OEG Publication

Aguado de Cea G, Bañón A, Bateman J, Bernardos MS, Fernández-López M, Gómez-Pérez A, Nieto E, Olalla A, Plaza R, Sánchez A

ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation

Workshop on Applications of Ontologies and Problem-Solving Methods
European Conference on Artificial Intelligence (ECAI'98)
August 1998.
Brighton, United Kingdom.

ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation

G. Aguado[1], **A. Bañón**[2], **J. Bateman**[3], **S. Bernardos**[2], **M. Fernández**[2]
A. Gómez-Pérez[2], **E. Nieto**[2], **A. Olalla**[2], **R. Plaza**[1] and **A. Sánchez**[2]

Abstract. A significant problem facing the reuse of ontologies is to make their content more widely accessible to any potential user. Wording all the information represented in an ontology is the best way to ease the retrieval and understanding of its contents. This article proposes a general approach to reuse domain and linguistic ontologies with natural language generation technology, describing a practical system for the generation of Spanish texts in the domain of chemical substances. For this purpose the following steps have been taken: (a) an ontology in the chemicals domain developed under the METHONTOLOGY framework and the Ontology Design Environment (ODE) has been taken as knowledge source; (b) the linguistic ontology GUM (Generalized Upper Model) used in other languages has been extended and modified for Spanish; (c) a Spanish grammar has been built following the systemic-functional model by using the KPML (Komet-Penman Multilingual) environment. As result, the final system named *Ontogeneration* permits the user to consult and retrieve all the information of the ontology in Spanish.

1 INTRODUCTION

One of the main goals of ontologies [38] is to increase shared understanding in a given domain, thereby eliminating differences, overlaps and mismatches in concepts, structures, terminologies, etc. In this way, ontologies can function as a framework that unifies different viewpoints and improves communication. An ontology has been defined by Gruber as "an explicit specification of a conceptualization" [19] that includes: concepts, instances, relations, functions and axioms [17]. However, while on the one hand, ontologies generally specify conceptualizations with a high degree of formality, on the other hand there are a few methodologies [12] [38] for building ontologies: different formalisms and languages (Ontolingua [18], CycL [27], LOOM [29], etc.) can be used to formalize the same domain knowledge at the symbol level. This fact makes it impossible that users without a certain background in this field can reuse, consult or understand the knowledge embedded in ontologies. Our experience shows that domain experts and human final users do not understand formal ontologies codified in such languages even if such languages have a browser and a graphic user interface to display the ontology content. One practical way that partially solves this problem is to present the ontology content in a set of intermediate representations at the knowledge level that can include graphs and tabular notations [7] which are more understandable for non ontologists than the formal languages used for codifying ontologies.

[1] Departamento de Lingüística Aplicada. Facultad de Informática de Madrid. Univ. Politécnica de Madrid. Campus de Montegancedo, sn. 28660 Boadilla del Monte, Madrid, Spain. *{lupe,rplaza}@fi.upm.es*

[2] Laboratorio de Inteligencia Artificial. Facultad de Informática de Madrid. Univ. Politécnica de Madrid. Campus de Montegancedo, sn. 28660 Boadilla del Monte, Madrid, Spain. *gum@delicias.dia.fi.upm.es*

[3] University of Stirling, Communication and Language Research, Dept. of English Studies, Stirling, FK9 4LA, Scotland U.K. *j.a.bateman@stir.ac.uk*

One way of effectively disseminating ontology contents appears to be to translate them into natural language. Wording in different languages all the information represented in an ontology is the best way to extend its accessibility to any user. Generation of NL texts from ontology contents would also permit domain experts to evaluate domain expert knowledge formalized by ontologists, as well as the reuse of domain ontologies in practical and commercial applications related to multilingual text generation, knowledge management, on-line information retrieval, natural language explanations in expert systems or intelligent tutoring systems, and database access.

Establishing a connection between the fields of text generation and ontology reuse is also of benefit for text generation in several ways. A generic key problem for generation systems is the availability of appropriately organized domain models. The sources of knowledge used for generation systems are many times hand-crafted (i.e. oriented towards the final language) or are built for some non-linguistic purposes which then fail to support the language production sufficiently to warrant the use of text generation technology at all. Here again, one possible solution is to reuse standardized or well-defined domain ontologies as a representational resource for text generation systems. The METHONTOLOGY framework [13] [16] supports precisely such a solution. It allows the specification of ontology at the knowledge level using a set of intermediate representations and the Ontology Design Environment translators [7] to generate standard, consistent and well-structured ontologies in several target languages (Ontolingua, SFK, SQL). ODE also includes inverse translators that transform Ontolingua code into our intermediate representations structures by a reverse engineering process described by [7]. METHONTOLOGY's tabular and graphically based notation is a user-friendly approach to knowledge acquisition and evaluation by domain experts that are not knowledge engineers. So, all the ontologies built under this approach are not hand-crafted, they rely upon the same conceptualization (they organize domain knowledge by means of concepts, instances, hierarchies, attributes, relations, axioms, etc.), they have been built independently of their final use and the ontology final code is generated automatically. CHEMICALS (that was not built for NL generation purposes) [12] is one of the ontologies that has been built under this approach and upon which an environmental ontology is also being built using a distributed architecture.

Since domain ontologies built under this approach do not include linguistic features, they need to be interfaced with the natural language generation systems employed. One means of interfacing with domain knowledge is to apply the linguistic ontology called the Generalized Upper Model (GUM) [1]. GUM offers a level of semantic abstraction that is sufficiently far removed from differences in surface realization to facilitate linking with well-structured domain models, but which nevertheless is still sufficiently close to linguistic form to support well-defined mappings from its concepts to linguistic expression.

The main advantages and direct consequences of reutilizing a linguistic ontology such as GUM are the complexity reduction of generation tasks (such as lexical and syntactic choices) and the lack of ambiguity in the text output: the set of possible sentences and

paragraphs derived from a high-formalized domain ontology constitutes a controlled language or unambiguous sublanguage.

Given these combined considerations, the aim of this paper is twofold: to propose an approach that reuses domain and linguistic ontologies in multilingual text generation systems and to introduce a specific system, Ontogeneration, as an instantiation of this approach. Ontogeneration is an information retrieval system which permits Spanish users to consult and access, in their own language, the knowledge contained in an ontology of chemical elements. Ontogeneration is a prototype whose development is in progress, but its current state serves as a clear example of how the two research fields, ontological engineering and natural language generation, can be linked, thereby solving some key problems in both disciplines.

Our focus in this paper is not, therefore, on the generation process itself and, in fact, we deliberate reuse as much of established generation technology and techniques as we can. Our main contribution must instead be seen from the point of view of:

(a) the reutilization of two different kind of ontologies built separately with different technology and purposes,
(b) the reuse of the Komet-Penman Multilingual technology (or KPML) [32] to build resources for Spanish text generation and
(c) the integration of these resources in a new application that generates Spanish texts.

With this in mind, in this paper we will firstly describe the general design of our application, Ontogeneration, and how ontologies and text generation can be combined in order to prove the benefits of our approach. Secondly, as our system integrates resources developed independently: a domain ontology (Chemicals, stored in a relational data base), a linguistic ontology (GUM, implemented in Loom) and a generation environment (KPML, in Common Lisp) already used in other projects, we will explain these resources, why and how we have reused them their adaptation or extension to Spanish, and their integration into the Ontogeneration architecture.

2 OVERVIEW OF *ONTOGENERATION*

The Ontogeneration architecture design is diagrammed in Figure 1. This figure is divided in three parts or levels: input/output, processes and resources. The overall process of any generation system is guided by goal pursuit. In our case, the main communicative goal is to offer all the information (about a given domain) requested by the user.

The system starts when the user makes a specific query by using a menu-based interface. All the possible queries have been previously predefined and classified in various patterns or templates. Thus, the user can compose easily a particular query selecting the different menu options.

The first process is the knowledge search and selection that extracts the relevant content from the domain ontology. Our system uses Chemicals [12], an ontology which describes and classifies all the chemical elements (see section 3). A further resource used by this process is the user model, i.e. a model of the intended user. The user model serves to guide the selection process in order to include unknown background information to him. It is useful to decide what information should be omitted and to measure the tone or degree of formality and other pragmatic effects (interpersonal and situational aspects).

The following process, text planning, arranges the selected knowledge into an appropriate rhetorical schema. Our rhetorical schemas represent standard patterns of scientific discourse. They can be described as stereotypical paragraphs templates that we have identified in scientific texts and chemistry manuals: definitions (of groups, elements, axioms, formulae and properties); comparisons (between groups or elements), examples (of groups or elements), classifications or constituency (of groups or elements) and others (complete descriptions or specifications). These patterns guide the text planning in the design of the text structure: this includes the organization of contents in a coherent discourse, paragraph decomposition into sentences, fixing the sentence boundaries and the use of conjunctions, use of reference expressions and ellipsis, and choice of marked syntactic constructions for rhetorical effects.

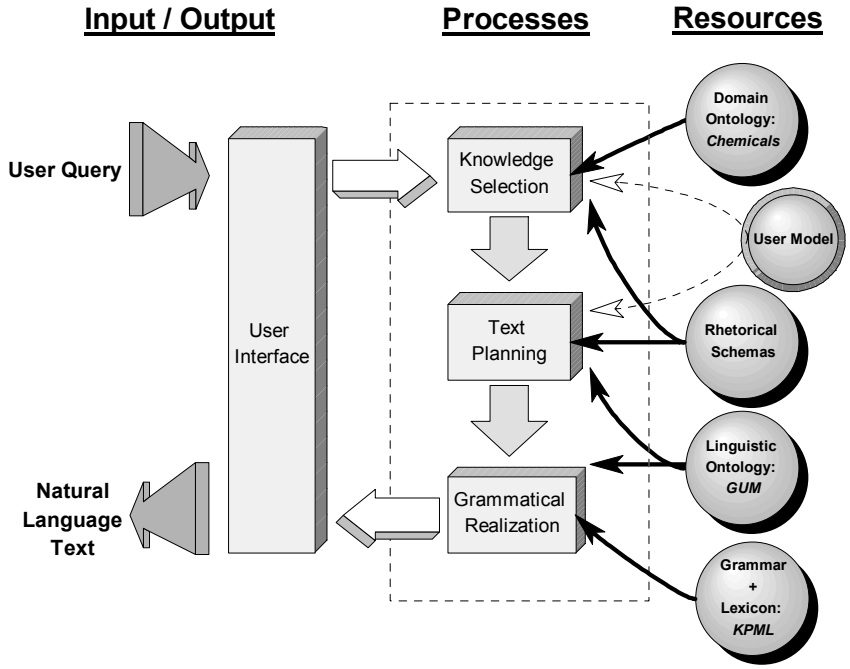

Figure 1. Design of the system

The process of grammatical realization is responsible for generating the final text by executing the text plan by turning it into wordings. This process relies on the linguistic ontology employed (GUM, see section 4) and lexicogrammatical resources that are being adapted for Spanish (KPML, section 5).

Finally, the generated text is edited by the user interface to improve the final output. Our system interface is prepared to control different ways of text formatting (including tables and graphics) and, in the near future, it will incorporate certain aspects of multimodal presentation (hypertext, 3D drawings, speech output, etc.). With this planned flexibility in presentation, it is essential that we adopt an adequate approach to information from the outset: this is a further motivation for our adoption of full natural language generation techniques.

The current prototype of Ontogeneration works as an interactive information retrieval system: the user can consult the domain ontology contents by building an appropriate query with the interface menu options and the system replies generating a Spanish text with the requested information. Different kinds of information and data can be retrieved from our domain ontology: definitions, classifications, examples, comparisons or complete descriptions of chemical elements.

The following example shows a real interaction:

User: "Dame toda la información disponible sobre la propiedad Densidad."

(User: "Give me a complete description of the property named Density.")

Ontogeneration: "La densidad a 20ºC se mide en gr / cm3 y toma valores entre 0 y 25 gr / cm3 . Además sólo puede tener un único valor para cada elemento. Esta propiedad depende del peso atómico y del volumen atómico según la fórmula:
Densidad a 20ºC = Peso atómico / Volumen atómico a 20ºC."

(Ontogeneration: "Density at 20ºC is measured in gr / cm3 and takes values between 0 and 25 gr / cm3. Besides, it can only have an unique value for each element. This property depends on the atomic weight and the atomic volume according to the following formula:
Density at 20ºC = Atomic weight / Atomic volume at 20ºC.")

3 THE DOMAIN ONTOLOGY: CHEMICALS

As we said before, Ontogeneration uses a well-defined domain ontology built for some non-linguistic purposes as a representational resource for text generation. This ontology is called CHEMICALS[4] [12]. CHEMICALS is a domain ontology developed under the METHONTOLOGY framework [16] and using the Ontology Design Environment [7]. It is composed of two ontologies: chemical elements and chemical crystals. Chemical-elements have 16 concepts, 103 instances, 3 functions, 21 relations and 27 axioms. Chemical-crystals has 19 concepts, 66 instances, 8 relations and 26 axioms. The ontologies are formalized in Ontolingua [19], SFK [14] and in a relational data base.

The METHONTOLOGY framework enables the specification of ontologies at the knowledge level [33] and includes: the identification of the ontology development process that refers to which tasks should be done when building ontologies (planning, controlling, quality assurance, specification, knowledge acquisition, conceptualization , formalization, implementation, evaluation, maintenance, documentation and configuration management); a proposal of a life cycle based on evolving prototypes that identifies the set of stages through which the ontology moves during its lifetime; and the methodology itself which specifies the steps to be taken to perform each activity; the techniques used, the products to be output and the way to be evaluated.

The main phase in the ontology development process under the METHONTOLOGY approach is the conceptualization phase. Its aims are: to organize and structure the acquired knowledge (concepts, instances, axioms, relations, functions, attributes, constants, etc.) in a complete and consistent knowledge model using external representations independent of the implementation languages and environments. As a result of this activity, the domain vocabulary is identified.

To build CHEMICALS, the knowledge structuring process can be described as follows:

a) We build a Glossary of Terms that includes all the terms (concepts, instances, attributes, verbs, etc.) of the domain and their description in natural language.
b) When the glossary of terms contain a sizeable number of terms, we structure the domain knowledge in Concept Classifications Trees following a hierarchical organizational principle. This principle aims at splitting up the domain knowledge in as many independent modules or taxonomies as possible in which inheritance can be applied. Each taxonomy will produce a sub-ontology.
c) To model binary "ad-hoc" relationships between concepts of the domain, we build "Ad -hoc" Binary Relation Diagrams between concepts of different concept classification trees or even inside a concept classification tree. Note that this diagram will set out the guidelines for integrating ontologies, because if concepts C1(source) and C2 (target) are linked by a relation R, this means that the sub-ontology containing C1 includes the sub-ontology containing C2, provided that C1 and C2 are in different concept classification trees.

For each concept classification tree generated we build the following intermediate representations:

- A Concept Dictionary, containing all the domain concepts, instances of such concepts, classes and instance attributes and optionally concepts, synonyms and acronyms.
- A Table of Binary Relations for each "ad-hoc" binary relation whose source is in the concept classification tree.
- An Instance Attribute Table for each instance attribute that appears in the concept dictionary. Instance attributes are those defined in the concept but that take values in the instances.
- A Class Attribute Table for each class attribute that appears in the concept dictionary. This kind of attribute describes the concept itself, not its instances.
- A Logical Axiom Table for defining the concepts by means of logical expressions that are always true.
- A Constant Table , for each constant identified in the domain.
- A Formula Table for each formula used to infer numerical instance attribute values from the values taken by other instance attributes, class attributes or even constants.
- Attribute Classification Trees to graphically display attributes and constants related in the inference sequence of the root attributes, as well as the sequence of formulae to be executed in order to infer the root attributes.
- An Instance Table to gather information about the domain instances.

[4] CHEMICALS is available at http://www.ksl.standford.edu:5915 and its mirror site at http://www.ksl-svc-lia.dia.fi.upm.es:5915

Figure 2 shows how the following knowledge is specified using METHONTOLOGY notation: "Halogens also named group VIIa are characterized by being non-metal and reactive elements. Their melting point is low and the electronegativity and ionization energy is high . Fluorine, Chlorine, Bromine, Iodine and Astatine are elements of the periodic table which belong to this group. Flourine's symbol is F, its atomic number is 9 , its boiling point is -188.14 and its electronegativity is 4.0". Note that in this sentence we have concepts, (elements, halogens), we have instances of the halogen concept (Fluorine, Chlorine, Bromine, Iodine and Astatine), we have instances attributes of all elements (symbol, electronegativity, boiling point) and we can state that the Fluorine fills in this attribute with concrete values (F, 4.0, 188.140), we also have a synonym (group VIIa), etc.

The CHEMICALS conceptualization and its implementation in several formats was supported by a software environment called Ontology Design Environment (ODE). The aim of ODE is to support the ontology maker during the entire life cycle of the ontology development process. Currently, ODE's main advantage is that the ontologist develops the ontology at the knowledge level using a set of intermediate representations independent of the target language in which the ontology will be implemented. ODE multilingual generator module automatically translates the knowledge model into target machine-readable languages like: SQL, SFK and Ontolingua. ODE also includes inverse translators that transform Ontolingua code into our intermediate representations structures by a reverse engineering process described in [7]. So, Ontogeneration architecture is prepared to take as a source of knowledge any Ontolingua ontologies that have been previously transformed into our notation, so this approach would go beyond current ontology viewing tools if one adds inverse translators to ODE architecture.

Since we have the ontology in several target languages (Ontolingua, SFK and SQL database), we would like to mention at this point that we have chosen the SQL implementation and not the Ontolingua implementation for two reasons. First, in order to interact with Ontolingua, we would need to include a GFP module inside [10]. Second, we can get the conceptual model attached to an Ontolingua ontology using ODE inverse translator that transforms Ontolingua code into METHONTOLOGY intermediate representations. Then, we can generate a SQL ontology using our forward translators, as it was described by [7].

We have also proved that the METHONTOLOGY tabular and graphically based notation provides a user-friendly approach for both knowledge acquisition and evaluation by computer scientists and domain-experts who are not knowledge engineers. In particular, our experience shows that:

- Domain experts and human final users do not understand formal ontologies codified in ontology languages at all. For instance, in one set of trials, two environmental experts, a chemical expert and two banking managers (all of them with no computer science background) were unable to understand Ontolingua and LOOM code. So, they could neither validate nor formalize knowledge without an ontologists help.
- The same people, using the Ontology Server [11] browser tools, could completely understand and validate taxonomies, partially understand instances, but could not understand abstract definitions of concepts, relations, functions and axioms. In fact, they did not understand what relations, functions and axioms are.
- From the knowledge acquisition point of view, they were also not able to formalize their knowledge at all.
- Such experts could, however, understand and validate 80% of the METHONTOLOGY intermediate representations.
- From the knowledge acquisition point of view, environmental experts can fill in many of the METHONTOLOGY intermediate representations. They are not able to work well with ontology server browsers to formalize that same knowledge.
- In contrast, Computer Science students (5 groups of 10 people each) with a background in frame-based and first-order logic knowledge representation after taking a 8 hours course on ontologies, could understand the majority of any ontolingua code. However, their understanding of the ontology before taking this course was limited to hierarchies, and a few concepts, instances, relations, functions and axioms.
- Often ontology experts (who are unfamiliar with or simply inexperienced in the languages in which ontologies are coded) may still find it difficult to codify a new ontology because the use of traditional ontology tools still focus too much on implementation issues rather than on questions of design [7].

These experiences support those of others in this area. For example, while ontology servers allow developers to input classes, taxonomies, attributes, relations, functions, axioms in a structured way and can automatically generate formal code, it has nevertheless been found with, e.g., the Ontology Server that generates Ontolingua code, that: "Although the ontology editor helps, many people may have experienced that building an ontology from scratch in Ontolingua is daunting, not in the last place because of slow network connections. Experience has shown that the Ontolingua editor is better suited for checking, mantaining and modifying the ontology that for building an ontology from scratch. Therefore, an alternative strategy is to build ontologies off-line, and then import them into Ontolingua. However, writing Ontolingua code is not a comfortable level for persons to work with, that is, it is too close to the symbol level." [8].

Therefore, to sum up, we can say that final end-users of ontologies (domain experts, computer science people in general and ontologists) generally require access to all the details of the representation and not just to the subsumption structure. This information is not well presented graphically in general, although particular methods might be found appropriate in particular domains. Natural language offers a generally applicable way of presenting this more complex information. In addition, the ability of natural language generation technology to flexibly describe and contrast collections of concepts provides a significant increase in viewing functionality. When this is further focused by a user model and more sophisticated text planning, we believe that the Ontogeneration architecture will provide a new degree of usability for ontologies in general.

Now that we have shown how CHEMICALS was developed, our current Ontogeneration work is concerned with demonstrating that:

- We can generate sentences using domain-ontology conceptual models as a source of information.
- The conceptual models built using METHONTOLOGY intermediate representations provide enough knowledge to be used in text generation.
- The text generated automatically can be used by domain experts to evaluate the domain ontology conceptual model.

- Text generation can also be used to build up semi-automatically internal documentation of the ontology.
- Domain ontologies and linguistic ontologies can be successfully merged.

This will be the major concern of the rest of this paper, considering the components individually and then summarizing their interaction.

Concept Name	Synonyms	Acronyms	Instances	Class Attributes	Instance Attributes	Relations
Halogen	Group VIIa	--	Astatine Bromine Chlorine Fluorine Iodine	--	--	--
...	...	...	...	...	...	...

Element
- **Reactiveness**
  - **Non metal**
    - **Halogen**
  - Semi metal
  - Metal
    - Transition metal
      - First transition series
      - Second transition series
      - Third transition series
        - Lanthanide
      - Actinide
    - Non transition metal
      - Alkali
      - Alkaline terreum

Axiom Name	Low melting point of halogens
Description	The highest melting point for halogens is 302 °C
Concept	Halogen
Referred Attributes	Melting point
Variables	H, M
Expression	Forall(H, M) Halogen(H) and Melting-Point(H, M) => M ≤ 302 * Degree-Celsius
Relations	--
References	[Handbook, 84-85]

Axiom Name	High electronegativity of halogens
Description	Electronegativity of halogens is higher than 2.1
Concept	Halogen
Referred Attributes	Electronegativity
Variables	H, E
Expression	Forall(H, E) Halogen(H) and Electronegativity(H, E) => E > 2.1
Relations	--
References	[Janssen, 90]

Axiom Name	High ionization energy of halogens
Description	The first ionization energy of halogens is higher than 10.4 electronvolt
Concept	Halogen
Referred Attributes	Ionization-Energies
Variables	H, E
Expression	Forall(H, E) Halogen(H) and Ionization-Energies(H, E) => nth(E, 1) > 10.4 * Electronvolt
Relations	--
References	[Janssen, 90]

Instance	Attribute	Value
Fluorine	Atomic number Boiling-point Electronegativity Symbol	9 -188.14 4.0 F ...
...	...	...

Figure 2. Example of Chemical Intermediate Representations

4 THE LINGUISTIC ONTOLOGY: GUM

In this section we present how we reused a linguistic ontology and how we adapted it into Spanish. The linguistic ontology used in our application is the Generalized Upper Model (GUM). GUM [1] is used to simplify the interface between domain knowledge and linguistic components. It is an abstract linguistically motivated ontology already used in other generation projects for different languages: particularly English, German and Italian. GUM offers a classification of the kind of meanings that grammatical constructions presuppose. Thus, it plays a main role providing semantics for domain concepts and connecting conceptual representations with lexical representations.

4.1 A general overview of GUM

GUM is a linguistic ontology with a level of abstraction that is halfway between linguistic realizations and "conceptual" or "contextual" representations. That is, it enables abstraction beyond the particular details of lexicogrammatical representations, while maintaining contact with the linguistic realizations so as to support operationalization and interfacing with natural language components. One of the main characteristics of GUM is its generality, which comes from the origin of its motivations: the lexico-grammatical systems of natural languages. The fact that it is a linguistically motivated ontology implies that it is bound to the semantics of a grammar and not to the possibly domain-transcendent general knowledge.

GUM is organized as two hierarchies: one of concepts and one of relations. The concept hierarchy represents the basic semantic entities entailed by natural language grammars, including: process configurations and the different classes of objects and qualities. The relation hierarchy represents the participants and circumstances involved in the processes and the logical combinations among them.

There are two main reasons why we have chosen the GUM ontology as a basis to develop Ontogeneration. First[5], previous work using GUM has shown that it can provide a solid basis for providing natural language generation capabilities where domain organization is insulated from the details of its linguistic realization [5] . Using GUM as an interface level therefore ensures that we do not have to import linguistically-motivated distinctions into our domain ontology in order to support natural language generation. This would compromise the domain model considerably and is generally recognized to be a violation of the desirable modularies of a complete system (cf., for example, [26] critique of such a violation in the LILOG project). The second reason is that previous work on developing multilingual linguistic resources for natural language generation has shown that such work can be significantly speeded if the linking mappings that are necessary between semantic representations and grammatical form can be largely reused. GUM allows this by providing a fixed anchor that is sufficiently general as to require only minor variations across languages. It is not necessary to adopt an interlingual position, but it is still possible to minimize the language-specific idiosyncratic aspects of the semantic description.

This reusability of GUM is one of its prime design motivations, as set out in, e.g., Bateman et al (1995). Both of these reasons strongly support the reuse of GUM in the current system and allow for the reuse of significant bodies of information, both at the semantic and grammatical levels of description needs for providing natural language generation capabilities. The investigation of the applicability of GUM for supporting generation in languages such as English, German, Dutch, French and Italian[6] certainly suggested that the move to Spanish would be likely to succeed. Certainly the demonstration that a Spanish generator can be built would imply a great influence in the Spanish-speaking world

[5] GUM has been used in other applications: Penman [4], Pangloss [25], KOMET [2], TechDoc [34], AlFresco [36] and GIST [15].

4.2 Method to adapt GUM to Spanish

The main criteria followed to adapt GUM to Spanish so that its concepts and relations can be reused as much as possible, are:

- To consider the distinctions in its lexico-grammatical expressions and capture the differences in the "experiential" meaning [4].
- To classify the pattern categories in dimensions, partitions, disjunctions and simple specializations [1].
- To include in the Spanish model the configurations with different number of participants and circumstances in different conceptual representations [1]. These participants and circumstances can or cannot appear explicitly.
- To detect patterns in the Spanish model where the syntactic variations referring to the order in which the arguments appear is not relevant, but it is relevant when the difference of patterns produces a change of meaning [1].
- To permit generating a linguistic realization from different semantic perspectives according to criterion 5 of [21].
- To maximize the cohesion between GUM and the Spanish model and thus minimize the changes in a future multilingual integration.

Every category of the concept and relation hierarchies aforementioned has been studied thoroughly. Then, the linguistic behaviours of Spanish compared to those captured in every category have been studied. If there is any discrepancy, the corresponding extensions are proposed, following the design criteria already exposed. A detailed explanation of this work can be found at [6].

To construct such extensions, the types of descriptive texts to be generated have been analyzed, and the GUM categories, within which their different components can be classified, have also been considered. When some of the categories for the Spanish model do not fit in any of the GUM categories, or a further specialization of an abstract category already existing is required, then we create the categories required to represent such kind of knowledge.

5 GENERATION TEXT ENVIRONMENT:KPML

As the third support of this project, we have reused the KPML (Komet-Penman Multilingual) development environment to build grammatical resources for Spanish. KPML is a system for building and maintaining multilingual linguistic resources and for using these resources in text generation (currently English, German, French and Italian). It substitutes and extends the functionality of Penman's generation system [4] for development supports and multilingual design.

5.1 Aims of KPML

By using KPML we try to simplify the generation tasks and improve the access and handling of the resources. We have chosen KPML because:

- It offers linguistic resources already tested and verified to large-scale generation projects and facilitates standardized input and output specifications suitable for practical generation.
- It offers the generation projects a basic engine for using these resources.
- It encourages the development of similarly structured resources for languages which do not have those resources.
- It minimizes the costs of providing texts in multiple languages.
- It allows us to develop more complex projects reusing other domain ontologies that we have already developed as well as to include Spanish resources into the multilingual environment.

KPML´s grammars, whose basic units consist of grammatical systems, choosers, inquiries, lexical items, punctuation rules and input specifications (SPL, Specification Planning Language [24]]), are system networks defined in systemic-functional linguistics and built as trees of communication options.

KPML´s generation engine [32] uses the system network to construct strings by traversing the system network from left to right for each grammatical constituent to be generated. In each grammatical system only one grammatical feature is selected. Each selected feature may bring a set of syntactic constraints to bear on the overall syntactic structure being generated.

Generation is complete when the structures constructed are sufficiently developed to allow the insertion of lexical items (which may have been chosen at any time during the generation process).

The selection of a grammatical feature in a grammatical system is determined by a chooser for that system. The chooser makes its selection by traversing a decision tree of semantic inquiries.

5.2 Development of Spanish Grammar

KPML can be used with many languages since it permits that any system with a specific name may specialize in a different way depending on the language to be used. By using information of a given system, KPML can analyze the differences and similitudes between systems defined in various languages. KPML permits building up large-scale sets of linguistic resources for different languages either from scratch or by using the resources developed for others. In this project the second via has been chosen. In the process followed to generate texts in Spanish the basic items of English have been taken as a basis to carry out the following tasks:

1. First, a representative set of texts to be generated in the chemicals domain was identified.
2. Then, the English grammatical resources have been studied and the most representative SPL that have correspondence with the possible texts to generate in Spanish have been selected.
3. We have worked with a reduced and representative SPL set for Spanish, obtained by adapting the English SPL or by developing them directly for Spanish when there were not similar English examples. However, when adaptation was carried out, structural, lexical and semantic changes have been done in grammar.
4. Once the set of new resources has been debugged, this may be remerged with the general multilingual resource set if required. This fact permits adding new languages to the generation system.

During development, we found that the range of changes necessary for the extension to Spanish were quite limited. This generally involved adding systems (choice points in the grammatical classification hierarchy), removing systems, adding and removing constraints on structure, etc. All operations that are now well known to be effective for achieving coverage of a new language within the general KPML methodology.

An example of a slighty different kind of change is the following, where we have provided a different approach to a phenomenon than that provided by the English grammar. This concerns the gender and number agreement between names and the determiners accompanying them. The English grammar adapted by us deals with various types ofdeterminers. If we had specialized the systems corresponding to those determiners, we would have had to add two systems, at least, for each determiner. That is why we have chosen another solution. For each system feature corresponding to a determiner we have included a lexical feature to the function “Deictic”. This feature relates the

```
(LEXICAL-ITEM
   :NAME   THISDET
   :SPELLING   "this"
   :SAMPLE-SENTENCE   "this chicken is speckled"
   :FEATURES   (NUMBER DEICTIC NOT-POSSESSIVEDETERMINER DETERMINER)
   :PROPERTIES   ((NUMBER SINGULAR UNCOUNTABLE))
)


(LEXICAL-ITEM
   :NAME   ESTA
   :SPELLING   "esta"
   :SAMPLE-SENTENCE   "Esta propiedad es diferente"
   :FEATURES   (DEICTIC NOT-POSSESSIVEDETERMINER SINGULAR DEMONSTRATIVE NEAR FEMALE
DETERMINER))
)

(LEXICAL-ITEM
   :NAME   ESTE
   :SPELLING   "este"
   :SAMPLE-SENTENCE   "El número atómico de este elemento es 5"
   :FEATURES   (DEICTIC NOT-POSSESSIVEDETERMINER SINGULAR DEMONSTRATIVE NEAR MALE
DETERMINER)
)
```

Figure 3. Example of the lexical items of a demonstrative determiner

function to the lexemes of the determiners. For instance, to manage the demonstrative determines, the lexical feature “DEMONSTRATIVE” is added in the system feature called “DEMONSTRATIVE-SELECTION”. Besides, in order to make the gender and number agreement, the lexical features “PLURAL” or “SINGULAR”, and “MALE” or “FEMALE” are added to the function “Deictic” during the generation of the name (see figure 3, which represents the Spanish terms, "este" and "esta", equivalent to the English term "this").

In general during the development of the Spanish grammar, we have found that the overall degree of resource sharing and reuse between the original English grammar and the current Spanish grammar is very high. Within this resource, we have the following reuse statistics. Overall, the grammar of Spanish consists of 745 grammatical systems, or choice points, and of these 724 (97%) are shared with those of the originating English resource. These choice points distribute their grammatical information over a total of 1345 grammatical features for both languages. Of these grammatical features, only 43 (3%) have so far needed to have their associated structural realization constraints altered so as to produce Spanish sentence structures rather than English. The new areas of grammar developed particularly for Spanish (consisting of 21 grammatical systems) involved the addition of 35 grammatical features: these are, predictably, mostly in the areas where Spanish differs noticeably from English, e.g., lexical gender and number agreement. There are also areas, however, where the grammar has been extended in coverage and these extensions could also be applied back to the originating English grammar (e.g., in the treatment of “relational” clauses and some types of nominalizations). The additions for the mapping between semantics and grammar are also very constrained: 11 new choosers have been developed for Spanish (out of a total of 443). These results demonstrate that the provision of significant generation capabilities for Spanish is indeed possible with high reuse of the previously existing generation resources available with KPML. Most effort was required to extend the grammar in areas that are not found in English and to construct the lexical resources necessary for our domain.

6 CONCLUSIONS: MAIN RESULTS AND BENEFITS

The first result of our work has been the development of a system, Ontogeneration, that is capable of generating texts from computable code stored in an ontology by reusing a linguistic ontology and other resources (already used in different projects and languages).

The current prototype runs on a Unix workstation (Solaris 2.5). KPML needs a Common Lisp and Loom ; in particular we have used Liquid Common Lisp 5.0 (Lucid Compilant) and Loom 2.1. The user interface is implemented in Java.

We want to point out now the main benefits of our system. Our approach has several advantages that we should take into account in the following fields:

a) Ontological engineering:

- Evaluation: natural language texts can be used by domain-expert both to evaluate the domain ontology and the conceptual model. Also, final texts can be useful to measure quickly the quality/quantity of knowledge represented in the ontology.
- Documentation: text generation can be viewed as a first step for building semi-automatically documentation about the domain ontology and its development process.
- Increasing shared understanding in a specific domain: the generation of texts from ontologies is one of the best ways to make available the knowledge of a domain to non-expert users or unfamiliar with ontologies.

b) Text generation:

- Domain ontologies as knowledge source: Chemicals is a well organized domain model that can be used as source of knowledge. METHONTOLOGY framework permits to build domain ontologies with a standard and well defined structure and organization. This methodology offers a generic solution to reuse domain ontologies as appropriate source for text generation systems.
- Easing information retrieval: the generated texts cover the whole knowledge included in the Chemicals ontology. Users can ask and retrieve in their own language different kinds of information: concept and instance definitions, descriptions of concept and instance properties, relation among concepts, comparisons between instances, etc. Although the chemicals ontology only includes 35 concepts and 169 instances, our tests demonstrate that text generation works better than browsers or viewers/graphical tools when the user is non-expert and wants to retrieve and understand all the information quickly.

c) Sharing and reusing knowledge:

- Domain ontologies and linguistic ontologies can be successfully merged. Our system integrates heterogeneous resources, like KPML, GUM and Chemicals, that have already been used in other projects.
- The possibility to reuse the current prototype components in future system extensions is a clear benefit and the key issue of this paper. The modularity of our design allows progressive expansions in depth and width. Other future goals are to share resources with other applications from other fields in order to extent the functionalities and possible uses of the system in different ways (see the next section).

To sum up, we have presented Ontogeneration, an information retrieval system that uses domain and linguistic ontologies for natural language generation in Spanish. Our system is only an example of a generic approach that links two different fields (Ontological Engineering and Text Generation) and whose main goal is to facilitate knowledge sharing and inter-operability between independently developed resources.

7 FUTURE WORK: SCALING-UP

The future work on Ontogeneration implies two kinds of extensions: concrete improvements (to be realized in a very short term) and generic ones (extensions that require a longer time). Some of the concrete improvements that will be added soon are:

- To prepare a version of Ontogeneration that can be consulted on-line via web. Also a new version will run on PC's.
- To make deeper extensions both of the Spanish GUM, including subtleties and specializations of abstract categories, and the Spanish grammar, covering complex linguistic

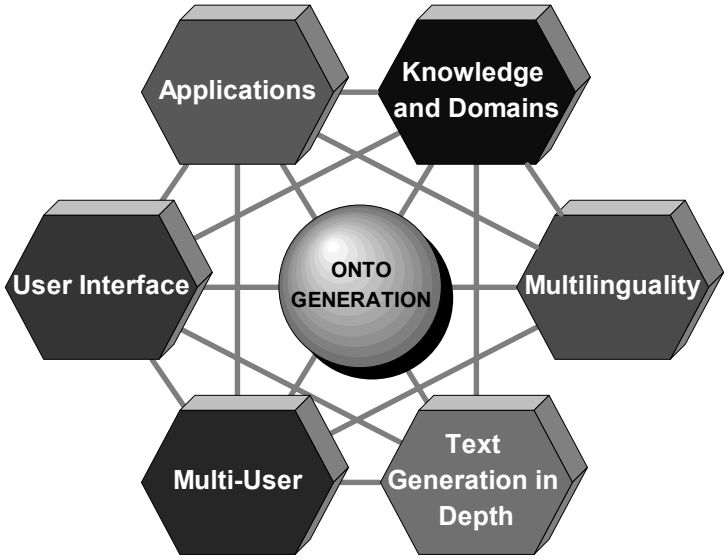

Figure 4. Future extensions of ONTOGENERATION

phenomena related to sentence planning and the discourse structure.

Generic extensions are related with the main goal of our approach. As we have claimed in this paper, the current system is only a starting point. Its development tries to prove an ambitious idea: the possibility of a continous scaling-up in many directions by reusing and extending resources without the need to begin from scracht. This approach will permit to extend functionalities simultaneously in various dimensions (Figure 4):

- Multilinguality. To offer multilingual generation of texts is one of the main advantages of KPML. This tool minimizes the effort of developing and reusing grammatical resources for different languages. We have already done some testing with English successfully.
- Knowledge & Domains. To add other domain ontologies different from Chemical, but with a similar structure (as, for example, other closed scientific taxonomies) to minimize the changes in grammar, which would be almost exclusively lexical. The METHONTOLOGY framework provides the basis to build structured ontologies.
- Multi-user. To extend the user models attending all the possible and significative variations: age (adults, children), expertise (experts or ontology developers vs. non-experts), background o previous knowledge of the domain, etc.
- Deep generation. In order to build a generation system which takes into account the variety of users, language registers and domains, it is crucial to tackle deep generation. Its goal is to produce specifications of fine granularity and degree of linguistic abstraction to drive surface generation.
- Multimodality. To develop a multimodal user interface which permits different modalities of input / output and interaction. Final texts could be combined with multimedia elements: hypertext, graphs, 3-D drawings, video, etc. By using a natural language interface the user could type or voice directly the queries in his own language.
- Applications. Text generation from ontologies can be reused in different applications such as intelligent tutors, knowledge-based systems, data bases access, multilingual information retrieval on-line, machine translation , etc.

8 ACKNOWLEDGEMENTS

This work is supported by the program "Ayudas de I+D para grupos potencialmente competitivos" of the Universidad Politécnica de Madrid (reference A9706).

9 REFERENCES

[1] J. A. Bateman, B. Magnini and G. Fabris. “The Generalized Upper Model Knowledge Base: Organization and Use”. Towards Very Large Knowledge Bases, pp. 60-72, IOS Press. 1995.

[2] J. A. Bateman. “KPML: The KOMET-Penman (Multilingual) Development Environment. Technical Report, GMD/IPSI, Darmstadt (Germany), 1994.

[3] J. A. Bateman, B. Magnini and F. Rinaldi. “The Generalized {Italian, German, English} Upper Model”. Proceedings of the ECAI’94, 1994.

[4] J. A. Bateman, R. T. Kasper, J. D. Moore and R. A. Whitney. “A General Organization of Knowledge for Natural Language Processing: the Penman Upper Model”. Technical Report, USC/ISI, Marina del Rey, CA (USA), 1990.

[5] J. A. Bateman and E. Teich. “Selective information presentation in an integrated publication system: an application of genre-driven text generation”. Information Processing and Management Vol. 31 No. 5 pp. 753-768., Elsevier Science Ltd.,1995

[6] S. Bernardos. "GUME: Extensión de la Ontología GUM para el Español". Facultad de Informática. Universidad Politécnica de Madrid. 1997.

[7] M. Blázquez, M. Fernández, J. M. García-Pinar and A. Gómez-Pérez. “Building Ontologies at the Knowledge Level using the Ontology Desing Environment”. KAW’98. Banf (Canada), 1998.
[8] R. Benjamins and D. Fensel "Community is Knowledge!" in $(KA)^2$, Knowledge Acquisition Workshop, KWA98, Bauff (Canada), 1998.
[9] D. Brill. *Loom Reference Manual Version 2.0*. University of Southern CA (USA), 1993.
[10] V. Chaudhri, A. Farquhar, R. Fikes, P. Karp and J. Rise "The Generic Frame Protocol 2.0", 1997.
[11] A. Farquhar, R. Fikes, W. Pratt and J. Rice. “Collaborative Ontology Construction for Information Integration”. Technical Report KSL-95-69. Knowledge Systems Laboratory, Standford University, CA (USA), 1995.
[12] M. Fernández. "Chemicals: Una Ontología de Elementos Químicos". Facultad de Informática. Universidad Politécnica de Madrid. 1996.
[13] M. Fernández, A. Gómez-Pérez and N. Juristo. "METHONTOLOGY: From Ontological Art Towards Ontological Engineering", AAAI-97 Spring Symposium Series on Ontological Engineering, Standford University, CA (USA), 1997.
[14] D. Fischer and L. Rostek. “SFK: A Smalltalk Frame Kit”, Technical Report, GMD/IPSI, Darmstadt (Germany), 1993.
[15] GIST: “Generating InStructional Text. Final Report”. Technical Report, IRST, Trento (Italy). 1996.
[16] A. Gómez-Pérez. “Knowledge Sharing and Reuse”. *Handbook of Applied Expert Systems*, edited by Liebowitz, CRC, 1998.
[17] T.R. Gruber. "ONTOLINGUA: A Mechanism to Support Portable Ontologies", KSL-91-66, Knowledge Systems Laboratory, Stanford University, Stanford (USA), 1992.
[18] T.R. Gruber. “Towards Principles for the Desing of Ontologies Used for Knowledge Sharing”. Workshop on Formal Ontologies, Padua (Italy), 1993.
[19] T.R. Gruber. “Ontolingua: A Translation Approach to Portable Ontology Specifications”. Knowledge Acquisition. Vol. 5, pp. 199-220, 1993.
[20] M. A. K. Halliday. *An Introduction to Functional Grammar*. Edward Arnold, London (UK), 1985.
[21] R. Henschel. ”Merging the English and the German Upper Model”, Technical Report. GMD/IPSI, Darmstadt (Germany), 1993.
[22] E. Hovy. “Creating a Large Ontology”. ANSI Ad Hoc Group on Ontology, Standford University, September, 1996.
[23] E. Hovy and S. Niremburg. “Approximating an Interlingua in a Principled Way” Proceedings of the DARPA Speech and Natural Language Workshop. Arden House, New York, 1992.
[24] R. T. Kasper. "A flexible interface for linking applications to PENMAN's sentence generator", Proceedings of the DARPA Workshop on Speech and Natural Language", 1989.
[25] K. Knight and S. K. Luk. “Building a Large-Scale Knowledge Base for Machine Translation” Proceedings of the American Association of Artificial Intelligence Conference AAAI-94, Seattle (USA), 1994.
[26] E. Lang. "The ontology lilog from a linguistic point of view". *Text understanding in lilog: integrating computational linguistics and artificial intelligence. Final report on the IBM Germany lilog-Project,* Springer-Verlag, pp. 464-481, 1991.
[27] D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Addison-Wesley Publishing Company Inc. 1989.
[28] D. B. Lenat, R. V. Guha, K. Pittman, D. Pratt and M. Shepherd. “Cyc: Towards Programs With Common Sense” Communications of the ACM. 1990.
[29] “Loom Users Guide Version 1.4”. ISX Corporation, 1991.
[30] B. Magnini, “Specification of Upper Model”, Technical Report Project 062-09 GIST. IRST, 1994.
[31] J. R. Martin, C. M. I. M. Matthiessen and C. Painter. *Working with Functional Grammar*, Arnold, London (UK), 1997.
[32] C. M. I. M. Matthiessen and J. Bateman. *Text Generation and Systemic-Functional Linguistics: Experiences from English and Japanese*. Pinter Publishers, London (UK), 1991.
[33] A. Newell. "The knowledge Level". Artificial Intelligence. pp. 18. 87-127, 1982.
[34] D. Rösner. “Generating Multilingual Documents from a Knowledge Base: The TECHDOC Project”. Technical Report FAW Ulm, Ulm (Germany), 1994.
[35] M. Stede. “Lexical Options in Multilingual Generation from a Knowledge Base”, *Trends in Natural Language Generation: an Artificial Intelligence Perspective*, Springer-Verlag, pp. 222-237, 1996
[36] O. Stock, G. Carenini, F. Cecconi, E. Franconi, A. Lavelli, B.Magnini, F. Pianesi, M. Ponzi, V. Samek-Lodovici and C. Strapparava. “ALFRESCO: Enjoying the Combination of Natural Language Processing and Hypermedia for Information Exploration”. In Mark T. Maybury, editor, Intelligent Multimedia Interfaces, The MIT Press, pp. 197-224, chapter 9. Extended and revisedversion of a paper previously published at IJCAI-91. 1993.
[37] P. E. van der Vet and. N. J. I. Mars. “The Plinius Project and its Ontology” annex in “Ontologies: Principles, Methods and Applications”. Knowledge Engineering Review, Vol. 11, 1996.
[38] M. Uschold and M. Gruninger. "ONTOLOGIES: Principles, Methods and Applications". Knowledge Engineering Review. Vol 11. No. 2., 1996