

7.4 THE MENU SYSTEM

The menu system organizes and updates the information displayed on the screen.

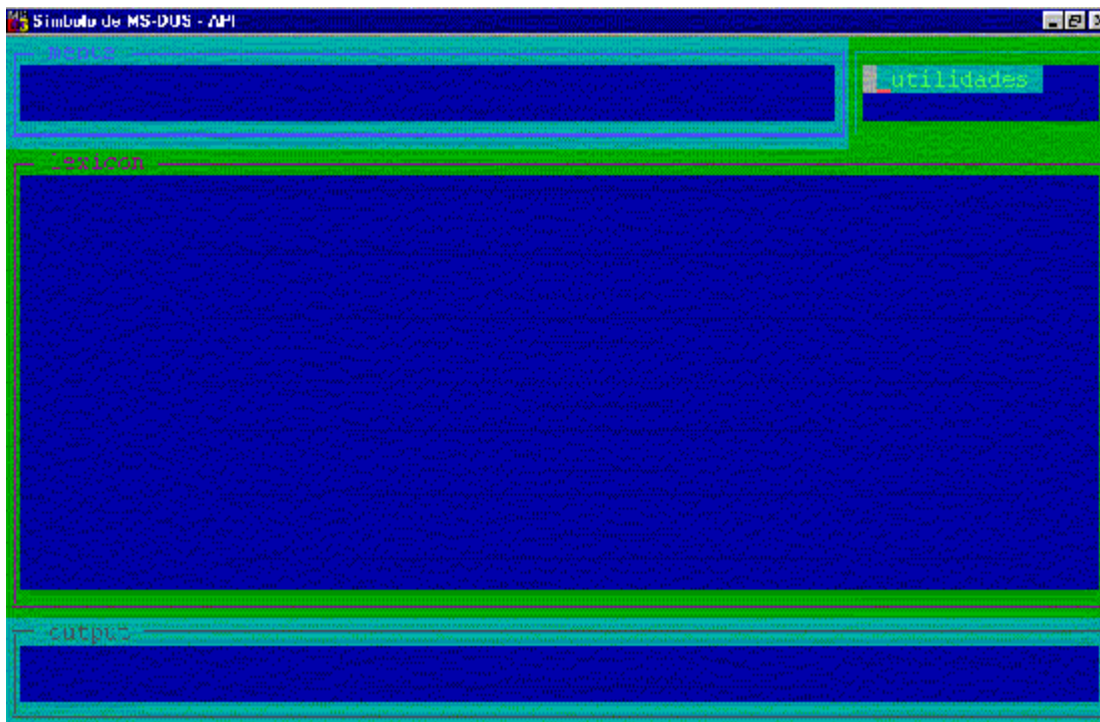


Figure 7.23: A view of the system of menus

The screen is divided into three different sections, as shown in Figure 7.23. The input sentence is shown at the bottom of the screen. In the central section, menus showing all possible entries that the parser may accept are displayed. In the right corner, all utilities incorporated into the NLI are displayed in a menu.

The menu system guides the user to introduce the information required by the application by showing all possible NL options the interface can recognize at each step of the communication process. Each NL option consists of one or more words. At each step, the name of the active menus and the NL options they contain are displayed. The user has to select a menu and one of the following options contained in the menu. Once the user has selected a NL option, the menu system passes it to the parser. When the parser returns all items that can be recognized in the next step, the menu system updates the menus displayed on screen. Only the correct options to continue the sentence are shown.

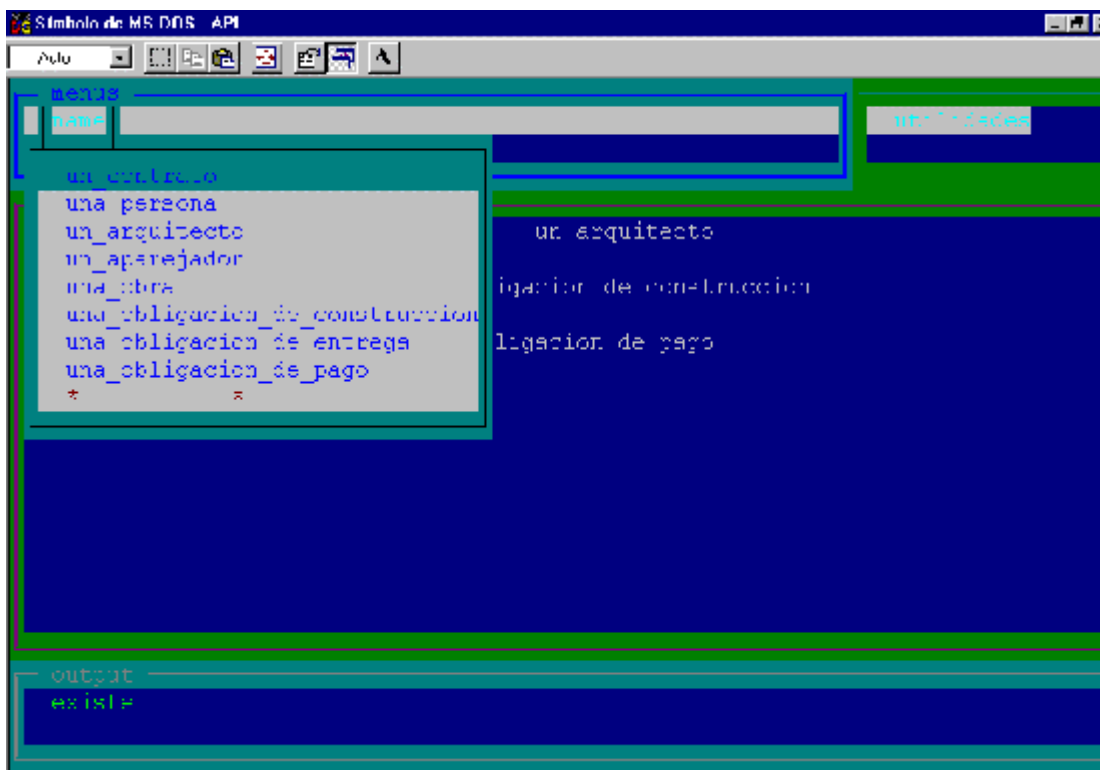


Figure 7.24: Guiding the user to build an NL sentence

The performance of the menu system for an example is shown in Figure 7.24. This example shows how a sentence for creating an instance of the concept **ARCHITECT** is built using the interface generated for SIREDOJ. Figure 7.24 shows all options displayed once the user has introduced the word *existe* (*exists*). These options correspond to the definite nominal groups representing the application concepts from which an instance can be created.

The superficial presentations contained in the lexicon can be distributed in menus as desired. That is, the number of menus and the syntactic category of the NL options these menus can contain can be defined for each grammar generated. By default, GISE provides four different menus. The menu **nombre** containing nominal groups, the menu **verbo** containing verbal groups, the menu **adjetivo** containing adjectival and quantity groups and the menu **preposition/adverb** containing prepositions and adverbs.

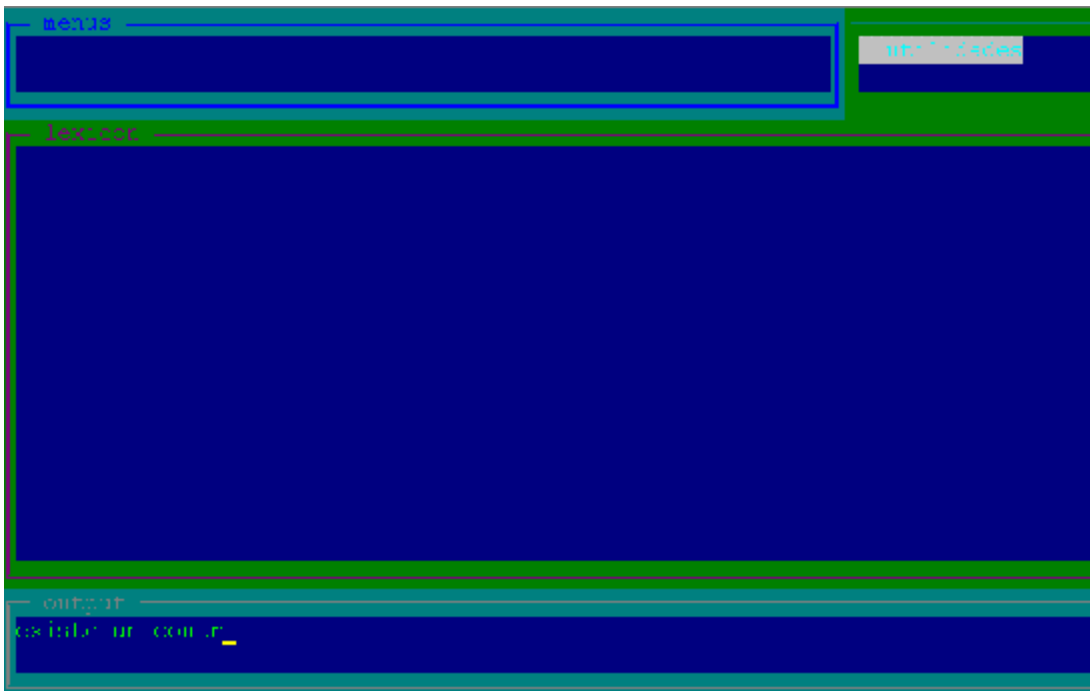


Figure 7.25: Introducing manually a sentence

If the user selects the option corresponding to a dynamic entry, the menu system is responsible for activating the function associated with the entry. As described above, there are three types of dynamic functions. There are functions asking the user to introduce a proper noun or a quantity at run-time. There are functions obtaining all instances of a CO concept and displaying their identifiers on screen. Finally, the function **menu**, associated with the values of conceptual attributes represented by a closed set of options, displays all possible values of one of these attributes in a menu.

Different utilities can be incorporated into the NLI. The incorporation of these utilities can be undertaken by considering the different type of users.

The utilities incorporated in the current prototype for the application SIREDOJ are described below. Each of these utilities corresponds to an option in the menu **utilidades**.

There are several options determining the way sentences are introduced. The option **seguir** (to continue) allows the user to introduce a sentence step by step, having the possibility to modified previous input by selecting the option **anterior** (previous). These options may be useful for casual users. For expert users, there is an option to introduce the sentences faster, the option **continuadamente**. When this option is selected, the menu **utilidades** cannot be accessed until a whole sentence has been introduced.

The option manual allows the user to introduce the NL sentences manually. Figure 7.25 shows an example of how sentences are built manually.

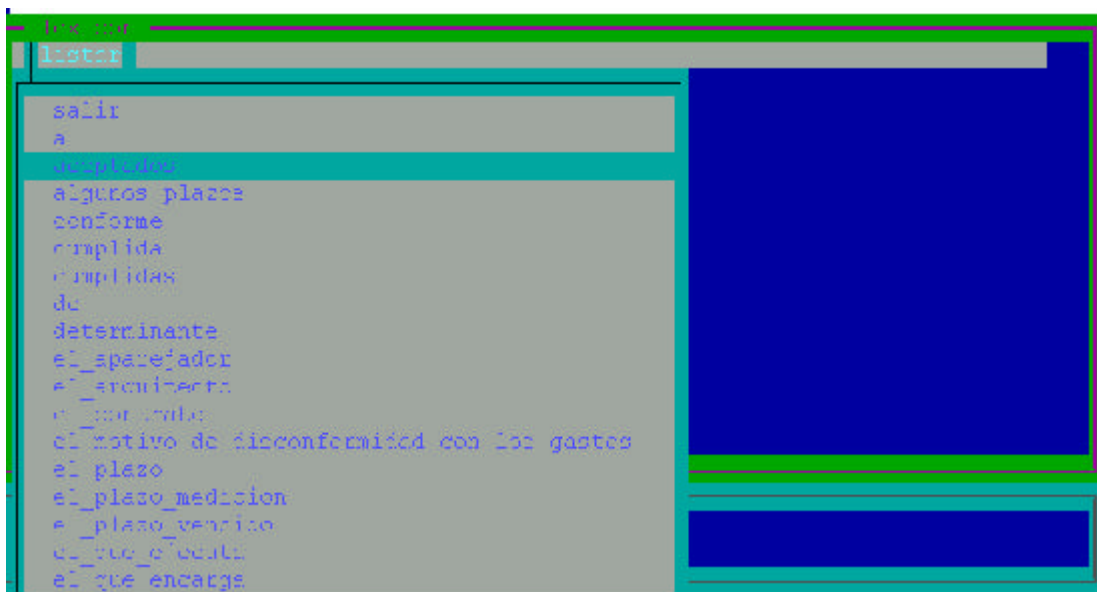


Figure 7.26: Displaying the words contained in the lexicon

The menu **utilidades** includes an option to execute the operation expressed by the last sentence introduced and the option to finish the session.

There is also a utility displaying the interpretation of the last sentence introduced. There is a tool displaying information about all entries contained in the lexicon. Figure 7.26 shows the first words displayed when this option is selected.

7.5 THE DIALOGUE COMPONENT

The Dialogue Component (DC) obtains a set of complete operations over the CO case level from the semantic representation built by the parser. It completes this semantic representation in the context of the ongoing dialogue. Then, it passes the resulting operation to the Communication Manager (CM).

The complexity of this DC is not as high as in such NLI where the user is free to introduce input. The NLIs generated by GISE use application-restricted grammars and guide the user in introducing the correct sentences. The process of obtaining the operation and its arguments from these sentences is simple.

Only when there is less than full use of the system mechanisms controlling the user interventions are used partially can the sentences introduced include mistakes. This is the case when the preconditions attached to grammar rules are not activated, when the user is allowed to introduce the sentences manually, or when the grammar used is not that generated by the basic sets of control rules.

All sentences recognized by the parser express operations consulting or modifying concepts in the CO. The communication, consulting or modifying mode is set before the sentences are introduced.

The information passed to the DC consists of a set of possible semantic interpretations. In current implementation, each semantic interpretation contains the information necessary to execute one or more operations over a concept. One semantic interpretation consists of a list containing sublists, each containing the information necessary to execute one operation. This information consists of a list of four elements: the category in the left-hand part of the rule expressing the operation, the name of the operation, the concept identifier and the operation parameters expressed. The CM obtains the information of the operation parameters not expressed.

A very simple attentional structure is implemented in the current prototype. The concept over which an operation is performed is considered the focus of attention and is stored in a stack. If the parameter representing the concept is not expressed in the next user intervention, it is obtained from the stack. This simple mechanism allows the correct processing of interrogative or declarative clauses referring to a previous one (we call them referential clauses). These clauses express consulting or modifying operations over a concept without giving its name. In this case, the concept identifier

can be obtained from the context. An example of a referential clause accepted is: *La persona que construye es el arquitecto* (*The person who builds is the architect*). This sentence expresses the operation of filling the attribute **who_builds** of an instance of the concept **BUILDING_CONTRACT**. It can only be accepted if a sentence expressing the operation of creating an instance of the concept **BUILDING_CONTRACT** had been previously introduced. Although the name of the concept is not expressed in these referential clauses, it can easily be obtained from the context.

If the grammar was generated by one of the two basic sets of rules, this information is obtained directly during the semantic interpretation from the semantic features associated with the rule categories. In these grammars, the semantic feature associated with the left-hand category of the rule expressing an operation represents the identifier of the concept. In this case, the attentional structure would not be necessary to solve the reference.

There is another specific case in which an operation parameter of the operation is not expressed] in the user intervention, but which can be obtained from the context. This case corresponds to the operation of filling an attribute with the value **yes**. The value **yes** does not appear in the clauses expressing this operation. The DC adds this value to the list of operation parameters.

For example, the clause *La obligación de construir está cumplida* (*the building requirement is fulfilled*) expresses the operation for filling the attribute **fulfilled** of the concept **BUILDING_REQUIREMENT** with the value **yes**. Although the value **yes** is not expressed in this clause, the DC can easily deduce it. Only in the particular case of a clause representing the filling of an attribute with the value **yes**, is the argument representing the value not expressed.

The DC also considers ambiguities in semantic interpretation. If there is more than one semantic interpretation to the user intervention, all these interpretations are checked and only the executable interpretations are passed to the CM. A mechanism asking the user for more information, in the case of more than one semantic interpretation being able to be correctly executed, can easily be incorporated into current implementation.

The DC detects when the user intervention cannot be interpreted correctly as an operation over the CO, and sends an error message to the menu system. As the conditions governing the operation execution are incorporated into the grammar rules, they do not have to be evaluated at this step. The grammar preconditions would be evaluated before operation execution only if they have not been activated.

In the current prototype for GISE, only a simple DC and a simple CM has been implemented to analyze the performance of the NLIs generated by the system. These modules were designed with the assumption that the user would introduce NL sentences by using the menu system. This system ensures that the options introduced are those acceptable to the grammar generated by GISE for a specific application. Dealing with the sentences introduced manually by the user would require an increase in the DC complexity. First, it would require the use of more general grammars, in the grammars generated by GISE, few forms of the same operation are considered because the interface guides the user to express each operation. Furthermore, when users build input without any help, they can also introduce sentences that do not express correct operations, or even ones that do not express any operation at all. To deal with these cases, the functionality of the DC would necessarily have to include new tasks, such as those reformulating or confirming user interventions, opening or/and closing the dialogue, etc.

7.6 THE COMMUNICATION MANAGER

The CM controls the interchange of information between the user and the application. The main function of the CM is performing the execution of the operations over the CO expressed by the user. The DC is also in charge of controlling how the information in the CO is completed when required by the application.

As mentioned above, other interfaces besides the GISE NLI can be also integrated into the whole system, thereby allowing different organizations. The current implemented prototype includes three different possible forms of integrating the NLI into the application. However, other forms could easily be integrated. The three current possible forms of integrating the NLI correspond to three different possible ways of configuring the CM. The performance of the CM is different for each case. Initially the desired configuration must be selected. The three possible architectures resulting when integrating the NLI are described below.

The first possible architecture includes two interfaces. The GISE NLI allows users to introduce whatever information they consider relevant before the application is executed. This information is represented in the CO case level. Application

information requests are passed to the CM. The CM consults the CO case level and if the information is not found it is requested from the user. A second interface is used in asking the user for the specific information required by the application. The user answers questions using the second interface.

This architecture is appropriate when the application has already had an interface integrated into it. The second interface can be very simple, probably menu-based. This interface is used to ask the user very specific information. The resulting architecture once the NLI and the second interface are integrated to the whole system is described in Figure 7.27.

In this first configuration, the NLI initially guides the user to introduce the information. When this process is finished, the NLI is disconnected and the second interface is activated. Then the application is called up. When the information the application needs is not found in the CO, it uses the second interface to ask for it.

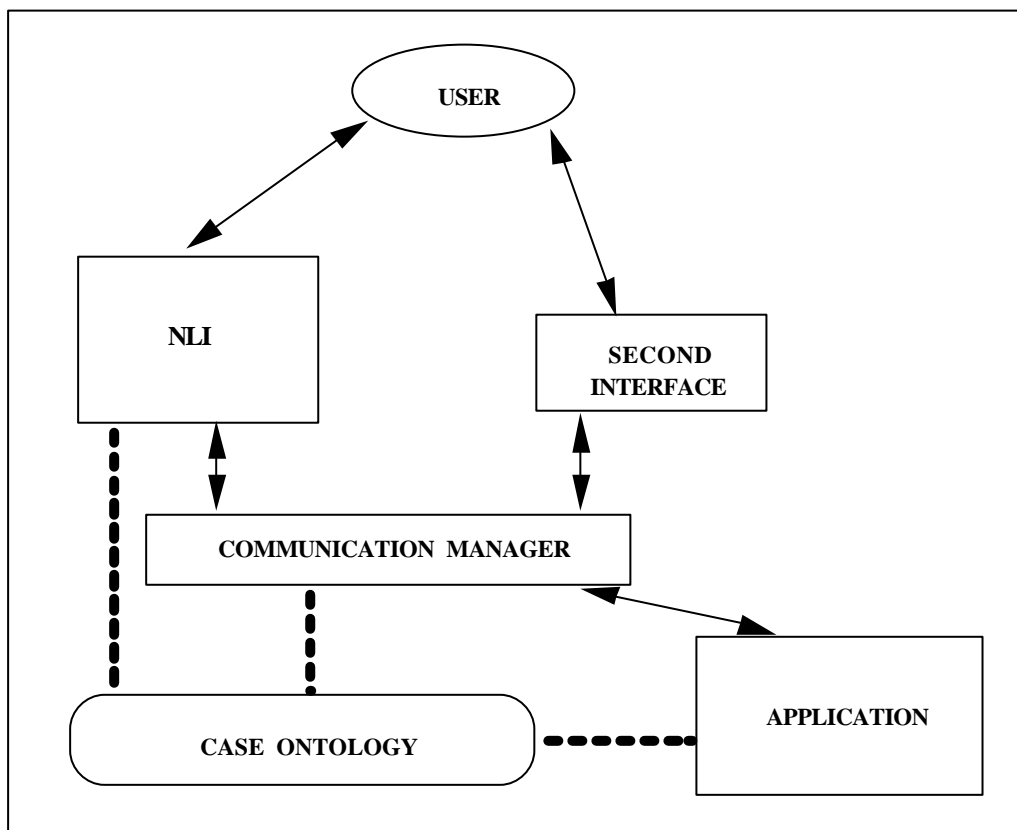


Figure 7.27: A possible architecture showing how the NLI and a second interface are integrated into the whole system

In the two other possible architectures, the GISE NLI controls all the information introduced by the user. Figure 7.28 shows the second possible configuration. In this

configuration, the NLI guides the user in introducing information before the application is executed, as well as during the execution, when more information is required by the application. In this configuration, once the NLI has guided the user to introduce the information, the application is called. When the application needs information about the specific cases introduced by the user, it directly consults the case ontology. If the information required is not there, then the application uses its own output module to request it from the user. In the case of the application not having an output module, a basic module displaying canned text on screen could be used. The user will answer these questions through the NLI. To guide the user in introducing the appropriate answers, the menu system will be used and the preconditions associated with the grammar rules will be evaluated in terms of the new state of the CO case level.

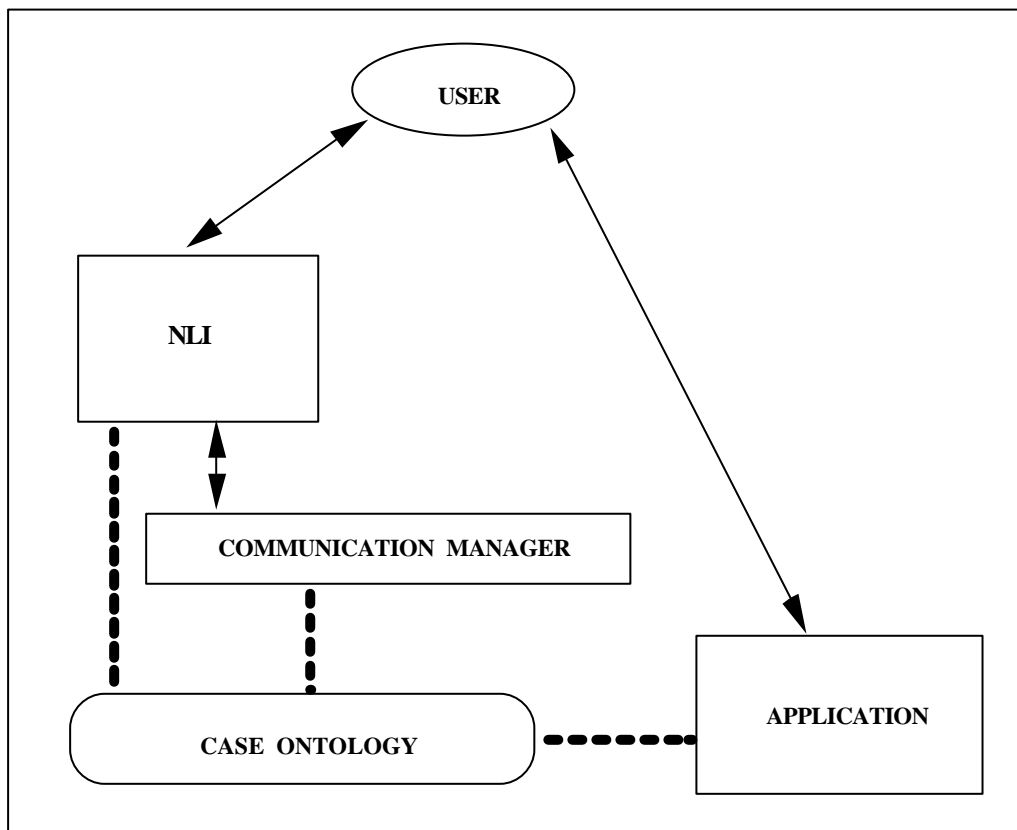


Figure 7.28: A second possible architecture for integrating the NLI

Finally, the third possible form of integrating the NLI configuration considers only the possibility of introducing the information before the application is executed. In such a case, the preconditions associated with the grammar rules must guide the user in introducing all the information required by the application during its execution. Once

the user has introduced this information, it is disconnected and the application is executed. Using this configuration, no new information could then be introduced by the user. The third possible architecture is shown in Figure 7.29.

The third configuration is appropriate for experts on the application, who would be capable of introducing all the information that the application will require from the user before it is executed. This would be the most efficient and friendly mode of communication in the case of the application performance being well known.

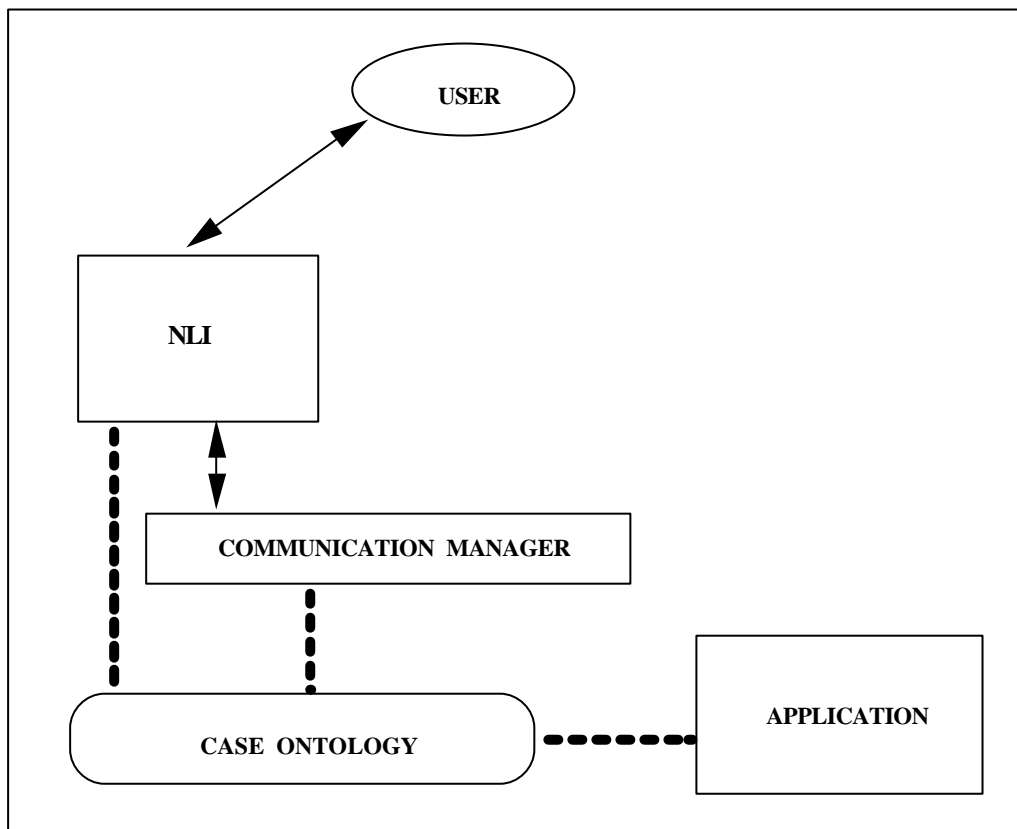


Figure 7.29: A third possible architecture for integrating the NLI

CHAPTER 8

APPLICATIONS OF GISE

8.1 INTRODUCTION

The central point in the design of GISE was to provide a general process for adapting general linguistic resources to express the communication actions required for a specific application. GISE has been designed for supporting the NL communication between the user and the KBS during the exploitation phase. The communication acts in this phase consist of guiding the user to describe specific problems, asking the information required by the application and displaying results. Adapting GISE to support different communication acts would not require major changes. For example, it could be easily adapted to generate expert-interfaces, those supporting the communication between the KBS and the expert during the acquisition phase.

GISE can also be adapted to several types of systems. It can be directly applied in generating NLI to DBSs. It could also be adapted to different types of consulting systems, such as those providing information from different media and WEB consultants. Furthermore, the NLI generated for a specific application could be integrated with other modes of communication, such as graphics and/or speech.

The main core of GISE could be reused for purposes other than that of supporting communication between the application and the user. The problem of adapting general

linguistic resources to a specific application is common to different types of NL systems. The functionality of the system could be enriched to support the generation of explanations, descriptions, summaries and other tasks related to the expression of application knowledge.

In this chapter, two different applications of GISE are described. The first section of this chapter describes the application of GISE to SIREDOJ, an ES in law. This section describes the process of generating the NL-guided interface for supporting communication between the user and SIREDOJ. The second section describes the application of the system to a railway communication system.

8.2 THE APPLICATION OF GISE TO SIREDOJ, AN EXPERT SYSTEM FOR LAW

GISE has been applied to SIREDOJ (Intelligent System for Legal Information Retrieval), an ES for law specializing in building agreements. The basic function of the system is to classify the legal cases introduced by the user, according to the knowledge base of the system, in order to obtain legal conclusions.

In a first release, SIREDOJ was built as a monolithic system with communicative and functional tasks fully integrated. The system requested information when it was needed. All communications were guided by the system, and no user-initiative dialogues were allowed. The communication was based on a telescopic set of menus in which Spanish text appeared. This text consists of canned sentences included in the application code.

The basic problems with the communication strategy used in SIREDOJ were the lack of user initiative, the need for using long menu chains and the poor quality of language appearing in the menus. As was stated in [Martí91], integrating a NLI to the system could solve these problems.

By applying GISE to SIREDOJ and thus obtaining an application-restricted NLI guiding user interventions, major improvements have been achieved in communication. Figure 8.1 compares the communication process in SIREDOJ prior to **(a)** and following **(b)** the application of GISE.

In order to obtain the appropriate NLI for SIREDOJ, the application domain and functionality needed to be incorporated into the system. SIREDOJ already used a

knowledge base to represent its domain knowledge. This knowledge base was adapted to the GISE CO.

The application concepts involved in the communication were represented in the CO application level. The attributes describing these concepts were also represented in the CO. The realizations of the application concepts and attributes were represented in a set.

Once all the application information was provided, the basic set of control rules was applied over the CO to generate the application-restricted NLI.

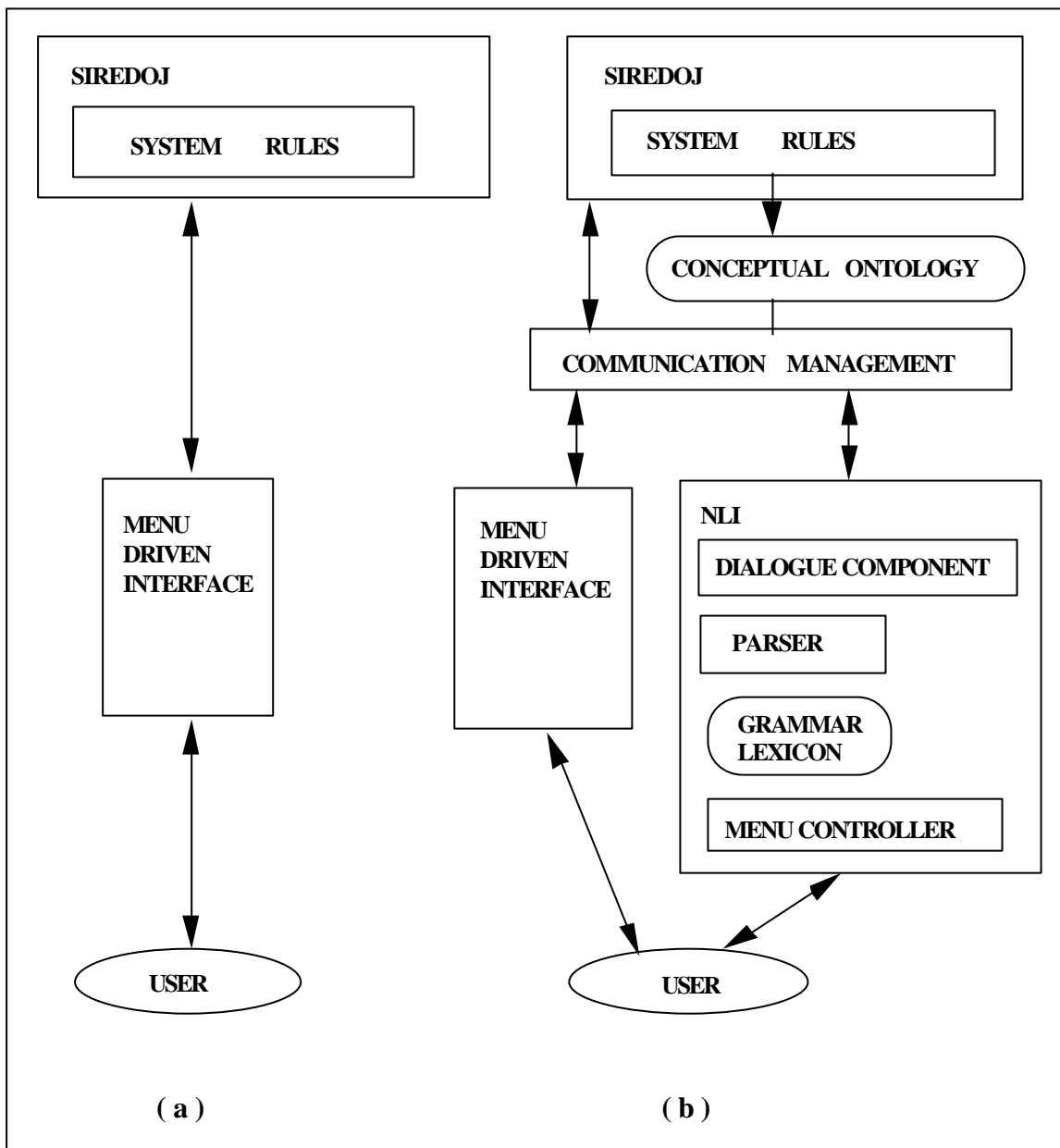


Figure 8.1: Applying GISE to SIREDOJ in order to improve the communication process

The representation of SIREDOJ in the CO as well as the menu-guided NLI generated for this application is described below.

8.2.1 Representation of SIREDOJ domain and functionality

The conceptual knowledge necessary for supporting the communication between the user and SIREDOJ was represented on the basis of the GISE CO. The application concepts involved in communication, as well as the attributes describing these concepts, were defined in the application level. This information was provided following the steps described in Section 3.3.1 in Chapter 3.

First, the description of all the application concepts necessary during the communication acts was provided. The description for each concept consists of the identifier, the **isa** or **instance** relation connecting it to the CO taxonomy of concepts, a set of attribute-values and a set of structural properties or facets. Only predefined facets can be used. Some of these facets are obligatory. For example, each concept description must include the facet **interface_entity** indicating if the concept is used during the communication (its value is **yes**) or if it is not (its value is **no**). Preconditions were also incorporated into some CO concepts.

In the CO application level describing SIREDOJ, 21 new conceptual classes were described, 13 representing interface concepts and 8 internal concepts. The internal concepts are briefly described below:

GLOBAL. This is a concept used by the application to keep trace of the ES rules already applied.

ASSIGNMENT. This is an internal concept representing a general assignment (i.e. a contract or other less specific forms).

CONTRACT. This is subclass of the **ASSIGNMENT** concept. It represents all information about general contracts the application must deal with.

BUILDING_CONTRACT. This is a subclass of **CONTRACT** conceptual class representing the specific type of contracts the ES is specialized in.

ASSIGNMENT_PARTS. This is an internal concept representing all parts involved in an assignment.

CONTRACT_PARTS. This is a subclass of the **ASSIGNMENT_PARTS** class. It represents the parts involved in a general contract.

REQUIREMENT. This is an internal concept used in the definition of the specific obligations involved in a contract.

RULES_ASSIGNMENT. This is concept the application uses for storing the conclusion text.

The internal concept was described by internal attributes. No preconditions were attached to the internal concept descriptions. The interface concepts were described by both internal (not used during the communication) and interface attributes (used during the communication). Preconditions were attached to the definition of most of these concepts. The 13 application concepts used during the communication are described below.

CONTRACT_INFORMATION. This is an interface concept representing relevant information general to any contract. When the user expresses the existence of a contract, an instance of the internal concept class **BUILDING_CONTRACT** is created in the CO case level. Possible linguistic realizations of this concept are *el contrato (the contract)* and *un contrato (a contract)*. It is described by two interface attributes: **isOK** and **contract_type**. The attribute **isOK** has two possible values, **yes** and **no**. Its linguistic realization is *todo en orden (OK)*. It belongs to the **IS** class. The **contract_type** attribute describes the type of contract. Its range is a menu containing two values that would be displayed on screen at run-time: *contrato de obra (building contract)* and *contrato de proyecto (project contract)*. It is realized as *tipo (type)*. It belongs to the class **OF_TYPE**.

BUILDING_CONTRACT_PARTS. This is subclass of **ASSIGNMENT_PARTS**. As its name indicates. It represents the three different parts of a building contract: the person (or company) who assigns, the person (or company) who is assigned and the object assigned. This concept can be realized as the nominal groups *el contrato* and *un contrato* or as the verbs *contratar (to contract)* and *encargar (to assign)*.

Each of the three parts involved in a building contract is represented by a different interface attribute. The attribute **subject1** represents the person who assigns a building. Its range is the concept **PERSON**. Its value must be an instance of this concept. This attribute can be realized as *el que encarga (the person who assigns)*. It belongs to the **WHO_DOES** class. The attribute **subject2** represents the person who is assigned to supervise a construction and to deliver it to the other party. Its value must be an instance of the concept **PERSON**, described below. A possible realization of this attribute is *el que ejecuta (the person who executes)*. It belongs to the **WHO_OBJECT** class. The attribute **object** represents the object to be built. Its value

must be an instance of the concept **BUILDING**, described below. This attribute can be realized as *el objeto* (*the object*) and *un objeto* (*an object*). It belongs to the **WHAT_OBJECT** class. The filling of these three attributes can be expressed in simple transitive clauses.

The definition of the concept **BUILDING_CONTRACT_PARTS** includes a precondition on the existence of an instance of the concept **CONTRACT_INFORMATION**.

The range of the three attributes describing the concept are the concepts **PERSON** and **BUILDING**. These conceptual classes are described below. Neither attributes nor preconditions are included in the description of these concepts.

PERSON. This is an interface concept used in the description of the human parts involved in a contract. Two possible realizations of this concept are *una persona* (*a person*) and *la persona* (*the person*). Three subclasses of this class are included in the application definition. They are: **ARCHITECT**, realized as *arquitecto* (*architect*), **QUANTITY SURVEYOR**, realized as *aparejador* (*quantity_surveyor*) and **OWNER**, realized as *propietario* (*owner*).

ASSIGNED_OBJECT. This concept represents the material party involved in a contract. It can be realized by the nominal groups *un objeto* (*an object*) and *el objeto* (*the object*). Its subclass **BUILDING** is also included in the application description. Two possible realization of this concepts are the nominal groups *una obra* (*a building*) and *la obra* (*the building*).

REQUIREMENTS_EXISTENCE. This is an interface concept describing information about the different types of requirement-specific contract states. Possible linguistic realizations of this concept are the indefinite nominal group *unas obligaciones del contrato* (*A_contract_requirements*) and the definite nominal group *las obligaciones del contrato* (*The_contract_requirements*).

This concept is defined by two interface attributes and three internal attributes. The interface attributes are **included** and **requirements_set**. The attribute **included** describes whether there are requirements in the contract. Its value can be **yes** or **no**. It can be realized as *presentes* (*included*). The attribute **requirements_set** represents all duties included in the contract. Its range is a menu containing the four options: the three possible requirements (to pay, to build and to deliver) and the option **others**. Its

cardinality is multiple, that is, it can have more than one value. One of its possible realizations is *el_conjunto* (*the_set*). It belongs to the **OF** class.

The definition of the concept **REQUIREMENTS_EXISTENCE** includes a precondition on the existence of an instance of the concept **CONTRACT_INFORMATION**.

REQUIREMENTS_FULFILMENT. This is a concept used to represent general information about the fulfilment of an existing contract. It can be realized as *obligaciones_del_contrato*. This concept is described by the interface attribute **fulfilled**, describing whether the requirements have been fulfilled or not. One possible realization of this attribute is realized as *cumplidas* (*fulfilled*). It belongs to the class **IS**. The definition of this concept also includes a precondition on the existence of an instance of the concept **CONTRACT_INFORMATION**.

BUILDING_REQUIREMENT, **DELIVERY_REQUIREMENT** and **PAYMENT_REQUIREMENT**. These three concepts represent all possible obligations involved in a building contract. They are subclasses of the internal concept **REQUIREMENT**. The communication between the application and the main user consists of the description of these three concepts. Each concept is represented by a different set of attributes the user would have to fill. The preconditions associated with these three concepts include a condition on the existence for an instance of the concept **CONTRACT_INFORMATION**, as well as conditions on the filling of the attributes.

Two possible linguistic realizations of the concept **BUILDING_REQUIREMENT** are the definite nominal_group *una_obligacion_de_construccion* (*a_building_requirement*) and the indefinite nominal_group *la_obligacion_de_construccion* (*the_building_requirement*).

This concept is described by three interface attributes: **bfulfilled**, **reasonotbuilt** and **reasonorefuse**. The attribute **bfulfilled** indicates whether the obligation is fulfilled or not. It is realized as *cumplida* (*fulfilled*) and belongs to the class **IS**. The attribute **reasonotbuilt** represents the reason why the obligation has not been fulfilled. Its range is a menu containing all possible causes. It is realized as *los motivos del incumplimiento* (*reasons for non-compliance*). It belongs to the class **OF**. There is a condition governing the filling of this attribute: it can only be filled if the value of the attribute **bfulfilled** is **no**. The attribute **reasonorefuse** describes the reasons for refusing to build. Its range is also a menu containing all possible reason. It is realized

as *los motivos de la negativa a cumplir* (reasons for refusal). It also belongs to the class **OF**.

Two possible linguistic realizations of the concept **DELIVERY_REQUIREMENT** are the definite nominal group *una obligacion de entrega* (a delivery requirement) and the indefinite nominal group *la obligacion de entrega* (the delivery requirement).

This concept is described by the four interface attributes: **dfulfilled**, **reasonotdeliver**, **reasontorefused**, **impossibility** and **finaldeadline**. The three first attributes are almost the same as those representing the **BUILDING_REQUIREMENT** described above. The last two attributes represent additional information to recover in the case of the obligation not being fulfilled. They have as possible values **yes** and **no**. They are compound attributes, that is, they are realized as two constituents. The attribute **impossibility** can be realized by the nominal group *una imposibilidad* (an impossibility) followed by the verb *sobrevenir* (arose). This attribute is incorporated into the **DOES_SUBJECT** class because it belongs to both classes: **DOES** (the attribute is realized as an intransitive verb) and **SUBJECT** (it is also realized as the constituent subject). The attribute **finaldeadline** can be realized by the nominal group *un plazo* (a deadline) followed by the adjective *determinante* (final). It is classified as an **IS_SUBJECT** attribute because it belongs to the class **IS** (it describes the subject) and **SUBJECT** (describes a new subject related to the concept). Preconditions governing the filling of the first two attributes are also included in the concept description.

The concept **PAYMENT_REQUIREMENT** can be realized as *obligacion de pago* (payment requirement). This concept is described by 15 interface attributes. Three of them correspond to the general information also included in the building and delivery requirement description (**pfulfilled**, **reasonotpaid**, **reasontorefusep**). The remaining attributes describe details concerning the type of payment, the number of payments, the payments that have been fulfilled, justification of the expenses and their measurement. The range of all these attributes is a closed set. Several of these have **yes** or **no** as possible values. The range of the other attributes is a menu containing all the possible options that the user can choose to fill. Preconditions governing the filling of these attributes have also been incorporated.

All the attributes of the application concepts were described in a second step. The identifier, the domain and the range were provided for each attribute. Other structural

properties (such as cardinality) were also incorporated into the attribute description as facets. The attributes of the interface concepts were described either as interface entities or as internal attributes. The attributes describing the internal concepts never appear during the communication. The description of the interface attributes includes a list of pointers to the set describing the syntactic information associated with each application term. Additionally, each interface attribute is classified into the syntactic-semantic taxonomy of attributes.

The CO concepts representing SIREDOJ were described by 49 different attributes. There are 18 internal attributes and 31 interface attributes. The interface attributes were classified according to the syntactic-semantic taxonomy of attributes in the CO general level. These classes give the linguistic information necessary for expressing all attributes describing the application concepts. Each of the basic classes represents a different grammatical role. Each compound class represents the combination of two of the basic roles. The classes used to define the conceptual attributes representing SIREDOJ were the basic classes **WHO_DOES**, **WHO_OBJECT**, **WHAT_OBJECT**, **OF**, **OF_TYPE** and **IS** and the compound classes **IS_SUBJECT** and **DOES_SUBJECT**.

The compound classes **IS_SUBJECT** and **DOES_SUBJECT** represent attributes described by two terms: the first corresponds to an entity related to the concept, and the second describes this entity. The class **IS_SUBJECT** describes the being or state of an entity describing the concept. The class **DOES_SUBJECT** describes the action carried out by an entity describing the concept. The expression of attributes in these classes requires a simple clause where the first term plays the role of subject. For example, the attribute **expenses_justified** describing the concept **PAYMENT_REQUIREMENT** belongs to the compound class **IS_SUBJECT**. The range of this attribute is the set **yes/no**. The filling of this attribute can be expressed by an attributive clause having as subject a realization of the first term (i.e. *the expenses*) and as attribute a realization of the second term (*justified*).

The range of the attributes described may consist of CO concepts, such as the values of the attributes **subject1**, **subject2** and **object** describing the concept **BUILDING CONTRACT_PARTS**, as well as closed sets (**yes/no** and **menus**). The range of most of the interface attributes corresponds to closed sets. 16 of these attributes describing the application concepts are binary-valued. The remaining 12 attributes are closed sets of values that can be presented as menu-lists at run-time. These menus are displayed on screen at run-time to guide the user in introducing the correct value.

For example, the range of the attribute **reasonnotpaid**, describing the **PAYMENT_REQUIREMENT** is a set containing all possible causes that people who

assign a building contract may have for not fulfilling their obligation to pay. These possible reasons are: intentional default (*voluntad deliberadamente rebelde*), an impossibility arose (*imposibilidad sobrevenida*), frustration of objective (*frustración del fin*), force majeure (*fuera mayor*) and refusal to comply (*negativa a cumplir*). A menu containing all such possible reasons for non-payment will be displayed on the screen.

In a third step all terms expressing the concepts, attributes and values described in the CO application level together with their syntactic information were provided. This information consists of all possible superficial representations and their associated categories, and the syntactic features. The syntactic category must correspond to the category of an LO object. More than one superficial presentation can be associated with a concept or attribute.

No new task was incorporated in the taxonomy of tasks because it already contained all communication tasks needed for SIREDOJ.

8.2.2 The interface generated for SIREDOJ

Linguistic coverage

The grammar and lexicon generated by the basic set of rules for SIREDOJ cover the expression of all the information the users were asked for during the system performance. This information consists of the description of specific cases of the building contracts.

The interface generated guides the user to introduce this information. The use of precise and simple language, adapted to the application, has been one of the main goals in order to achieve a friendly and efficient interface.

Coverage of the grammar and lexicon generated for SIREDOJ is minimal, that is, only the more simple realizations of the application objects are considered. As the user is guided to introduce the sentences, a minimal grammar proves to be more efficient than a grammar covering different realizations for the same objects. However, in order to explore the limits of the system, a second grammar and lexicon was generated by using the alternative set of rules provided by GISE. As described in Chapter 6, this alternative considers several forms of expression for the operations filling the different classes of attributes. Additionally, new possible realizations of the conceptual attributes were incorporated into this second lexicon. The resulting grammar is not so efficient as the reduced one but its broader coverage makes it more appropriate for use

when the user introduces the input manually. A more extensive grammar could be also obtained by incorporating new linguistic structures into the LO.

The minimal coverage grammar and lexicon generated by the basic set of rules is described below. The rules and lexical entries are shown in the appendix.

The grammar generated

The language supported by the grammar generated for SIREDOJ covers various linguistic phenomena, such as passive voice and coordination. These phenomena are necessary to support the friendly and natural expression of all the information required from the user.

The grammar generated expresses the operations for creating and describing particular instances of the concepts representing the application. It contains 26 rules. Two of them correspond to the rules expressing the creation of instances of concepts, described in Chapter 7. There are 16 rules representing all the clauses that express the simple operations filling the attribute of a conceptual instance. These grammar rules cover the expression of the 8 different syntactic-semantic classes of attributes appearing in the description of the application conceptual attributes.

The realization of the more complex operations filling more than one attribute of an instance is supported by six different grammar rules. Two of these rules represent coordinated clauses. The remaining four rules represent the transitive clauses filling more than one attribute of a conceptual instance.

The lexicon generated

The lexicon generated contains all words necessary to express the operations modifying the CO case level.

All terminal categories appearing in the grammar are associated with one or more lexical entries. Lexical entries correspond to application terms as well as to general words, such as auxiliary verbs and prepositions. Lexical entries representing the application terms can be associated with words or with dynamic functions.

Lexical entries representing words common to all applications are obtained from the LO. Those entries representing the application concepts, attributes and values are generated by the control rules. The syntactic information in these entries is obtained

from the definition of the application terms provided by the user. All the terms necessary for the SIREDOJ application are represented in lexical entries. The syntactic categories associated with these terms correspond to LO objects belonging to the rank **WORD** and **GROUP**. That is, those application terms containing more than one word are associated to linguistic objects in the rank **GROUP**. For example, a nominal group containing an article and two nouns may express a concept, as in *The payment requirement* expressing the concept **PAYMENT_REQUIREMENT**.

In the interface generated, the user introduces a sentence by selecting all possible options displayed on screen. Grouping the words necessary to express a CO object in one lexical entry reduces the number of selections, facilitating the task of the user. It also improves the processing of user interventions.

As described in previous chapters, the control rules incorporate semantic information into the lexical entries representing the application terms. This information is obtained from the application representation in the CO. The syntactic category is augmented with semantic features. Additionally, each lexical entry includes its semantic interpretation.

The lexicon generated for SIREDOJ contains 112 lexical entries. These lexical entries correspond to the concepts, attributes and values representing the application, as well as a few general words.

There are 39 lexical entries representing the expression of the application concepts that appear in the communication. These concepts are the 9 classes of interface concepts described in the CO application level. Each one of these concepts was associated with two or more lexical entries in the lexicon provided when describing the application. All these concepts were associated with lexical entries representing nominal groups. Thus, when generating the interface lexicon, four lexical entries were provided for each concept: one representing the definite nominal group, one representing an indefinite nominal group and two dynamic entries. One of these dynamic entries is associated with the predicate **instance**, obtaining all existing instances of the concept at run-time. The other dynamic entry is associated with the predicate **name**, asking the user to introduce the proper noun of an instance of the concept. There are three more lexical entries representing the concepts that can also be expressed by verbal form (i.e. the verb *contract* and its synonym *assign* express the concept **BUILDING_CONTRACT**).

The use of dynamic entries significantly reduces lexicon size. If dynamic entries were not used, then all possible instance names and attribute values would have to be represented as lexical entries.

The 31 interface attributes in the CO application level are expressed by 36 different lexical entries. Each attribute is related to one entry, except for the attributes belonging to the compound classes that are related to two different entries. These lexical entries correspond to definite nominal groups, descriptive adjectives and verbs. In the generation of the minimal coverage grammar and lexicon, non-different forms of attribute realizations are considered.

The values of these attributes are associated with 13 dynamic entries in the lexicon. Three of these entries correspond to instances of concepts and 10 to menus that are displayed during the communication.

The remaining lexical entries correspond to closed functional categories (negation particles, prepositions, auxiliary verbs, etc.)

The integration of the NLI generated

The NLI generated for SIREDOJ has been integrated into the whole system, including the menu driven interface the system previously used. This interface is very simple, and was wholly integrated with the system functional tasks. This menu-driven interface requests the user for specific information when the application requires it. The NLI generated by GISE guides the user to introduce the information. The information introduced by the user consists of operations modifying the CO case level. Once all operations expressed by the user are executed, then the application is called. When the application needs information, it searches for it in the CO. If more information is required, then the second menu-based interface requests it from the user. This architecture corresponds to the second possible configuration of the Control Module.

8.2.3 Assessment of the NLI generated for SIREDOJ

Applying GISE to SIREDOJ solved earlier problems in communication between the user and the application. The GISE interface guides users to introduce any information required by the system at any stage. The wide language coverage of the interface generated makes the introduction of information easier and more natural. Information requiring a long menu chain in the original interface can be expressed in one simple user intervention in the interface generated by GISE. For example, in the previous SIREDOJ communication process, once it had been established that the user wanted to

know about a specific building contract, the system requested the following information:

(Each question and its corresponding answer was represented in a different menu)

- *¿Quién es la persona que encarga?* - *El propietario*
(*Who is the assignor?* *The owner*)
- *¿Quién es la persona encargada?* - *El constructor*
(*Who is the assignee?* *The constructor*)
- *Se asume la existencia de un contrato de obra entre las partes*
(*The existence of a contract to build is assumed between the parties*)
- *¿Se asume que el contrato contiene cláusulas esenciales?* - *Sí*
(*Are essential clauses assumed to be contained in the contract?* *Yes*)
- *¿Han cumplido las obligaciones las dos partes?* - *Sí*
(*Have both parties accomplished the duties?* *Yes*)
- *¿Ha cumplido el constructor su obligación de construir?* - *Sí*
(*Has the constructor fulfilled the building requirement?* *Yes*)
- *¿Ha cumplido el constructor con su obligación de entregar?* - *No*
(*Has the constructor fulfilled the delivery requirement?* *No*)

The acquisition of all this information by the previous interface required seven menu choices. Each time the user chose an option from a menu, it was processed by the application and generated a new menu asking for additional information.

In the GISE NLI the grammar has been obtained from the application representation in the CO. Thus, all the information that must be expressed by the user is encoded in the grammar. The same information described above can be expressed directly in one simple user intervention.

The interface guides the user to build the following sentences:

El propietario ha asignado una obra al constructor

(*The owner assigned a building to the constructor*)

La obligación de construcción está cumplida

(*The requirement to build is fulfilled*)

La obligación de entrega no está cumplida

(*The requirement to deliver is not fulfilled*)

Furthermore, in the interface generated by GISE, the user retains the initiative when introducing the sentence. The interface generated guides the user to introduce the information needed at each state of the communication process. For this reason, the user does not need to wait for the KBS intervention to introduce the input.

The incorporation of a menu-mode displaying all accepted NL choices on screen improves the efficiency and friendliness of the NLI generated. The user does not need to type the NL sentences, a sentence can be introduced by clicking the desired options appearing on screen.

Several possible interfaces could be obtained for SIREDOJ. GISE allows the modification of the knowledge bases involved in the communication process. New tasks could be included in the CO taxonomy of tasks. Linguistic classes supporting new linguistic phenomena could be incorporated into the LO. Even different control rules establishing new relations between the CO and LO could be used for obtaining the application-restricted grammar. However, the modification of these bases is not simple. It requires a deep knowledge of their organization and implementation. For this reason, all the knowledge bases have been designed to cover the communication acts required in specific type of application: the ESs performing heuristic classification.

In fact, as SIREDOJ is the first application of GISE, the different data bases have been revised when generating the most appropriate NLI for the KBS. Compound classes have been included in the syntactico-semantic taxonomy of attributes. In particular, the classes **IS_SUBJECT** and **DOES_SUBJECT** have been incorporated to represent the linguistic behaviour of the attributes describing an entity related to the concept. Including compound classes has proved satisfactory for representing the linguistic behaviour of attributes describing an entity directly related to the concept.

The basic set of control rules was improved by evaluating the efficiency of the grammar and lexicon generated. The basic requirements supported a natural expression of all information required by the system and efficiency in processing user interventions. Those requirements have been achieved by generating a small grammar supporting the natural expression of all possible application operations.

The incorporation of dynamic mechanisms, dynamic entries and preconditions, allows the adaptability of the lexicon and grammar to the application performance at each step of the communication.

Using menus displaying all possible attribute values guides the user to introduce the information correctly. This help is particularly useful when describing specialized terms, such as those involved in SIREDOJ.

The representation of the lexical entries representing nominal groups improves efficiency when introducing and processing by using sentences.

Applying GISE to KBSs similar to SIREDOJ would not require important changes. New classes representing new linguistic realizations can be incorporated in the basic attribute taxonomy without requiring any change in the other knowledge bases. The same basic set of control rules could be applied to these applications without requiring any changes. However, if the user knows the system, new control rules can be incorporated to create different grammar rules. One important aim in the system design is the reuse of the knowledge involved and the reduction of the intervention by an expert in GISE.

8.3 THE APPLICATION OF GISE TO A TELEPHONIC RAILWAY COMMUNICATION SYSTEM

GISE has been adapted to generating an NLI for a train consulting system. The purpose was to adapt GISE to a different type of application it was designed for, and thus to enrich the different knowledge bases of the system. In the telephonic railway systems, user interventions consist of consulting specific information. Unlike in the KBSs, the concepts involved in railway communication systems are simple, and no specialized language is required. However, user interventions in such systems cover a rich variety of linguistic phenomena.

The knowledge required to generate an NLI for consulting trains has been obtained from the project **Spontaneous-Speech Dialogue Systems in Limited Domains** ([TIC98]). The aim of this project is developing a telephonic consulting information system for a Railway company (RENFE, the Spanish public Railway Company). For this purpose, a corpus of dialogues of the RENFE telephone service has been studied. In this service, a specialist answers user questions about Spanish train lines. Most of the user interventions are about the train schedule. Many of these interventions are not grammatically correct.

In this case, the interface generated has not been used directly in the telephone communication application, but rather only for testing and tuning purposes. Once generated, the grammar must be translated to the format used in the project.

8.3.1 Representation of the domain

In order to study the phenomena in this communication arise, a corpus of dialogues registered in the train consulting service mentioned above was analyzed. The information for answering user queries is provided by the RENFE Web site.

The purpose of adapting GISE to this new application was not to obtain a grammar covering all possible user interventions, but to study new communication phenomena. For this reason, only a subset of the application domain was represented in the GISE CO. For the same reason, although linguistic information has been incorporated into the LO and the syntactico-semantic classification of attributes to cover most natural forms of express questions, not all possible expressions in this domain are covered.

In the CO application level, 33 new conceptual classes representing the train information required were defined. These concepts have been defined in base to the CO general level. The application conceptual classes consist of 23 concepts used in the communication, 10 concepts are not used. No preconditions were attached to these concepts. The 23 concepts used in the communication are described below:

TRAIN. This concept is the core of the application. It is described by 24 interface attributes. These attributes are: **departure**, **arrival**, **time_departure**, **time_arrival**, **duration**, **type**, **capacity**, **seats**, **complete**, **stops**, **days**, **fares**, **discounts**, **numvagons**, **toilet**, **couchette**, **sleeping_compartment**, **airconditioning**, **restaurant**, **telephone**, **car_transport**, **smalcar_fare**, **mediumcar_fare** and **bigcar_fare**. These attributes describe different information related to trains. The filling of these attributes cover different linguistic phenomena. Most of these attributes belong to several subclasses of the class **OF**, such as **departure** and **arrival**, belonging to the class **OF_WHERE**; **time_departure** and **time_arrival**, belonging to **OF_TIME**; **duration**, **fares**, **smalcar_fare**, **mediumcar_fare** and **bigcar_fare**, belonging to **OF_QUANTITY**, and **type** belonging to the class **OF_TYPE**. Most of the remaining attributes belong to the class **has**, such as **restaurant**, **telephone**, **sleeping_compartment** and **toilet**.

TRAIN_STATION. This concept is described in Figure 4.2. The five interface attributes describing it are: **name**, **city**, **conects_to**, **taxi** and **car_park**.

LINE. This is described by the attributes: **origin**, **final** and **station_set**.

TRAIN_STOP. This is described by the interface attributes: **name**, **time_departure**, **time_arrival**.

SEAT_TYPE. This concept is described by the attribute **type** and the attribute **number**, indicating the quantity of a specific type of seats in the train.

SEAT_FARE. This concept gives information of the fares for a specific type of seat in a train. It is described by the interface attributes: **type** (representing the seat type), **daily** (the fare for working days), **holiday** (the fare for holidays), **return** (the working day return fare) and **returnholiday** (the holiday return fare).

DISCOUNT. This concept is represented by three attributes: **name**, **quantity** and **condition**. It has 8 subclasses representing the different types of discount that can be applied to all types of RENFE trains. These subclasses are: **RETURNND** (discount for return journeys), **PENSIONERD** (for people over 65), **CHILDREND** (for children between 4 and 6), **GROUPD** (for groups), **BLUEDAY** (for special days), **BONUS** (for frequent travelers), **4*1D** (for four people traveling in the sleeping compartment) and **TOGETHERD** (for two adults traveling in the sleeping compartment).

PAYMENT. Described by the interface attribute **type**.

GENERAL_INFORMATION. This concept describes general information. It has only two attributes: **information_panel**, describing where the information is available, and **normative**, describing the regulations related to trains. The value of the second attribute is the concept **NORMS**.

NORMS. This concept describes regulations covering the transporting of objects and animals. It is described by the interface attributes **allowed_weight**, **allowed_transport**, **allowed_animal**.

The attributes describing the concepts were also defined in the CO. 44 interface attributes and 6 internal attributes were described in the application level. These attributes were classified according to the taxonomy of attributes. The basic classes of attributes used to describe the linguistic behavior of the conceptual attributes are: **HAS**, **IS**, **OF** and the subclasses **HAS_QUANTITY**, **OF_QUANTITY**, **OF_TIME**,

OF_WHERE and **OF_TYPE**. The compound classes **OF_SUBJECT**, **OF_QUANTITY_SUBJECT** and **OF_TYPE_SUBJECT** were also used.

A few subclasses of the basic classes of attributes have been incorporated.

These new classes have been defined to represent attributes in the classes described above, associated with specific verbs and prepositions. The newly incorporated classes enrich the system as they also appear in other domains and types of communication.

The range of the attributes described consists of CO concepts as well as closed sets of values. The range of 8 of the interface attributes corresponds to concepts defined in the application level. The range of 11 attributes are units expressing quantities and times. The range of one attribute is a proper name. The rest of the interface attributes corresponds to closed sets. The range of 11 of these attributes describing the application concepts is the closed set **yes/no**. The range of 12 of these attributes is a menu containing a closed set of values.

8.3.2 Assessment of the NLI generated for the train consulting system

Adapting GISE to generating an NLI for a train consulting system has required the incorporation of certain crucial knowledge as the system was designed for a different type of application. The system was designed to guide the user in introducing specific information about a domain in a natural manner. Applying the system to cover the broad-ranging language a user can articulate when asking a question about trains requires more information regarding the linguistic realization of attributes in interrogative sentences. A few classes representing interrogative clauses were incorporated into the LO domain level. A few subclasses of the basic classes of attributes were also incorporated into the syntactic-semantic taxonomy of attributes.

As mentioned before, incorporating new classes of attributes does not imply any change in the other knowledge sources in GISE. However, it *does* require some linguistic background and knowledge with regard to the system. Adapting the system to a given type of application reduces the necessity of incorporating new knowledge for a specific application.

The coverage of the generated interface is not exhaustive, but can be easily extended by incorporating new information retrieved from new dialogue corpora.

The resulting interface has not been integrated into the prototype developed within the project mentioned before since, being menu-guided, it is not an appropriate interface to support communication in a telephone consulting system. Furthermore, the grammar

generated is not the most appropriate in processing interventions in a spoken consulting system where the user introduces the input spontaneously. GISE has been designed for generating application-restricted grammars by guiding the user and providing an efficient processing of user interventions. The rich variety of linguistic phenomena appearing when users spontaneously express their requirements is not supported in these grammars. However, they could be used to model user interventions. The interaction between the user and the system in this application, as in many other speech applications, is an interactive resolution problem. Frequently, users do not express all their requirements within a single sentence; they wait for the help of the system. In this type of situation, a dialog system can guide users towards obtaining their objectives.

CHAPTER 9

CONCLUSIONS

This thesis is devoted to the study of how communication with KBSs can be improved. Its aim is to improve efficiency in the development and performance of NLI to KBSs. For this purpose, GISE, a system automatically generating application-restricted NL-guided interfaces was designed. The main decisions in the system design have been discussed in previous chapters. The conclusions of the research carried out within this thesis are presented in this chapter. In the first section, the main features of the system design are compared with other systems following similar approaches. In the second section, the main contributions of the system to obtain a reusable organization supporting communication with complex applications are discussed. Finally, the last section briefly describes future research that may be undertaken.

9.1 COMPARING GISE WITH OTHER EXISTING NLI SYSTEMS

The aim of GISE is to improve both the development and the functionality of the NLIs for KBSs. For this purpose, a modular organization of the different types of knowledge involved in communication was proposed. The system was designed to automatically generate menu-guided application-restricted NLIs from the general knowledge bases

and the application requirements. This approach reduces the high cost of development and the large run-time requirements, main drawbacks in NL communication.

In this section, main features of the system design are compared with other systems following similar approaches. The first part of this section compares the GISE interface development process and the second part compares the functionality of the resulting interfaces.

9.1.1 The process of building NLI

GISE adapts the LO to the application requirements represented in the CO to supporting communication with different applications. There are, however, NLI systems following different strategies. First, significant similarities and differences between the strategy followed by GISE and that followed by other existing systems are described. Then, the CO and LO of GISE are compared with most well known ontologies used in NL processing.

Building the interface linguistic resources

Application-restricted linguistic resources reduce the large run-time requirements of large coverage linguistic resources. Application-restricted NLIs have proven efficient especially when the sentences introduced by the user are limited to those supported by the grammar and lexicon. For this reason, many NLIs include a menu-system to guide the user to introduce the NL, as GISE does.

Application-restricted grammars and lexicons are, however, expensive to develop and difficult to reuse, especially for complex systems like KBSs. To avoid this problem, several systems, GISE being one of them, have been designed to deal with the construction of transportable NL communicative modules adapted to the application.

Several works in this area follow a simple strategy consisting of adapting a domain independent grammar to the specific application requirements. For each application, a semantic grammar and lexicon adapted to the tasks is obtained. The resulting grammar allows the user to express the operations over the application in a natural form. That is, it avoids the user having to learn any low-level query language or any other

implementation language. NL_MENU, INKA(INGlish Knowledge Acquisition) and the Interface Structured Languages (ISLs) are examples of relevant work following this strategy.

A major problem with semantic grammars is that they are completely domain dependent and thus they have to be built for each application. This is not an important problem for interfaces performing a limited number of simple operations. This is the case of the NLI's generated by the system NL-MENU, supporting easy questions over databases, and those built by the system INKA, supporting a limited number of simple constructing operations. The construction of the linguistic sources required in these interfaces is not significant, because it is reduced to the cost of adapting general linguistic resources to the concepts of the application. The linguistic resources consist of a grammar supporting the expression of the operations general to a type of application. This general grammar is adapted to the objects appearing in an application. The NL supported by the application-restricted grammar and lexicon is limited to a number of statements. Semantic restrictions encoded in the grammar are few.

There are systems automatically generating NLI's, as GISE does. The NL-Menu system is a successful example of this. This system is divided into two basic components. One component produces a domain specific data structure by engaging the user in an NL-Menu dialogue. This data structure consists of a list of categories specifying all relations and types of attributes that the interface will cover. From the resulting data structure, the second component generates a semantic grammar and lexicon. This process consists of instantiating a domain independent core grammar and lexicon.

Semantic grammars are also used for supporting NL communication with ESs, as in the system INKA. This system supports knowledge acquisition during the building of ESs. It is an NLI system included in the DETEKTR (Development Environment for Tektronix Troubleshooter Environment), an environment including different tools for the development of a family of ESs. The semantic grammar supporting NL is expressed in GLIB (General Language for Instrument Behavior), a ISL. The syntax of the ISLs is similar to English and their semantics is close to the tasks to be performed. The most well known of the ISLs is INGLISH.

Nevertheless, semantic grammars are not the most appropriate for systems supporting communication with different KBSs, as GISE does. In interfaces to KBSs, extremely complex communicative relationships arise. The number of statements the users may need to produce increases. The complexity of the objects and relations involved is larger. Furthermore, the syntactic complexity of Spanish is higher than that of English, the language supported by the NLI systems described above. For these reasons, a syntagmatic approach is followed in GISE when generating the linguistic grammar and lexicon supporting Spanish communication.

There are other NLI systems dealing with the construction of syntactic grammars adapted to the applications. A relevant example of these is SESAME, a system generating interfaces for different relational databases. The language supported by the generated interfaces is French. The resulting interfaces include a menu-system. For this reason, the coverage of the grammar is restricted to supporting a paraphrase of each possible operation. The SESAME system guides the user in adapting a syntactic grammar and lexicon for a specific database. The aim of the system is, however, to build a general grammar reusable for all databases without any (or with few) changes. Once this grammar is obtained, only the lexicon will have to be built for each application.

The process of obtaining syntactic grammars adapted to the application is much simpler for relational databases than for complex applications, such as KBSs. The linguistic and conceptual resources necessary for supporting communication with different KBSs are much larger than those required for supporting communication for databases.

One of the most relevant systems supporting communication in Spanish with KBSs is the MMI2, a multi-modal interface. Although the system supports more than one language of communication, each language module has been developed independently. Furthermore, only one language is accepted in each session. The Spanish module of the MMI2 uses a general grammar. It is a dependency grammar represented in complex feature structures. The parser is a bottom-up parser. The complex features resulting from the parser are translated to first order predicates. The lexicon is adapted to the application. The morphological process is robust, it recognizes abbreviations, proper names, dates, etc. To disambiguate the result of the morphological process, a markovian filter was included. This filter passes a ranking of all possibilities to the

parser. The filter can be adapted to the language supported in the communication with KBSs.

GISE does not use a general grammar for all KBSs as each language module in MMI2 does, but rather an application-restricted grammar. As a consequence, the resulting grammar is more efficient. The number of grammar rules is lower. The interpretation process is much simpler. Furthermore, because a menu-system is included in the GISE interfaces, not every possible paraphrase of a sentence needs to be understood. In fact, only one paraphrase for each operation is required.

Although the grammar formalism used in MMI2 and GISE is not the same they have some similarities. The two formalisms allow the incorporation of syntactic and semantic features. The syntactic and semantic process is undertaken in parallel in the two systems. The interpretation process is much more complex in MMI2 than in GISE because in MMI2 all user interventions must be translated to a representation formalism common for all communication modes.

Using ontologies for processing NL

An additional problem in KBSs is that, in most, no schema or description is available. The propositions in such systems may have arbitrary meanings. For this reason, many NLI systems to KBSs incorporate a knowledge base or ontology in representing the application task and domain needed to support communication. The MMI2 system incorporates a semantic representation (or expert) to build the application lexicon. It represents the application and communication concepts and relations. This semantic expert is also used during the interpretation process for disambiguation. The MMI2 includes, in addition, a second knowledge base with certain similarities to the CO in GISE. The most important of these similarities is that the two systems use this knowledge base to interchange knowledge with the KBS.

In most of the ontologies used in NLI systems for complex applications, the concepts and relations between them are not formally defined, and the number of concepts represented is not very large. Most of these ontologies are linked to lexicons to provide the words for expressing the concepts and relations represented. Many of these systems use these ontologies for obtaining lexical information and for disambiguating

during the language processing. In GISE, the CO is used for generating the linguistic resources most appropriate to an application as well as to interchanging knowledge with the application.

A linguistic ontology to organize the general linguistic resources is also incorporated in many NLP systems. One of the most complete linguistic ontologies is SENSUS, which integrates large-scale linguistic ontologies for the machine translation of several languages, text summarization and generation. The LO in GISE, although following a general linguistic theory, is reduced to the linguistic structures necessary to support the NL communication with KBSs. In fact, it incorporates a domain level adapted to the CO. A similar approach is proposed in the system described in [Busemann98], using a domain-motivated linguistic ontology that supports rapid adaptation to new tasks and domains.

The syntactic-semantic taxonomy of attributes incorporated in GISE represents an intermediate level between conceptual and linguistic knowledge. Most existing NLI systems incorporating a conceptual ontology only distinguish the conceptual and linguistic level. In these systems, the categories correspond directly to the application concepts and relations represented in the ontology. However, several NLP systems incorporate, an intermediate level representing syntactic-semantic information. The most well known syntactic-semantic representation is that of GUM ([Bateman90]), a general representation for use in different NLP systems. The GUM ontology is used in several NLP systems, most of them for generating text, such as Penam, Komet, TechDoc, ALFresco and OntoGen. In GISE, however, such a complete ontology is not necessary. GISE only requires a minimal syntactic-semantic taxonomy covering the expression of the operations supported by the CO. As those operations are frame-based, the taxonomy is limited to the attributes describing the CO concepts. A kind of syntactic-semantic level was also distinguished in earlier works, such as those of Jacobs and Perkins.

Using larger conceptual and linguistic resources complicates the process of adapting the general resources to a specific application. This process is more complex in GISE than in NL-MENU or other systems using simpler resources. Nevertheless, in GISE, the grammar and lexicon are obtained automatically from the application representation in the CO.

The generality of the linguistic resources used allows the obtaining of larger and more complex grammar. An important advantage in GISE is that grammars of different coverage can be generated for an application. A small grammar representing only one paraphrase of each possible operation is appropriate for interfaces including a menu-system. Larger grammars representing more operation paraphrases are required in interfaces that permit the user to type NL input. Although GISE automatically generates a small grammar for menu-system interfaces, the generality of the linguistic resources can support the generation of larger grammars. Only the rules controlling the process must be modified. These control rules are expressed in a declarative and powerful formalism not difficult to understand.

Furthermore, the organization of the knowledge in GISE facilitates the adaptation of the system in supporting other communication languages. The organization of the linguistic resources in other systems is wholly oriented to a specific language. There are systems supporting more than one language, as the MMI2 does, using independent linguistic resources for each language, including different parsers. In GISE, for example, few changes in the LO were necessary in adapting the system to supporting communication in Catalan. Supporting communication in other languages not as similar to Spanish, such as English, would require the incorporation of a few classes in the LO as well as a few changes in the syntactic-semantic taxonomy of attributes. To provide the system of a basic set of rules for automatically generating the application-restricted grammar and lexicon in English would also require a few changes to adapt the existing control rules to the new classes incorporated in the LO and in the taxonomy of attributes.

9.1.2. The performance of the resulting interface

Using application-restricted linguistic resources has proven efficient when the sentences introduced by the user are limited to those supported by the grammar and lexicon. For this reason, GISE, as well as NL-MENU, INKA, INGLISH, SESAME and many other NLIs include a menu-system to help the user build the sentences. The menu-system controls the displaying in the screen of the NL options acceptable to the system at each step of the communication process. Rather than requiring users to type

input to the system, input is made by selecting items from a set of dynamically changing menus. Once an NL option has been selected, the next possible options are displayed. In these systems, the parser must be able to analyze a word at a time and to predict the set of the next possible words in a sentence, given that the input has come before.

The performance of the menu-system is not exactly the same in all NLI. In the NLIs generated by the NL-MENU, for example, the application-interface menus contain the same words at all times. Furthermore, the entire set of menus is displayed, including the inactive menus. The advantage of this approach is that users can consult all the menus when they have doubts on how to proceed. However, this approach requires a small number of lexical entries. As the lexicon in GISE is not so small, a different strategy is followed. In GISE, the words contained in a menu change during communication. The words contained in a menu are only displayed when the name of the menu is clicked. If the menu is inactive, its content is not displayed.

The use of menu-guided application-restricted NLIs has been successfully applied because it improves the friendliness and efficiency of NLIs. The main goals in man-machine communication, user satisfaction, task success and low dialogue cost have been achieved in those NLIs. Although efficient NLIs including menu-systems already exists, they have been developed for simple applications, such as databases. GISE is designed to achieve fundamental goals in man-machine communication for complex systems, such as KBSs.

User satisfaction

Interfaces including a menu-system displaying all NL options have proved friendly because they solve the problem of the lack of user knowledge concerning the language recognizable to the system. The scope and limitations of the system are made immediately clear to the user and only understandable sentences can be input. Thus, all user inputs fall within the linguistic and conceptual coverage of the system. Furthermore, users can avoid typing long sentences. The user has only to choose among the options displayed on-screen in order to gradually build a sentence.

Another advantage of using the menu-system is that words are parsed one at a time. Most of the work necessary to parse a sentence is done before the sentence is completely input and thus the perceived parse time is much less.

Several tools are also incorporated into the GISE, as well as in other menu-base systems to improve user satisfaction. The rubout facility enabling the users to erase phrases, as well as other facilities included in the NL_MENU, was also included in GISE.

Although the interfaces generated by GISE only support the NL mode, the grammar and lexicon generated, as well as the parser could be integrated with any other mode of communication. Most existing multi-modal systems incorporate the NL module as an independent module.

The task success

One of most important advantages of the menu-based application-restricted NLIs is that all inputs can be understood, thus giving a 0% failure rate. The task success is complete when the grammar and lexicon are automatically generated from the application model. In this case, only the expression of the tasks the application can perform are supported by the grammar. Furthermore, semantic information is incorporated into the grammar rules to optimize the interpretation process, which returns a task to perform.

Task success can be assured in GISE, where the approach of automatically generating menu-based NLIs adapted to the applications is followed. In GISE, the grammars and lexicon are obtained from the CO representing the application, and thus they support the expression of all possible operations for an application. The semantic information necessary to successfully perform the tasks expressed by the users is also incorporated into the grammar. Because GISE support communication with complex systems, preconditions are attached to the grammar rules in order to activate only the rules expressing the operations that can be performed at each state of the communication. Such preconditions are not necessary in NLIs for simple applications, such as consulting databases.

However, task success is not assured in GISE, or in other similar systems, in cases in which user input is introduced manually, or when the grammar used is not that automatically generated by the system. Most of the systems generating grammars and lexicon, such as GISE and the NL-MENU, allow for user-modification. The need to consider all communication mistakes that may arise in such cases would enormously complicate the control dialogue.

The dialogue cost performance

The high cost of the dialogue performance is one of the main drawbacks in NL communication. This cost has been drastically reduced in NLI systems including a menu-system. In these systems, the user can only correctly introduce the sentences expressing the possible operations. As mentioned before, these systems have been successfully developed for simple applications, such as consulting databases. Integrating a menu-system into the NLI systems generated by GISE reduces dialogue cost performance, even when the complexity of the applications is high.

The use of a small grammar and lexicon improves language processing. As mentioned before, the grammars in these systems are usually small because only one paraphrase is needed to express all operations. The user is guided to this paraphrase by the menus. Furthermore, the incorporation of semantic information in the grammar restricts the options to be considered at run-time.

The parser of many of these systems, as that of GISE, is a left-corner parser using a reachability structure to optimize the obtaining of the next acceptable choices. This reachability structure is built from the semantic grammar and the lexicon. It indicates all possible grammatical categories that can be recognized from a specific category.

To reduce the number of lexical entries required, GISE, as the NL-MENU, use open categories associated with functions asking the user to introduce proper nouns and quantities. GISE also includes dynamic categories representing the particular information previously described by the users. As this information is represented as instances of conceptual concepts, these dynamic categories are associated with a function returning all existing instances of a concept at run-time.

9.2 MAIN CONTRIBUTIONS

The research carried out on achieving the most appropriate representation of the different types of knowledge needed in the communication process has been discussed in previous chapters. The main contributions of this thesis in achieving a reusable organization supporting communication with different types of application were described in detail.

This section is a brief summary of the main contributions to the dissertation. First, it outlines the main contributions in the organization of the different types of knowledge involved in the process of obtaining the most appropriate linguistic resources for each application. Following this, the improvements in the communication achieved by using the resulting application-restricted linguistic resources are pointed out.

9.2.1 The reusable organization of the different types of knowledge involved in the communication process

This thesis proposes the representation of the different types of knowledge involved in the communication in separate and reusable knowledge bases. This organization consists of a CO, representing conceptual knowledge, an LO and GL, representing the linguistic knowledge and a set of control rules in charge of adapting the general linguistic knowledge in the LO to the application knowledge represented in the CO.

This organization favors the reusability of the different knowledge bases, reducing the cost of generating application-restricted interfaces. The representation of the knowledge involved in the communication process in separate data structures provides the system with great flexibility in adapting it to different types of applications,

platforms, etc. The consistency and clarity of the formalism used to represent these knowledge bases facilitates their extendibility, if necessary.

The main contributions to the design of these knowledge bases are outlined below.

The CO

The CO provides a general framework for representing application concepts and tasks involved in the communication. The CO is organized in three independent taxonomies, representing the general concepts modeling the application, the attributes describing these concepts and the operations to be performed on the application concepts. A set of predefined structural descriptors or facets is provided for representing concepts, attributes and operations. These facets include preconditions on the concepts and operations. A consistent formalism for describing preconditions was also defined.

The CO taxonomy of operations describes the general tasks expressed in communication. Including a detailed description of all possible communication tasks facilitates the obtaining of the application-restricted subgrammar and lexicon. Only the linguistic resources to express these tasks must be considered. Additionally, information concerning these operations (their parameters and preconditions) is included in the application grammar rules and lexical entries in order to improve the processing of user intervention.

Since there is no linguistic theory establishing all possible relations between general conceptual knowledge and its detailed linguistic realization, these general relations were obtained by studying the sublanguage used in communication with KBSs. Due the communication tasks are mainly operations describing and consulting the conceptual attributes, a major problem is that of the expression of the attributes. Detailed information about the realization of the conceptual attributes has been incorporated into the taxonomy of attributes included in the CO. This taxonomy is syntactic-semantic oriented. Including a taxonomy capturing specific linguistic distinctions facilitates the generalization of the relations between the conceptual attributes and their detailed linguistic realization. This taxonomy acts as an interface between the application conceptual knowledge and the linguistic knowledge. It facilitates the obtaining of the linguistic resources necessary to express the conceptual

attributes. Representing the classes of attributes as CO objects provides a consistent and declarative form of describing the linguistic realization details associated with each class.

The basic classes of attributes are associated with the different grammatical roles appearing in the sublanguage used in communication. Subclasses of these basic classes are obtained considering more detailed information relevant for the realization of the attributes. Although the taxonomy is not very extensive, it captures the necessary linguistic distinctions when consulting and describing the conceptual attributes. New classes can be incorporated to support different linguistic phenomena.

The attribute classification is based on Spanish linguistic distinctions. However, it could be reused in other languages with few changes. The basic classes are associated with grammatical roles common to other languages. No changes were required to cover Catalan linguistic distinctions. Few changes are required to adapt it to English.

The LO

The linguistic knowledge was organized following the basic principles of the Nigel grammar, a large systemic functional grammar (SFG) of English. This grammar, already implemented as a component of the Penman text generation system, provides the base to describe consistently the linguistic knowledge needed during the communication and to extend it to other (different) purposes. The main reason to adapt the basic principles followed by this grammar to our proposal is that it places the communication function in the foreground. In the system proposed in this thesis, an organization of the linguistic resources considering the functional and pragmatic issues as the central areas seems more appropriate than an organization focused on the structure and syntax. Such an organization facilitates the obtaining of the most appropriate linguistic structures to express the application communication tasks. Furthermore, an organization focussed on functionality is easier to adapt to different languages because there are many functional issues common to them.

Detailed information required to express the communication tasks common to all applications was also included in the linguistic knowledge base. This knowledge covers the expression of all operations described in the CO. The incorporation of this

information improves the efficiency of the process of obtaining the linguistic resources required to support the communication tasks for a specific application.

The LO was designed using the same form of representation as used in the CO, an object-like fashion. In the LO, the general linguistic structures are represented as classes. The specific linguistic structures needed for an application are represented as instances of general classes. The application-restricted grammar and lexicon is obtained from these instances. Using a consistent and reusable representation formalism improves the process of obtaining the specific application grammatical structures. It also facilitates the enlarging of general linguistic resources if required.

The control rules

The process of obtaining the linguistic resources necessary in the application-restricted interfaces is controlled by a set of rules. This process is performed in two steps. In the first step, general communication tasks are adapted to the application domain-specific knowledge. In this step, a set of rules controls the generation of instances of the CO classes of operations adapted to the specific application concepts. In the second step, the linguistic knowledge in the LO is adapted to cover the application task-specific communication. In this step, a different set of rules controls the process of generating instances of the LO classes expressing all possible operations for a specific application. This set of rules establishes the relations between the operations represented in the CO and the linguistic structures required for their realization. The broad coverage of the LO provides different linguistic structures to express each operation. For this reason, more than one grammar could be obtained for an application.

In order to provide the system with an automatic mechanism to generate application-restricted grammar and lexicon, a basic set of rules controlling this process for different types of applications was defined. The basic objective when defining these rules was that of obtaining the most efficient grammar for each application. For this reason, the grammars this basic set of control rules generates are small. They only cover one form of expressing each possible operation. However, there are other sets of

rules generating larger grammars that cover different forms of expressing the operations allowed for an application.

A declarative and powerful formalism to express the control rules was defined. The clarity of this formalism facilitates the definition of new rules capturing new relations between the communication tasks and their realization, if required.

Reusing the general knowledge bases (the CO, the LO, the GL and the control rules) reduces the high cost of the process of developing NLI. Following the generation process proposed, the cost of creating an application-restricted grammar is reduced to the cost of representing the application specifications in the general CO as well as introducing the application vocabulary in the GL. The adapting of the specific NL coverage to the application task-specific communication can be undertaken automatically by the system. Adapting this knowledge to support other communication tasks does not require important changes. Once the new tasks are described in the task taxonomy, if necessary, the linguistic knowledge to cover their expression must be incorporated into the LO application level. The control rules assure the generation of the interface best suited to the complexity and size of the application.

9.2.2 Improving communication

The knowledge organization proposed in this thesis supports an efficient NL communication because the general linguistic resources are adapted to the application requirements. The strategy of using application-restricted grammars had already been followed by other NLI systems. This strategy has already been applied successfully in NLIs to simple applications, such as consulting databases. As mentioned before, one of the main objectives of the GISE design was to adapt this strategy to support communication with KBSs. The linguistic resources needed in NLIs to those systems are highly complex.

The size of the grammar and lexicon GISE generates for an application is not large. Only the grammar rules and lexical entries necessary to express all possible communication acts for a specific application are incorporated. For this reason, the space and time for processing user interventions is not large even when the meanings expressed are complex and the linguistic structures used are not always simple. All

user interventions express the specific-tasks of an application represented in the CO. The process of interpreting these interventions is simple because of the conceptual information necessary to execute these tasks into the grammar and lexicon.

The dynamic mechanisms incorporated into the linguistic resources generated restrict the language accepted for the expression of the operations that the system can execute at each state of the communication. Additionally, using dynamic mechanisms reduces the number of rules and entries the parser must consider at run-time.

Application-restricted NLI has proven efficient especially when a menu-system guiding the users in introducing the NL options that are acceptable to the system is incorporated. NLIs using application-restricted resources and including a menu-system reduce the cost of dialogue and assure the successful performance of the task. For these reasons, a menu-system was also included in the interfaces generated by GISE. The NLIs generated by GISE guide the user to build sentences expressing all the information that the application can process. The description of specific information is also facilitated by the incorporation of functions asking the user for proper names and quantities, and of menus displaying all possible values of an attribute.

Additionally, the architecture of the system provides more than one possible configuration to integrate the NLI generated in order to adapt the communication process to the application characteristics (i.e. the numbers of terms must appear in the menus, etc.). The utilities integrated into the NLI can also be adapted by considering different types of users.

Finally, the modular architecture of GISE allows for the easy portability of the whole system, or a part of it, to different platforms, using different implementation languages.

Applying GISE to an ES in law has proved how interfaces to KBSs can improve their functionality by incorporating NL. The NLI generated for the ES SIREDOJ has proved much friendlier than its previous menu-based interface. The interface generated by GISE, being dynamically adapted to the application performance and guiding the user to introduce the information required, supports efficient and friendly communication. It proves that once the main shortcomings inherent in the use of NL have been solved, the semantic complexity and friendliness of the NL mode can improve communication between users and KBSs.

9.3 FURTHER RESEARCH WORK

The system described in this thesis proposes solutions to the main drawbacks in NL communication with KBSs. Applying GISE to complex applications improves efficiency in obtaining an application-restricted NLI as well as in the communication supported by the resulting interface. However, there is still work to be done on improving the system. Some of this work simply refers to the extension of the different knowledge bases, some corresponds to different uses of the application-restricted linguistic resources generated and some to adapting one (or more) particular knowledge base for purposes different to that of communication. The main lines of this future work are briefly described below.

9.3.1 Enriching the GISE knowledge bases

Although the different knowledge bases developed (the CO, the LO and the control rules) have been designed for supporting user interventions for a specific type of KBSs, they can be easily adapted for communication with different types of applications. Current implementation of the system also covers the generation of grammars for simpler systems, such as consulting systems.

Adapting GISE to other types of applications would not require major changes. For example, the design proposed could be easily adapted to generating expert-interfaces, supporting communication with the expert when knowledge acquisition is undertaken, in the building phase. Even though some information would have to be incorporated, it would not affect the process of adapting the general linguistic knowledge in the LO to the application represented in the CO. The knowledge base design facilitates the incorporation of new classes to the taxonomy of operations, the syntactic-semantic taxonomy of attributes and even to the LO.

The current implementation of the system only covers simple communication tasks. These tasks correspond to operations concerning one CO concept. Enlarging the

taxonomy of operations to cover more complex operations, involving more than one concept, is work that is still to be carried out.

System knowledge bases could easily be extended to cover new communication tasks and more complex linguistic phenomenon. In particular, future work could be undertaken to deal with linguistic phenomenon not supported in current implementations, such as coordination in negative clauses, subordination, different types of anaphora, nominal groups supporting various complements, the use of non-grammatical expressions

Although the formalism representing the different knowledge bases in GISE is easy to understand and to use, the incorporation of various aids would simplify the modification of these bases. In particular, guidance in representing the application requirements following the CO commitments would facilitate the process of providing domain-specific knowledge. This guidance could include graphics showing the taxonomy of concepts and the taxonomy of attributes. If the functionality and domain of the KBS were already represented in an ontology, then it could be easily adapted to the GISE's CO. If the domain to be represented in the CO is complex, the conceptual taxonomy can be defined with the help of existing environments for building ontologies.

9.3.2. Adapting the NLI's generated to different types of communication

The NLI's generated by GISE include a menu-system to guide the user to choose the NL options accepted by the system during communication. Although these interfaces also allow the user to type NL sentences, the control dialogue does not cover all possible mistakes and misunderstandings that may appear when the menu-guidance is not used. Future work could be undertaken in extending the dialogue component to deal with all possible communication problems when sentences are typed directly by the user.

A related work to be done is that of using the application-restricted linguistic resources generated by GISE in spoken interfaces. The grammar generated is not the most appropriate to process user interventions in spoken systems. GISE has been designed for generating reduced grammars guiding users to build sentences expressing their

requirements. Spoken systems require long grammars supporting the rich variety of linguistic phenomena appearing in oral user interventions. However, the grammar generated by GISE could be adapted to model dialogue. The interaction between the user and the system in this application, as in many other speech applications is an interactive resolution problem. Frequently, users do not express all their requirements in a sentence; they wait for the help of the system. In this type of scenario, a dialogue system can use a grammar for encoding all possible tasks to guide users in obtaining their objectives. Dialogue modeling is being incorporated in current spoken systems to facilitate error recovery and thus achieve a robust communication. Further work in adapting GISE to generating grammars modeling the communication in spoken systems is still to be done.

The integration of other modes of communication in the interfaces generated (such as speech, graphics and commands) is also an interesting topic to study in the future. The interfaces supporting multimodality deal with specific problems not considered in the NLIs generated by GISE. The most important of these problems is the use of a representation formalism common for all communication modes. The treatment of the anaphoric reference between the NL and other modes is also a major problem when integrating NL to different modes of communication. The solution to these problems will require carefully study. In GISE, the generation of the grammar and lexicon from the application representation in the CO benefits the incorporation of the conceptual information necessary to solving these problems.

Finally, the interfaces generated by GISE could also be adapted to a new area, that of NLIs for systems searching the Internet. Future work will study how the organization of knowledge proposed in this thesis relating specific conceptual knowledge to its linguistic realization could be adapted to generating NLIs to internet search systems.

9.3.3 Using knowledge bases for purposes other than communication

Current work has focused on the study of an appropriate representation to support general relations between the conceptual knowledge involved in the communication with KBSs and its linguistic realization. The representation of the general relations between different knowledge bases and, in particular, between the conceptual and

linguistic knowledge bases, is a problem common to different types of applications. Adapting the proposed design for purposes other than communication is research work to be undertaken in the future.

The organization proposed in this work could easily be adapted to different types of applications in NL processing, such as generating explanations, descriptions and summaries describing applications and domains.

The separate organization of the knowledge bases allows the use of each one individually. Each of the knowledge bases, the CO, the syntactic-semantic taxonomy, the LO and the control rules could be adapted in other systems and for purposes other than that of supporting NL communication.

In particular, the representation of the general relations between different ontologies could be applied to other purposes. Problems concerning relations between different ontologies, such as integrating ontologies and particularizing ontologies, still have to be solved. The formalism used in this thesis to establish relations between knowledge in the CO as well as between CO and LO knowledge could be adapted to relate different types of knowledge represented in ontologies. The use of a declarative and high level form of describing relations has proved a powerful way to work with knowledge represented in different ontologies. The definition of control rules capturing general relations between knowledge represented in more than one ontology is also an interesting line of research. These rules could be defined for different purposes, such as integrating and particularizing ontologies.