NLG technology and multilingual generation. Sections 3, 4, and 5 contain the core of the paper and describe the GenTur system and XGTLing, the interlingua used for formal content representation of contracts. Section 6 offers the conclusion.

2. NATURAL LANGUAGE GENERATION

Many authoring tools are based on the technology of Natural Language Generation (NLG). NLG is a relatively young discipline, although it is somehow one of the most representative sub-fields of Natural Language

[1] For further information, refer to the project official website (http://www.turicor.net).

[2] The research reported in this paper has been carried out in the framework of two projects: *Turicor: A multilingual corpus of tourism contracts (German, Spanish, English, Italian) for automatic text generation and legal translation* [Funding source: Spanish Ministry of Science and Technology (National R&D Program) and European Union (European Regional Development Fund-ERDF). Ref. no. BBF2003-04616] and R&D Project for Excelence *La contratación turística electrónica multilingüe como mediación intercultural: aspectos legales, traductológicos y terminológicos* [Funding source: Andalusian Ministry of Education, Science and Technology. Ref. no. HUM-892 (2006-2009)].

Processing (NLP). Most work carried out in the 60's revolved around machine translation, whereas the 70's marked the beginning of research on NLG. As Vander Linden (2000) has clearly stated, «the goal of NLG process can be viewed as the inverse of that of natural language understanding (NLU) in that NLG maps from meaning to text, while NLU maps from text to meaning». In the following sections we will give a short account of NLG systems, which will allow us to place it in reference to current research lines.

2.1. Basics of NLG

Many authors have proposed definitions of Natural Language Generation. Some of the most relevant are the ones formulated by Hovy (1996) and Reiter and Dale (2000: 1). The former states the following: «The area of study called natural language generation (NLG) investigates how computer programs can be made to produce high-quality natural language text from computer-internal representations of information». For Reiter and Dale (2000: 1), NLG systems are computer software systems that start from some type of non-linguistic representation of information as input and rely on knowledge about language and the application domain to automatically produce reports, documents, explanations and other kind of texts.

In short, NLG consists in the production of natural language texts from an abstract semantic knowledge representation, called interlingua. In this way, a NLG system takes into account abstract information (generally non-linguistic, unequivocal and well-structured information), as well as data about the communicative situation, in order to produce a text with a coherent structure and appropriate linguistic expressions. In the case of multilingual generation, the system is able to produce texts in several languages from the same abstract source, either in a simultaneous or sequential way.

As pointed out above, the input provided to a NLG system is generally of a non-linguistic nature (Dale, Di Eugenio and Scott, 1998: 347): thus, it can be symbolic (e. g., taken from an expert system knowledge base) or numeric (e. g., taken from a database containing stock market prices). However, there are some systems, such as CourseViewGenerator (Barrutieta, 2001), which do rely on the use of linguistic input.

The applications of NLG are varied: SIGGEN[3], the Special Interest Group in Natural Language Generation, mentions report generation, machine translation, and explanations for knowledge-based systems; Bateman and Zock (2003: 286) add text summarization and multilingual and multimodal presentation of information. Likewise, Langkilde (2002: 1) states that NLG is a subtask of many applications:

> Such applications include machine translation, human-computer dialogue, summarization, report creation, automatic technical documentation, proof/decision explanation, customized instructions, item and event descriptions, question answering, tutorials, stories, and more.

NLG methods or *processes* can be classified according to their sophistication and expressive power. Eduard Hovy (1996) distinguishes four generation methods: canned text systems, template systems, phrase-based systems, and feature-based systems.

1.— Canned text systems are used in the

[3] We can find more information at the URL http://www.siggen.org/. SIGGEN is the most important interest group devoted to the study of NLG. [Last visited: 1-7-2007]

majority of software applications and consist in the presentation of strings of words without any change (error messages, warnings, letters, etc.). The approach can be used equally easily for single-sentence and for multi-sentence text generation, but it proves insufficient when we need to adapt the text to different situations.

2.— Template systems represent the next level of sophistication and are used to produce similar messages, in which a few open fields are filled in specified constrained ways. The template approach is used mainly for multisentence generation, particularly in applications whose texts are fairly regular in structure such as form letters and some business reports.

3.— Phrase-based systems employ generalized templates, whether at the sentence level or at the discourse level. Hovy explains that in such systems, a phrasal pattern is first selected to match the top level of the input (say, [SUBJECT VERB OBJECT]), and then each part of the pattern is expanded into a more specific phrasal pattern that matches some subportion of the input (say, [DETERMINER ADJECTIVES HEAD-NOUN MODIFIERS]), and so on; the cascading process stops when every phrasal pattern has been replaced by one or more words. Phrase-based systems can be powerful and robust, but are very hard to build beyond a certain size; that is why this method is used mainly for single-sentence generation.

4.— Feature-based systems are among the most sophisticated generators. According to Hovy, in feature-based systems, each possible minimal alternative of expression is represented by a single feature; for example, a sentence is either positive or negative, it is a question or an imperative or a statement, its tense is present or past and so on. In this way, each sentence is specified by a unique set of features. The researcher continues to explain that generation proceeds by the incremental collection of features appropriate for each portion of the input (either by the traversal of a feature selection network or by unification), until the sentence is fully determined. This type of system has strengths and weaknesses: «Their strength lies in the simplicity of their conception: any distinction in language can be added to the system as a feature. Their weakness lies in the difficulty of maintaining feature interrelationships and in the control of feature selection (the more features available, the more complex the input must be)» (Hovy, 1996).

The process performed by a NLG system generally comprises three phases —document planning, microplanning and surface realisation, which correspond to three modules:

1.— Document planner, which determines the content and the structure of the document.

2.— Microplanner, which decides which words and syntactic structures will be used in order to communicate the content and the structure chosen by the document planner.

3.— Surface realiser, which maps the abstract representations of the microplanner into real text.

NLG systems are designed around an architecture based on the implementation of the former generation phases. Although efforts have been made to determine a standard conceptual architecture (Cahill et al., 2000), in this paper we present a classification based in the way generation phases are carried out[4]:

[4] For more information on NLG architectures, see Reiter and Dale (2000), Vander Linden (2000), Cahill et al. (2000) and Nirenburg, Lesser and Nyberg (1989).

1.— Pipelined architecture. This kind of architecture is based in two distinct serial phases, i.e. the discourse planning is designed first and then the surface realisation is carried out. These are the simplest kinds of architectures. They do not allow backtracking nor feedback. Some examples are MUMBLE, TEXT, Naos, Wisber, IPG, PARRY, ERMA and PROTEUS.

2.— Interleaved architecture[5]. It was designed to fill the lack of refeeding in pipelined architectures. Thus, it provides actual refeeding communication between the discourse planning and the surface realisation modules, as well as backtracking and decision-taking during the message building phase (target text production). Some examples are PAULINE, SPUD, INDIGEN and POPEL.

3.— Integrated architecture. This architecture consists in identifying an orthogonal set of independent processes that can be used on demand during text production. Representative samples are GLINDA, Oz KAMP, PICARD and DIOGENES.

2.2. *Authoring tools for multilingual text generation*

Multilingual text generation stands out as one of the most innovative fields within Translation Technologies (TT). Recently, John Hutchins, one of the great precursors of TT and machine translation, considered the incipient research on NLG to be one of the most promising fields in the area of TT and pointed out that it would be eventually integrated into MT systems:

> Users will want seamless integration of information retrieval, extraction and summarization systems with translation. Research has begun in such areas as cross-lingual information retrieval, multilingual summarization, multilingual text generation from databases, and so forth and, before many years, there may well be systems available on the market and the Internet. (Hutchins, 2005: 3)

Other authors, such as Paris et al. (1995), consider multilingual generation as an alternative to (machine) translation, due to the several advantages it offers:

> Multilingual generation is also more appealing than monolingual generation followed by translation because (1) the texts can be generated in several languages simultaneously rather than waiting for the translation process, (2) the underlying knowledge being expressed in monolingual instructions can be used to generate instructions in different languages, and (3) generating directly from the underlying knowledge base can produce more natural texts as the output text is not constrained by a source text. (Paris et al., 1995: 1)

Thus, the strength of multilingual text generation lies in its capacity to generate documents in several languages from a sole source of information: the conceptual choices made by the user. As Hartley and Paris (1997: 110) point out, «This technology shifts the attention to an even earlier stage in the authoring process, that of specifying the semantics of the text to be produced, also called the 'message'». In this way, these systems, called *authoring tools* or *drafting tools*, allow the users to manually specify the conceptual content of the document and then obtain the linguistic realisation in the selected languages. Two important benefits arise: first, additional generators for other languages can be added, so a single authoring process supports multilingual variants of a document directly: one update to the document is reflected in all languages simultaneously; secondly, each generator can be adapted to its own language and cultural settings, choosing its own most

5 They are also named interactive systems.

appropriate realisation strategy independently of the others (cf. Scott and Evans, 1998). Some of these systems are DRAFTER (Paris et al., 1995), AGILE (Hartley et al., 2001) and PILLS (Scott et al., 2001). DRAFTER is a software-manual drafting tool for English and French. AGILE produces technical instruction texts in Bulgarian, Czech and Russian and generates several types of texts, common for software manuals, in two styles. PILLS, for its part, is an authoring tool designed to allow (monolingual) technical authors to generate patient information leaflets in multiple languages.

In order to store the concepts of the text(s) to be produced, authoring tools rely on semantic knowledge bases.[6] They are, as Hartley and Paris (1997: 118) explain, «the system's central repository for all the information that might potentially be expressed in the texts, regardless of the language or style used» and contain a collection of entities —actions, states, objects, and the relations between them— representing the information commonly occurring in the given domain (e.g. software, tourism, automobiles). This knowledge is generally derived from multilingual corpora of domain texts and is language-independent, so it constitutes the foundations of the interlingua underlying the authoring system: it provides a conceptual representation of the meaning or content of the document in terms of a set of primitive concepts from which the lexical units of different languages can be constructed[7].

The next steps in the production of texts depend on the specific authoring tools, but they coincide to a great extent with the phases set out before. First, the concepts of the generated texts are drawn from the knowledge base and fed to the document planner, which determines the structure of the text to be produced; then, a microplanner «maps domain concepts and relations into content words and grammatical relations» (Fiedler, 2005), that is, it chooses the right linguistic expressions to be conveyed. Finally, a surface generator, one for each of the natural languages, performs the realisation of the sentences.

DRAFTER (Paris et al., 1995; Hartley and Paris, 1997) is a good example of drafting tool. It generates instructions on how to use software applications, in English and French, and comprises three modules: a *Domain Knowledge Base*, the repository of information about the software domain; an interface for the technical writer (*the Developer's Tool*), which allows technical communicators to specify formally the procedures necessary for the end-users to achieve their goals when using the software application under consideration, and also allows them to control the drafting process; and the drafting tool (*Automated Drafter*), which comprises two major components: the text planner and the tactical generator. The result is English and French drafts of the instructions for the procedures defined so far by the technical communicator using the interface. These

[6] Knowledge bases use ontologies to structure the knowledge of a given domain. An ontology, according to Gruber (1993), is a specification of a conceptualization; that is, it is a concept hierarchy of a certain field, which specifies the relationships existing between the concepts and the properties that they have. Its applications are well-known and have meant major breakthroughs, for example, in the field of Terminology (WordNet, MikroKosmos) and the Semantic Web. On the other hand, as Bernardos (2003: 232) points out, the use of ontologies in NLG is more than justified, because they allow the reutilization of information resources, for both multilingual generation and other related applications. Some examples of ontologies used for text generation are Upper Model (Bateman, 1997; Reiter and Dale, 2000) and SENSUS (Knight and Luk, 1994; Hovy, 1998).

[7] This approach is also widely used in machine translation.

drafts may be modified by editing the resulting text or by modifying the underlying procedures and re-drafting.

With regard to this system's scope, a study carried out by Power and Scott (1997) showed the commercial potential of DRAFTER and GIST; the main benefit is that they can reduce the time needed for producing texts in several languages, but there are other important advantages (op. cit.: 8):

1.— Improving document structure: the conceptual schemes for defining content in DRAFTER and GIST oblige users to base their documents on a clear structure that includes all essential components.

2.— Improving coherence: Automatic generation would ensure coherence of style and terminology across all documents for a given product line.

3.— New working procedures: Non-professional writers or translators could produce acceptable versions of routine passages, and the task of drafting these passages could thus be assigned to domain experts.

These benefits can also be applied to texts related to tourism. In recent years the tourism sector has seen a considerable increase in the number of bookings made on the Internet, which implies in many cases the need to create multilingual forms, general terms of business, and contracts. Our system, GenTur, has taken advantage of this opportunity and provides users with tourism contracts, specifically terms and conditions for package travel contracts.

On the other hand, the possibilities of authoring tools in the field of translation should be highlighted, as the latter can benefit from many of the former's multilingual applications: in this way, a generation system can serve as a valuable source of parallel texts for the translator, who can not only obtain documents in several languages, but also select different conceptual options in a language and obtain their equivalents in the rest of them. In this way, the documentation process would speed up. In this respect, Hartley and Paris (1995), as well as Power and Scott (1997), consider that translators can play an important role in DRAFTER:

> The new emphasis on user-centred design, of documentation as well as of products, requires translators to move outside their traditional job description to assume an authoring role with responsibility for the usability of the texts they produce. Given the imperative need to reduce the time-to-market of international products, translators in their emerging role find themselves involved at a much earlier stage than previously in the documentation process, and even in the design of the product itself. (Hartley and Paris, 1995: 1)

In other words, authoring tools give translators the opportunity to take part in the process of knowledge acquisition —that is, building language-independent conceptual models of a specific domain—, as well as the opportunity to use the multilingual sketches generated by the program to write and edit the complete texts (manuals, instructions, contracts, etc.).

3. THE GENTUR ARCHITECTURE

Our system, GenTur, could be considered both a pipelined system oriented to a serial process and an intercalated system, since discourse planning of utterances is determined at sentence level in a combined form. Another outstanding feature of GenTur is the presence of components related to integrated-type architectures, such as a common information space that stores plannings and variable values to be used later on during the generation process. Figure 1 depicts the GenTur architecture.

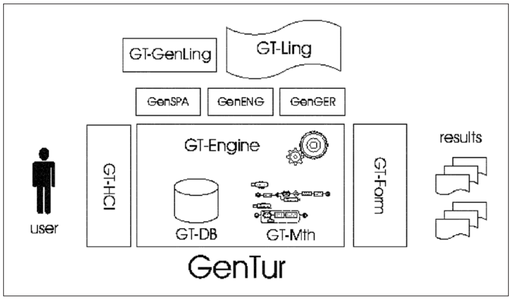


Figure 1: The GenTur Architecture

3.1. The GenTur Modules

GenTur is a modular system for the automatic generation of tourism contracts. In the following paragraphs we will describe the functionality of its modules and structures.

XGTLing: This is the interlingua used for the formal representation of tourism contracts and is language-independent at the discourse planning level. For this reason, XGTLing is not an interlingua in the standard sense of the term, since there is no underlying ontology, but a representation interlingua which expresses the information extracted from the *Turicor* corpus (textual forms, variables, etc.) —considering the contract as a collection of clauses, where each clause can be analysed in smaller units (textual forms and variables)— and which marks up the macrostructure and the superstructure of the information contained in the contracts (cf. Suthers, 1995; Benetos, 2005). In this way, the XGTLing tags are written directly with the text that realises them in the different languages.

On the other hand, we will use the terms *base contract* and *generating contract* in a given language to denominate the XGTLing–written formal superstructure that is able to generate a tourism contract correctly. In addition, it will be marked as GenL(XGTLing).

Closely related to this interlingua is the GT–GenLing module, which has been designed as an interface to assist translators and law experts when producing base contracts in the various target languages. At present, base contracts are produced in plain text (UNICODE). The data result from linguistic analyses which are performed on the *Turicor* corpus (Aguayo et al., 2004).

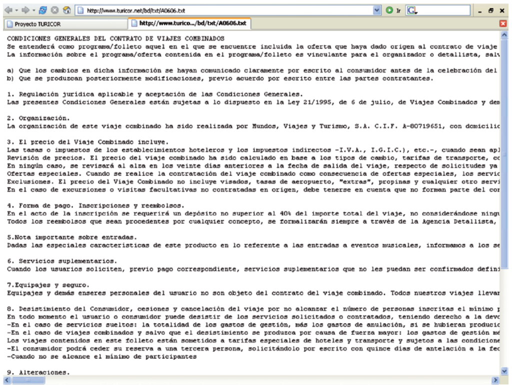
GT–Engine: This is the core component of the NLG system for producing tourism contracts. This module feeds from base contracts written in XGTLing. GT–Engine consists of two fundamental items:GT–Mth, the algorithm for text generation, and GT–DB, the database that stores the specific data of a given contract (variables).

GT–HCI (GenTur user's interface): This subsystem interacts with users to guide them when acquiring data for the GT–Engine to feed the database with a new contract instance and when choosing the different instantiations allowed by the base contract as specified in the XGTLing.

GT–Form: This subsystem provides the final representation of the generated document (surface realisation). It has been designed to generate contracts as XML documents with a uniform format.

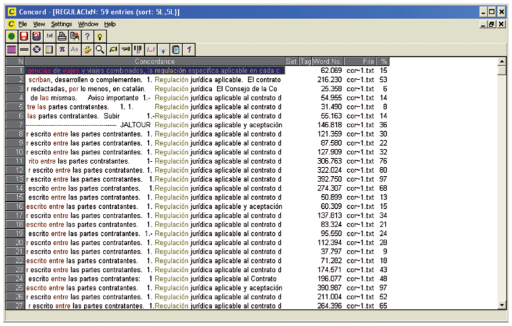
3.2. Preliminary work

Preparing base contracts in XGTLing is a non-trivial process. Two essential elements


http://www.turicor.net/bd/txt/A0606.txt
Ir
Proyecto TURICOR
http://www.turico.../bd/txt/A0606.txt

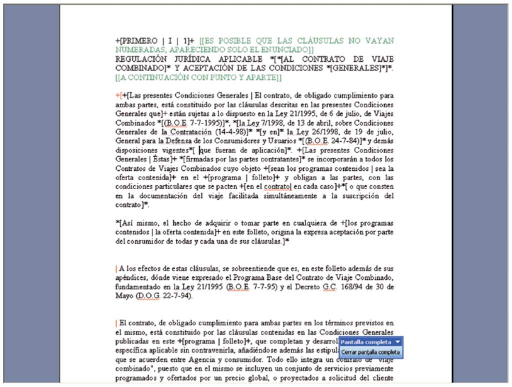
```
CONDICIONES GENERALES DEL CONTRATO DE VIAJES COMBINADOS
Se entenderá como programa/folleto aquel en el que se encuentre incluida la oferta que haya dado origen al contrato de viaje
La información sobre el programa/oferta contenida en el programa/folleto es vinculante para el organizador o detallista, salv

a) Que los cambios en dicha información se hayan comunicado claramente por escrito al consumidor antes de la celebración del
b) Que se produzcan posteriormente modificaciones, previo acuerdo por escrito entre las partes contratantes.

1. Regulación jurídica aplicable y aceptación de las Condiciones Generales.
Las presentes Condiciones Generales están sujetas a lo dispuesto en la Ley 21/1995, de 6 de julio, de Viajes Combinados y dem

2. Organización.
La organización de este viaje combinado ha sido realizada por Mundos, Viajes y Turismo, S.A. C.I.F. A-80719651, con domicilic

3. El precio del Viaje Combinado incluye.
Las tasas o impuestos de los establecimientos hoteleros y los impuestos indirectos -I.V.A., I.G.I.C.), etc.-, cuando sean apl
Revisión de precios. El precio del viaje combinado ha sido calculado en base a los tipos de cambio, tarifas de transporte, cc
En ningún caso, se revisará al alza en los veinte días anteriores a la fecha de salida del viaje, respecto de solicitudes ya
Ofertas especiales. Cuando se realice la contratación del viaje combinado como consecuencia de ofertas especiales, los servic
Exclusiones. El precio del Viaje Combinado no incluye visados, tasas de aeropuerto, "extras", propinas y cualquier otro servi
En el caso de excursiones o visitas facultativas no contratadas en origen, debe tenerse en cuenta que no forman parte del cor

4. Forma de pago. Inscripciones y reembolsos.
En el acto de la inscripción se requerirá un depósito no superior al 40% del importe total del viaje, no considerándose ningu
Todos los reembolsos que sean procedentes por cualquier concepto, se formalizarán siempre a través de la Agencia Detallista,

5.Nota importante sobre entradas.
Dadas las especiales características de este producto en lo referente a las entradas a eventos musicales, informamos a los se

6. Servicios suplementarios.
Cuando los usuarios soliciten, previo pago correspondiente, servicios suplementarios que no les puedan ser confirmados defini

7.Equipajes y seguro.
Equipajes y demás enseres personales del usuario no son objeto del contrato del viaje combinado. Todos nuestros viajes llevar

8. Desistimiento del Consumidor, cesiones y cancelación del viaje por no alcanzar el número de personas inscritas el mínimo p
En todo momento el usuario o consumidor puede desistir de los servicios solicitados o contratados, teniendo derecho a la devo
-En el caso de servicios sueltos: la totalidad de los gastos de gestión, más los gastos de anulación, si se hubieran producic
-En el caso de viajes combinados y salvo que el desistimiento se produzca por causa de fuerza mayor: los gastos de gestión má
Los viajes contenidos en este folleto están sometidos a tarifas especiales de hoteles y transporte y sujetos a las condicione
-El consumidor podrá ceder su reserva a una tercera persona, solicitándolo por escrito con quince días de antelación a la fec
-Cuando no se alcance el mínimo de participantes

9. Alteraciones.
```

Figure 2: Document from Turicor corpus: terms and conditions of a package holiday contract (Spanish)


Concord - [REGULACIxN: 59 entries (sort: 5L,5L)]
File View Settings Window Help

N	Concordance	Set	Tag	Word No.	File	%
1	gencias de viajes y viajes combinados, la regulación específica aplicable en cada c			62.069	cor~1.txt	15
2	scriban, desarrollen o complementen. 1. Regulación jurídica aplicable. El contrato			216.230	cor~1.txt	53
3	r redactadas, por lo menos, en catalán. Regulación jurídica El Consejo de la Co			25.358	cor~1.txt	6
4	de las mismas. Aviso importante 1.- Regulación jurídica aplicable al contrato d			54.955	cor~1.txt	14
5	tre las partes contratantes. 1. 1. Regulación jurídica aplicable al contrato d			31.490	cor~1.txt	8
6	las partes contratantes. Subir 1.-Regulación jurídica aplicable al contrato d			55.163	cor~1.txt	14
7	------------------------------------ JALTOUR Regulación jurídica aplicable y aceptación			146.818	cor~1.txt	36
8	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			121.359	cor~1.txt	30
9	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			87.580	cor~1.txt	22
10	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			127.909	cor~1.txt	32
11	rito entre las partes contratantes. 1- Regulación jurídica aplicable al contrato d			306.763	cor~1.txt	76
12	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			322.024	cor~1.txt	80
13	r escrito entre las partes contratantes. 1.Regulación jurídica aplicable al contrato d			392.750	cor~1.txt	97
14	escrito entre las partes contratantes. 1.Regulación jurídica aplicable al contrato d			274.307	cor~1.txt	68
15	escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			50.899	cor~1.txt	13
16	escrito entre las partes contratantes. 1. Regulación jurídica aplicable y aceptación			60.309	cor~1.txt	15
17	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			137.813	cor~1.txt	34
18	escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			83.324	cor~1.txt	21
19	escrito entre las partes contratantes. 1.- Regulación jurídica aplicable al contrato d			95.550	cor~1.txt	24
20	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			112.394	cor~1.txt	28
21	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			37.797	cor~1.txt	9
22	escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			71.282	cor~1.txt	18
23	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			174.571	cor~1.txt	43
24	escrito entre las partes contratantes: 1 Regulación jurídica aplicable al Contrato			196.077	cor~1.txt	48
25	escrito entre las partes contratantes. 1. Regulación jurídica aplicable y aceptación			390.987	cor~1.txt	97
26	r escrito entre las partes contratantes. 1. Regulación jurídica aplicable al contrato d			211.004	cor~1.txt	52
27	r escrito entre las partes contratantes. 1.Regulación jurídica aplicable al contrato d			264.396	cor~1.txt	65

Figure 3: Concordances performed by WordSmith Tools for the term regulación within Spanish package holiday contracts



+[PRIMERO | I | 1]+ [[ES POSIBLE QUE LAS CLÁUSULAS NO VAYAN NUMERADAS, APARECIENDO SOLO EL ENUNCIADO]]
REGULACIÓN JURÍDICA APLICABLE *[*[AL CONTRATO DE VIAJE COMBINADO]* Y ACEPTACIÓN DE LAS CONDICIONES *[GENERALES]*]*.
[[A CONTINUACIÓN CON PUNTO Y APARTE]]

+[+[Las presentes Condiciones Generales | El contrato, de obligado cumplimiento para ambas partes, está constituido por las cláusulas descritas en las presentes Condiciones Generales que]+ están sujetas a lo dispuesto en la Ley 21/1995, de 6 de julio, de Viajes Combinados *[(B.O.E. 7-7-1995)]*, *[la Ley 7/1998, de 13 de abril, sobre Condiciones Generales de la Contratación (14-4-98)]* *[y en]* la Ley 26/1998, de 19 de julio, General para la Defensa de los Consumidores y Usuarios *[(B.O.E. 24-7-84)]* y demás disposiciones vigentes*[que fueran de aplicación]*. +[Las presentes Condiciones Generales | Éstas]+ *[firmadas por las partes contratantes]* se incorporarán a todos los Contratos de Viajes Combinados cuyo objeto +[sean los programas contenidos | sea la oferta contenida]+ en el +[programa | folleto]+ y obligan a las partes, con las condiciones particulares que se pacten +[en el contrato| en cada caso]+*[o que consten en la documentación del viaje facilitada simultáneamente a la suscripción del contrato]*.

[Así mismo, el hecho de adquirir o tomar parte en cualquiera de +[los programas contenidos | la oferta contenida]+ en este folleto, origina la expresa aceptación por parte del consumidor de todas y cada una de sus cláusulas.]

| A los efectos de estas cláusulas, se sobreentiende que es, en este folleto además de sus apéndices, dónde viene expresado el Programa Base del Contrato de Viaje Combinado, fundamentado en la Ley 21/1995 (B.O.E. 7-7-95) y el Decreto G.C. 168/94 de 30 de Mayo (D.O.G. 22-7-94).

| El contrato, de obligado cumplimiento para ambas partes en los términos previstos en el mismo, está constituido por las cláusulas contenidas en las Condiciones Generales publicadas en este +[programa | folleto]+, que completan y desarrol específica aplicable sin contravenirla, añadiéndose además las estipul que se acuerden entre Agencia y consumidor. Todo ello integra un contrato de viaje combinado", puesto que en el mismo se incluyen un conjunto de servicios previamente programados y ofertados por un precio global, o proyectados a solicitud del cliente

Figure 4: Text prototype for the legislation clause of the Spanish terms and conditions of package holiday contracts

intervene in this: on the one hand, the actual *Turicor* corpus, which comprises a wide range of documents (tourism contracts, forms, rules and regulations, legislation, reference material, etc.). And, on the other hand, the law experts and linguists, who are in charge of developing the base contracts from corpus-driven data. Then, a three-step protocol is followed in order to feed the GenTur system:

1.— Writing the base contract in XGTLing. Firstly, the contracts in the *Turicor* corpus (fig. 2) are studied —using software applications such as WordSmith Tools— in order to determine their main linguistic and textual features (fig. 3). Then, the data taken from the corpus are used to build text prototypes (fig. 4): they are, as Corpas Pastor (2003) explains, texts written according to the characteristic superstructure, macrostructure and microstructure of the previously analysed contracts. The next step is to formally write these prototypes in XGTLing, so that a superstructure is reached in the form of a full formal logic structure division of the source contract.

2.— Creating and adapting the textual segments in XGTLing to the selected target languages that will be later used by the generation engine. Once the contract has been analysed and segmented, appropriate base contracts have to be prepared for the remaining selected target languages.

3.— Adapting the structures of the contracts described in XGTLing to each of the selected languages chosen for generation. Base contracts are obtained for each language and contract type. Together with the user's data and choices, base contracts will be used later on to generate contracts as output by means of the GT–Mth algorithm.

4. THE XGTLING INTERLINGUA

XGTLing is an interlingua that has been developed to formally represent the semi–structured paragraphs that make up a contract. Taking XGTLing as a starting point, it is possible to generate a contract in any target language for which there is a formal specification written in this interlingua.

4.1. General Structure

XGTLing is based on a tag scheme (similarly to XML-based languages) according to the following general structure:

```
‘<’ <tag> <parametres_list> ‘>’
          <block_content_tag>
‘</’ <tag> ‘>’
```

Taking into account the linguistic features of the contracts in the *Turicor* corpus, XGTLing has been structured around:

a) the preamble, which contains by way of header the basic data of the contract;

b) the body, which includes the description of the contract clauses.

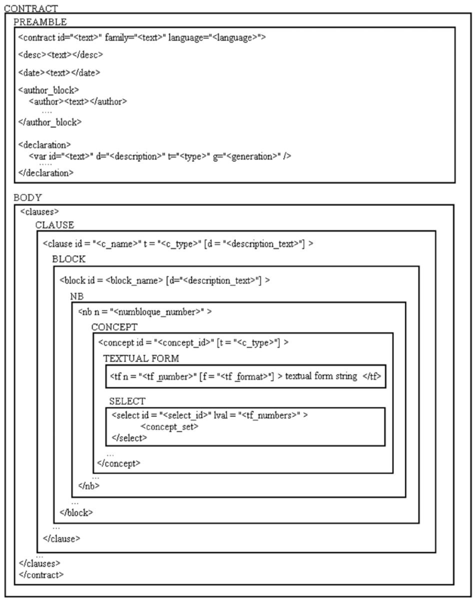
The preamble and body structures are described in the next section. Figure 5 depicts the general structure of XGTLing.

4.2. Preamble of a contract

The preamble of a base contract will contain data such as the name or title of the contract, the description, the authors, the revision dates, etc. There will also be a declaration of variables that will appear later in the contract clauses. Relevant tags or directives are the following:

<contract> This directive defines a contract. It contains several attributes: id, family and language.

<desc> This directive contains a brief description of the document.



```
<contract id="<text>" family="<text>" language="<language>">
<desc><text></desc>
<date><text></date>
<author_block>
   <author><text></author>
      ....
</author_block>

<declaration>
   <var id="<text>" d="<description>" t="<type>" g="<generation>" />
      .....
</declaration>

<clauses>
   <clause id = "<c_name>" t = "<c_type>" [d = "<description_text>"] >
      <block id = <block_name> [d="<description_text>"] >
         <nb n = "<numbloque_number>" >
            <concept id = "<concept_id>" [t = "<c_type>"] >
               <tf n = "<tf_number>" [f = "<tf_format>"] > textual form string  </tf>
               <select id = "<select_id>" lval = "<tf_numbers>" >
                        <concept_set>
               </select>
               ...
            </concept>
            ...
         </nb>
         ...
      </block>
      ...
   </clause>
   ...
</clauses>
</contract>
```

Figure 5: XGTLing general structure

<date> This is the tag of initial date.

<author_block> This directive names all the people who have created or modified the base contract in XGTLing.

<declaration> This directive contains all variables that appear in the textual forms of the entire document. The variables contained within the directive <var> enable users to adapt the generated contract to meet their needs, as the values for such variables are selected by the own users. This tag contains the following attributes:

<clauses> This directive contains the concept structure of the contract. This tag will be composed of tags <clause>. A set of clauses defines the document and, thus, contains the body of the contract. The set of clauses specifies the tags <block>, <nb>, <concept>, <tf> and <select>, which will be described in the next section.

We can see the general contract structure in Figure 5.

4.3. Clause Type tag

The clause is the semantic unit of the contract. The general format of a clause is determined by the structure shown in Table 1 (from now onwards we will provide a BNF (Backus-Naur Form) description of the interlingua). The parameter id corresponds to the clause name, while t represents the clause type. If t has value «N», this specific clause is necessary and should be included in the contract. On the contrary, when t has value «O» the clause can appear optionally in the contract. Finally, the optional parameter d corresponds to the textual description of the clause semantic content. A clause consists of blocks.

```
<clause> ::== '<'clause id="<c_name>" t="<c_type>"
        [d="<description_text>"] '>'
        <block_set>
'</' clause '>'
```

Table 1: Tag specification of clause type

4.4. Block tag

Table 2 illustrates the block format. The parameter id identifies the block. A block may be rendered in different textual forms. Each optional textual form is marked by a blockNum tag. This tag designates the units in which a block can be decomposed. When generating a contract, a command will skip a line to separate the blocks.

```
<block> ::== ’<’ block id=»<block_name>»
                        [d=»<description_text>»] ’>’
        <nb_set>
’</’ block ’>’
```

Table 2: Block tag specification

4.5. BlockNum tag

During text generation users have to select one textual realisation (blockNum) among the several possibilities that there may exist for a given block. Table 3 shows the structure of this directive, which has only one attribute (n). The attribute n refers to the number of the blockNum (≥ 0). A blockNum is composed of concepts that express similar ‘conceptual structures’.

```
<nb> ::== ’<’ nb n=»<blockNum_number>» ’>’
        <concept_set>
’</’ nb ’>’
```

Table 3: blockNum tag specification

4.6. Concept tag

Table 4 shows the format of this directive. It has two attributes. The attribute id identifies a given, unique concept among several concepts within the same blockNum. The second attribute —t— refers to type and it indicates whether the concept is obligatory or optional. In case that attribute t does not appear, the concept will be considered as obligatory.

Each concept consists of a series of possible textual forms and selection tags. Of the textual forms in which a given concept can be rendered, only one can be selected to appear in the target document. Selection tags indicate a series of concepts that will be included in the target document if a certain textual form has been selected.

```
<concept> ::== ’<’ concept id=»<concept_id>»
                              [t=»<c_type>»] ’>’
        <tf_select_set>
’</’ concept ’>’
```

Table 4: Concept tag specification

4.7. Textual Form tag

Any given concept can be expressed by a range of possible textual forms. The user has to select the particular textual form in which a given concept is to be rendered in the target document. Table 5 illustrates the format of this directive:

```
<tf> ::== '<'tf n=»<tf_number>» [f=»<tf_
                     format>»]'>'<string> '</'tf'>'
```

Table 5: Tag specification of textual forms

n: this attribute indicates the number of the textual form (≥ 0). This attribute shows a unique value for each textual form in which a given concept can be expressed.

f: this is an optional attribute that indicates the format of the textual form. If this attribute does not appear, the textual form will not have any particular formatting in the target document. Otherwise, this attribute will be a chain of four characters, that make reference to centred, bold, underlined and italics.

The content of a textual form is a chain of characters. The chain will be analysed in case a given textual form has been selected to render a specific concept (among other possibilities). The following structures can appear in a chain of characters:

Variables: A variable appears in the form %var_name, where var_name makes reference to a variable whose value for the attribute id is var_name.

Optionality: The sequence *[option1 | option2…| optionN]* indicates optionality. If a textual form is marked as optional, that means that it may appear as option 1, 2, etc., or else that it may not appear at all, depending on the user's choice.

Obligatority: Obligatority is marked as +[option1 | option2…| optionN]+, which means that the generated document will contain option 1, option 2, … or option n, according to the user's selection.

The remaining content of a textual form is not processed and will appear unchanged in the target contract that is generated.

4.8. Select tag

The selection tag select determines that certain information may appear in the generated text provided a textual form has been previously selected. Concepts are composed of both textual forms and selection tags. A selection tag consists of a set of concepts. Table 6 shows the format of this new directive.

```
<select> ::== '<' select id=»<select_id>»
                         lval=»<tf_numbers>» '>'
      <set_of_concepts>
'</' select '>'
```

Table 6: Select tag specification

5. TEXT GENERATION ALGORITHM, IMPLEMENTATION AND SOFTWARE

The generation algorithm is central to the system core. It generates in sequence contracts in various languages according to user's specifications and choices (see 5.1.). It also performs a series of complementary tasks (see 5.2.). Section 5.3 provides a sample of the entire process of text generation.

5.1. The generation algorithm

Prior to generating the target document, it is necessary to annotate in XGTLing the set of contracts that are going to be used. Then,

```
cl := select all clauses
go_beginning(cl)
While there are_clauses(cl) left Do{
    If it is _obligatory (cl) or it has_to appear compulsorily (cl){
            bl := select all blocks(cl)
            go_beginning(bl)
            While there are_blocks(bl) left Do{
                nb := select all blocks(bl)
                nb_sel := select_numBlock(nb)
                cpt := select all concepts (nb_sel)
                While there are _concepts (cpt) left Do{
                    If it is_obligatory(cpt) or has to appear obligatory(cpt){
                        tf := select textual forms(cpt)
                        Include_target_contracts(Analise(tf))
                    }
                    go_next(cpt)
                }
                go_next(bl)
            }
    }
    go_next(cl)
}
```

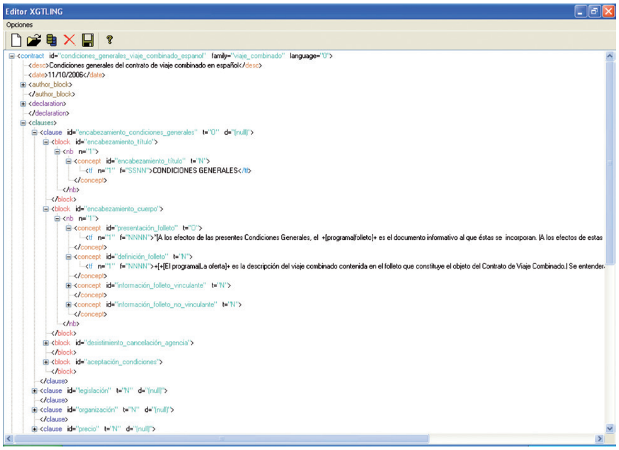
Table 7: Generation algorithm

annotated base contracts are analysed by a syntactical and a lexical parser in order to determine whether they have been correctly tagged. If any of these base contracts contained errors, a text file would show a list of errors and their corresponding lines. When all base contracts are correctly tagged, the information is stored in the database. Then, the system generates target contracts in the languages selected. The algorithm executes upon all the clauses, the blocks contained in such clauses, and so on until it reaches the most internal tag, namely, the textual forms. Table 7 depicts a pseudocode approach to the general process of text generation.

The generation algorithm is based on the selection of a guide language. This language bears correspondence to the language in which the variables, clauses and text forms of the contract to be generated are going to be instantiated. In this case it should be the user's source language. Once the guide language has been selected, the user will select the different alternatives among clauses, blocks and text forms and also complete the variables that the system requests.

Each time a selection or variable instantiation is carried out, it impinges on the rest of languages in which the contract is going to be generated, producing at the end of the process

Figure 6: XGTLing main menu



```
<contract id="condiciones_generales_viaje_combinado_espanol" family="viaje_combinado" language="0">
  <desc>Condiciones generales del contrato de viaje combinado en español</desc>
  <date>11/10/2006</date>
  <author_block>
  </author_block>
  <declaration>
  </declaration>
  <clauses>
    <clause id="encabezamiento_condiciones_generales" t="0" d="(null)">
      <block id="encabezamiento_título">
        <nb n="1">
          <concept id="encabezamiento_título" t="N">
            <tf n="1" f="SSNN">CONDICIONES GENERALES</tf>
          </concept>
        </nb>
      </block>
      <block id="encabezamiento_cuerpo">
        <nb n="1">
          <concept id="presentación_folleto" t="0">
            <tf n="1" f="NNNN">"[A los efectos de las presentes Condiciones Generales, el +[programa|folleto]+ es el documento informativo al que éstas se incorporan. |A los efectos de estas
          </concept>
          <concept id="definición_folleto" t="N">
            <tf n="1" f="NNNN">+[+[El programa|La oferta]+ es la descripción del viaje combinado contenida en el folleto que constituye el objeto del Contrato de Viaje Combinado.| Se entender
          </concept>
          <concept id="información_folleto_vinculante" t="N">
          </concept>
          <concept id="información_folleto_no_vinculante" t="N">
          </concept>
        </nb>
      </block>
      <block id="desistimiento_cancelación_agencia">
      </block>
      <block id="aceptación_condiciones">
      </block>
    </clause>
    <clause id="legislación" t="N" d="(null)">
    </clause>
    <clause id="organización" t="N" d="(null)">
    </clause>
    <clause id="precio" t="N" d="(null)">
```

Figure 7: Spanish interlingua written with the XGTLing editor

a contract written in the guide language, as well as the corresponding ones in the languages wanted for generation.

5.2. *The Generation Software*

The most important function of the generation software is to produce target contracts according to users' choices. However, the system can also perform other tasks, such as:

Contract management. Base contracts are stored in the database until the user decides to discard them. Thus they can be re-used to skip part of the generation process, as some variables and users' choices may remain the same.

Family management. Contracts are organised in families. In fact, there is an attribute family in the directive <contract> which allows the user to have access to existing families, to create new ones or to add new contracts to those families.

Language management. The system provides users with the possibility of checking the actual languages in the database, to eliminate any particular language or to enter a new one in the database.

Editor. An editor for XGTLing has been designed in order to ease the tedious but necessary task of tagging the base contracts. Documents are then easily rendered in the interlingua from a menu with options for authors, variables and so forth.

Dictionary management. The system incorporates a basic multilingual dictionary that links translation equivalents that are repeatedly used during the generation process.

All these operations can be performed from the main menu (see fig. 6).

5.3. *The process of text generation*

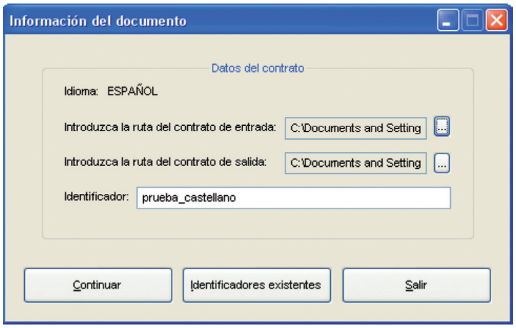
The first step of the process of multilingual text generation is to produce documents written in the XGTLing interlingua in the several languages involved. When creating a new document, the editor will insert the necessary tags (<clause>, <block>, <nb>, <concept>, <tf>, <select>, <author>, <var>) as appropriate.

Figure 7 illustrates a document in Spanish which has been written by the editor in XGTLing.

The second step is to enter the information relevant to each document and target language, that is to say:

—the route for the input document;

—the route for the output document;


Información del documento

Datos del contrato

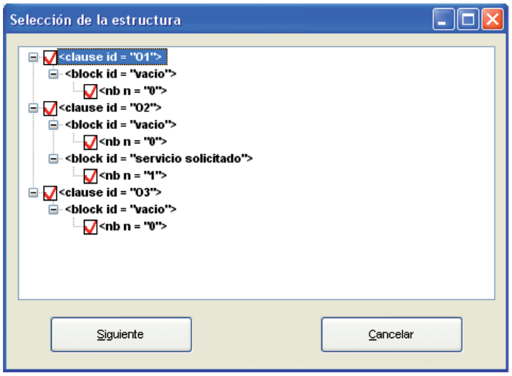
Idioma: ESPAÑOL

Introduzca la ruta del contrato de entrada: C:\Documents and Setting

Introduzca la ruta del contrato de salida: C:\Documents and Setting

Identificador: prueba_castellano

Figure 8: Form to enter the document data


Selección de la estructura

```
<clause id = "01">
  <block id = "vacio">
    <nb n = "0">
<clause id = "02">
  <block id = "vacio">
    <nb n = "0">
  <block id = "servicio solicitado">
    <nb n = "1">
<clause id = "03">
  <block id = "vacio">
    <nb n = "0">
```

Figure 9: Selection of contract structure

—the document identification for storage in the database.

The information is then entered through the form illustrated in figure 8.

Once all relevant information has been provided for the input document, the third step is to check that the document is written correctly in the XGTLing interlingua, that the appropriate language has been selected and that all documents (if several) belong to the same contract family.

The checker is composed of a lexical parser and a syntactic parser. The lexical parser divides the input document in minimum information units (tokens). The syntactic parser checks if the tokens follow the interlingua formal grammar.

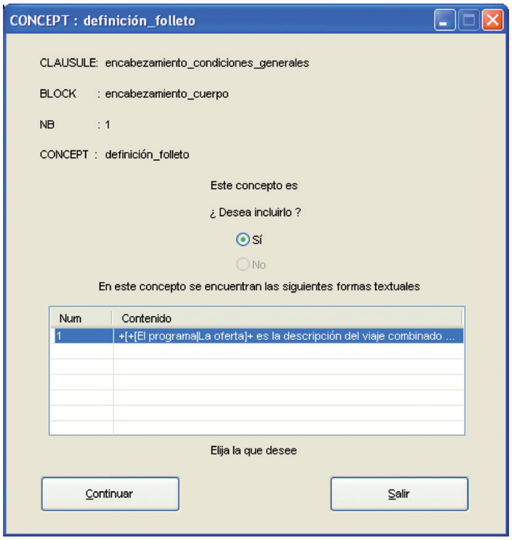
Next a dialogue box will tell the user whether the base contract is correctly constructed. After the input document has been analysed and checked for correctness, the system stores in the database all the information pertaining to the document, such as authors, variables, clauses, etc. The actual process of generation starts once all the information has been conveniently stored in the database.

A fourth step is the selection of the structure of the document, i.e. any optional clauses that it may contain and which blockNums within each block should be analysed. Choices are made through a conceptual tree (fig. 9).

Then, the system analyses all the concepts included in the blockNums which the user had selected in the previous step. For each concept analysis a dialogue box will provide the relevant data: the blockNum, block and corresponding clause, optionality and textual forms assigned to the concept (see fig. 10 and 11). Finally, once all concepts have been analysed, a dialogue box will allow visualising any output document that has been just generated. Figure 12 illustrates a sample target contract in Spanish.

6. CONCLUSION

In this paper we have introduced an authoring tool for tourism contracts, GenTur, which might prove a plausible solution for an ever-growing multilingual working environment. Thanks to its XGTLing interlingua, GenTur is an open authoring system which can be used to generate practically all kinds of contracts in many languages. In addition, it may serve as a full guide to any user who wishes to reproduce the contents of a given contract in several languages simultaneously.


CONCEPT : definición_folleto

CLAUSULE: encabezamiento_condiciones_generales

BLOCK : encabezamiento_cuerpo

NB : 1

CONCEPT : definición_folleto

Este concepto es

¿ Desea incluirlo ?

Sí [✓]

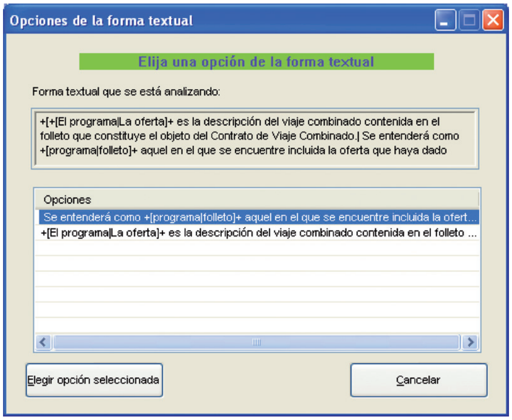
No

En este concepto se encuentran las siguientes formas textuales

Num	Contenido
1	+[+[El programa|La oferta]+ es la descripción del viaje combinado ...

Elija la que desee

Figure 10: Analysed concept


Opciones de la forma textual

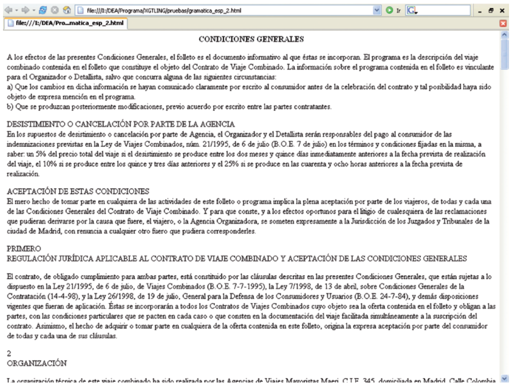
Elija una opción de la forma textual

Forma textual que se está analizando:

+[+[El programa|La oferta]+ es la descripción del viaje combinado contenida en el folleto que constituye el objeto del Contrato de Viaje Combinado.| Se entenderá como +[programa|folleto]+ aquel en el que se encuentre incluida la oferta que haya dado

Opciones
Se entenderá como +[programa|folleto]+ aquel en el que se encuentre incluida la ofert...
+[El programa|La oferta]+ es la descripción del viaje combinado contenida en el folleto ...

Figure 11: Selection of text form


file:///I:/DEA/Programa/XGTLING/pruebas/gramatica_esp_2.html

CONDICIONES GENERALES

A los efectos de las presentes Condiciones Generales, el folleto es el documento informativo al que éstas se incorporan. El programa es la descripción del viaje combinado contenida en el folleto que constituye el objeto del Contrato de Viaje Combinado. La información sobre el programa contenida en el folleto es vinculante para el Organizador o Detallista, salvo que concurra alguna de las siguientes circunstancias:
a) Que los cambios en dicha información se hayan comunicado claramente por escrito al consumidor antes de la celebración del contrato y tal posibilidad haya sido objeto de expresa mención en el programa.
b) Que se produzcan posteriormente modificaciones, previo acuerdo por escrito entre las partes contratantes.

DESISTIMIENTO O CANCELACIÓN POR PARTE DE LA AGENCIA
En los supuestos de desistimiento o cancelación por parte de Agencia, el Organizador y el Detallista serán responsables del pago al consumidor de las indemnizaciones previstas en la Ley de Viajes Combinados, núm. 21/1995, de 6 de julio (B.O.E. 7 de julio) en los términos y condiciones fijadas en la misma, a saber: un 5% del precio total del viaje si el desistimiento se produce entre los dos meses y quince días inmediatamente anteriores a la fecha prevista de realización del viaje, el 10% si se produce entre los quince y tres días anteriores y el 25% si se produce en las cuarenta y ocho horas anteriores a la fecha prevista de realización.

ACEPTACIÓN DE ESTAS CONDICIONES
El mero hecho de tomar parte en cualquiera de las actividades de este folleto o programa implica la plena aceptación por parte de los viajeros, de todas y cada una de las Condiciones Generales del Contrato de Viaje Combinado. Y para que conste, y a los efectos oportunos para el litigio de cualesquiera de las reclamaciones que pudieran derivarse por la causa que fuere, el viajero, o la Agencia Organizadora, se someten expresamente a la Jurisdicción de los Juzgados y Tribunales de la ciudad de Madrid, con renuncia a cualquier otro fuero que pudiera corresponderles.

PRIMERO
REGULACIÓN JURÍDICA APLICABLE AL CONTRATO DE VIAJE COMBINADO Y ACEPTACIÓN DE LAS CONDICIONES GENERALES

El contrato, de obligado cumplimiento para ambas partes, está constituido por las cláusulas descritas en las presentes Condiciones Generales, que están sujetas a lo dispuesto en la Ley 21/1995, de 6 de julio, de Viajes Combinados (B.O.E. 7-7-1995), la Ley 7/1998, de 13 de abril, sobre Condiciones Generales de la Contratación (14-4-98), y la Ley 26/1998, de 19 de julio, General para la Defensa de los Consumidores y Usuarios (B.O.E. 24-7-84), y demás disposiciones vigentes que fueran de aplicación. Éstas se incorporarán a todos los Contratos de Viajes Combinados cuyo objeto sea la oferta contenida en el folleto y obligan a las partes, con las condiciones particulares que se pacten en cada caso o que consten en la documentación del viaje facilitada simultáneamente a la suscripción del contrato. Asimismo, el hecho de adquirir o tomar parte en cualquiera de la oferta contenida en este folleto, origina la expresa aceptación por parte del consumidor de todas y cada una de sus cláusulas.

2
ORGANIZACIÓN

La organización técnica de este viaje combinado ha sido realizada por las Agencias de Viajes Mayoristas Maeri, C.I.F. 345, domiciliada en Madrid, Calle Colombia

Figure 12: Sample of output document (Spanish)

Our primary focus has been the XGTLing interlingua, the true core of the GenTur system. XGTLing main features make it suitable for information representation, as it is flexible enough to provide a formal representation of the kind of semi–structured information that may be found in a contract. Especially remarkable is its portability: it allows processing by tools for uniform representation of information as base contracts are described in a XML-like language.

Future lines of research may involve ontology-based work to link the textual forms of similar concepts in several languages, with a view to developing a multilingual generation system that is able to generate the same base contract in several languages simultaneously. Furthermore, this ontology-based approach will provide an interesting and very up-to-date application: the reutilisation of information resources in areas such as tourism, translation or terminology.

RECIBIDO EN AGOSTO 2007
ACEPTADO EN DICIEMBRE 2007

7. REFERENCES

Aguayo, A., J. L. Caro, G. Corpas, I. Gómez, and A. Guevara. (2004). «Turicor: Un modelo conceptual de datos para almacenamiento y consulta de un corpus multilingüe». In *V Congreso Nacional: Turismo y Tecnologías de la Información y las Comunicaciones*, pp. 327–344.

Barrutieta, G. (2001). *Generador inteligente de documentos de formación* [on line]. <http://sapiens.ya.com/cyberformacion/Generador.pdf> [Accessed 26 May 2007].

Bateman, J. (1997). «Enabling technology for multilingual natural language generation: the KPML development environment». *Journal of Natural Language Engineering*, 3/1, pp. 15-55.

Bateman, J. and M. Zock. (2003). «Natural Language Generation». In R. Mitkov (ed.) (2003). *The Oxford handbook of computational linguistics*, Oxford: Oxford University Press.

Benetos, K. (2005). *XML grammar for constructing argumentative essays* [on line]. <http://tecfaseed.unige.ch/staf18/modules/ePBL/uploads/proj7/paper4.xml> [Accessed 26 June 2007].

Bernardos, S. (2003). *Marco metodológico para la construcción de sistemas de generación de lenguaje natural*. Doctoral thesis. Madrid: Facultad de Informática, Universidad Politécnica de Madrid.

Cahill, L., C. Doran, R. Evans, R. Kibble, C. Mellish, D. Paiva, M. Reape, D. Scott, and N. Tipper (2000). «Enabling resource sharing in language generation: an abstract reference architecture». In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*.

Corpas Pastor, G. (2003). «Diseño de un *tipologizador* para la traducción jurídica: del corpus al prototipo textual». In G. Corpas Pastor (ed.). (2003). *Recursos documentales y tecnológicos para la traducción del discurso jurídico (español, alemán, inglés, italiano, árabe)*, Granada: Comares.

Dale, R., B. Di Eugenio and D. Scott (1998). «Introduction to the Special Issue on Natural Language Generation» [on line]. *Computational Linguistics*, 24/3, pp. 345-353. <http://www.cs.mu.oz.au/acl/J/J98/J98-3001.pdf> [Accessed 16 June 2007].

Fiedler, A. (2005). *Natural Language Generation* [on line]. <http://www.ags.uni-sb.de/~afiedler/lehre/NLG/lectures/nlg-05.pdf> [Accessed 17 June 2007].

Gruber, T. R. (1993). «A translation approach to portable ontology specification». *Knowledge acquisition*, 5/2, pp. 199-220.

Hartley, A. and C. Paris (1995). «Supporting Multilingual Document Production: Machine Translation or Multilingual Generation?». In *Working Notes of the Multilingual Text Generation Workshop, International Joint Conference in Artificial Intelligence (IJCAI-95)*, pp. 34-41.

Hartley, A. and C. Paris (1997). «Multilingual Document Production From Support for Translating to Support for Authoring». *Machine Translation*, 12, pp. 109–129.

Hartley, A. F., D. Scott, J. Bateman and D. Dochev (2001). «AGILE - A System for Multilingual Generation of Technical Instructions». In *Proceedings of 8th Machine Translation Summit (MT Summit VIII)*, pp. 145-150.

Hovy, E. 1996. «Overview (of Language Generation)». In R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen and V. Zue (eds.) *Survey of the State of the Art in Human Language Technology*. [on line]. <http://cslu.cse.ogi.edu/HLTsurvey/ch4node3.html#SECTION41> [Accessed 15 June 2007].

Hovy, E. (1998). «Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses». In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*.

Hutchins, J. (2005). «Computer-based Translation in Europe and North America, and its Future Prospects» [on line]. In *JAPIO 20th anniversary*. Tokyo: Japan Patent Information Organization. <http://ourworld.compuserve.com/homepages/WJHutchins/JAPIO-2005.pdf> [Accessed 15 June 2007].

Knight, K. and S. Luk (1994). «Building a Large-Scale Knowledge Base for Machine Translation». In *Proceedings of the American Association of Artificial Intelligence AAAI-94*.

Langkilde, I. (2002). «An Empirical Verification of Coverage and Correctness for a General-Purpose Sentence Generator» [on line]. In *Proceedings of the International Natural Language Generation Conference*. <http://faculty.cs.byu.edu/~irenelg/papers/verifcc.ps.gz> [Accessed 18 June 2007].

Nirenburg, S., V. Lesser, and E. Nyberg. (1989). «Controlling a language generation planner». In *Proceedings of the 11th. International Joint Conference on Artificial Intelligence*, pp. 1524–1530.

Paris, C., K. Vander Linden, M. Fischer, A. Hartley, L. Pemberton, R. Power, D. Scott (1995). «A Support Tool for Writing Multilingual Instructions».

In *International Joint Conference in Artificial Intelligence* (IJCAI-95), pp. 1398-1404.

Power, R. and D. Scott. (1997). «NLG tools to support technical authors and translators». ITRI technical report 97-06.

Reiter, E. and R. Dale. (2000). *Building Natural Language Generation Systems*, Cambridge: Cambridge University Press.

Scott, D. and R. Evans (1998). «Multilingual Document Management Without Translation: Using natural language generation in the Multilingual Information Society» [on line] *Elsnews*, 7/1. <http://mcs.open.ac.uk/ds5473/publications/IEE-paper.htm> [Accessed 15 June 2007].

Scott, D. R., N. Bouayad-Agha, R. Power, S. Schultz, R. Beck, D. Murphy and R. Lockwood. (2001). «PILLS: A Multilingual Authoring System for Patient Information». In S. Bakken (2001). *Visions of the Future and Lessons from the Past. Proceedings of the 2001 AMIA Annual Symposium, Washington, November 3-7*, Philadelphia: Hanley & Belfus, p. 1023.

Suthers, D. (1995) «Towards an Interlingua for Information Objects» [on line]. Position Paper presented at CHI'95 Research Symposium, Denver, Colorado, U.S.A <http://lilt.ics.hawaii.edu/lilt/papers/1995/suthers-chi95argml.pdf> [Accessed 8 July 2007].

Vander Linden, K. (2000). «Natural Language Generation». In D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Upper Saddle River (NJ): Prentice Hall, pp. 763-798.