International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 3 (2011) pp. XXX-XXX

The use of Domain Ontologies for Improving the Adaptability and Collaborative Ability of a Web Dialogue System

Marta Gatius and Meritxell González

Software Department, Technical University of Catalonia,
Jordi Girona Salgado 1-3, 08034 Barcelona, Spain
{gatius,mgonzalez}@lsi.upc.edu

***Abstract*: In this paper we describe the use of domain ontologies in a mixed-initiative web dialogue system for improving both its adaptability and its collaborative ability. Dialogue systems guiding the user when accessing the web services can enhance web usability, however they are expensive to develop and difficult to adapt to different types of web services. The use of the web service knowledge model as the basis to define the semantics of information exchanged by the system components facilitates the integration of the different types of knowledge involved in communication and provides a unified system easier to apply to new web services. Furthermore, the representation of the web service knowledge according to an ontology enhance the reasoning capabilities.**

***Keywords*: Dialogue systems, web assistants, domain ontologies, communication task modeling, dialogue plans, adaptability.**

I. Introduction

As web is becoming more central to our daily activities the need of web assistants enhancing its usability increases. In many situations we can still find it difficult to access web content because there is a large amount of documents and web services in different languages, and they change rapidly. In order to fulfill the user's need web systems need to be more collaborative and adaptable to different domains and users. Interfaces supporting natural language (henceforth, NL) mode seems useful for accessing the changing information sources in the web because they can handle friendly and collaborative communication. NL modes (text and speech) can support several types of interactions, such as menus (the user is asked to choose an option), form filling (specific information is asked to the user) and commands (the user can express an order). Furthermore, language can support phenomena not supported by other modes of communication, such as references to previously evoked entities. Besides in a simple sentence we can express questions that would require several interactions using other modes, i.e., "*Opera concerts on Saturday night for the next two months*".

Dialogue systems (henceforth, DSs) are focused on achieving a friendly conversation when guiding the user accessing a specific application or domain. DSs have evolved from simple systems interacting with users in a very restricted way (asking the user simple questions) to complex DSs supporting richer interactions, more similar to those among persons.

The improvement of NL and speech technologies has made possible DSs capable of dealing not only with dialogues in which the system drives the interaction but also with mixed-initiative dialogues, in which both the user and the system can take the initiative. However, inferring the user's intention becomes complex when the user controls the dialogue because his interventions are not restricted to previous system's questions, and the user can even change the conversation topic. The procedures developed for modeling simple interaction have limitations for modeling user's initiative. For this reason, many flexible DSs use more complex approaches, such as domain knowledge, information about the user, general dialogue mechanisms (such as clarifications and corrections) and communication plans (defining the steps to follow to solve a domain problem).

Adapting existing mixed-initiative DSs to guide the user accessing web applications present several challenges. One of the main limitations is that practical DS are mostly adapted to the functionality of a specific application and are not easily adaptable to new applications. The main reason is that DSs adapted to the communication needs of a particular application improve their performance because mistakes and ambiguities are reduced. Although there are mixed-initiative dialogue systems, having reusable components of discourse management, NL and speech, most of them are developed for specific type of application and its adaptation to other types of applications is still difficult. The main problem is that adapting the DS implies the modification of heterogeneous knowledge sources.

The use of the application knowledge model as the basis to define the semantics and the content of information exchanged by the system components facilitates the integration of the different types of knowledge involved in communication and provides a unified system easier to apply to new applications. Increasingly, research DSs incorporate ontologies to model the application knowledge. The main advantage of organizing conceptual knowledge according to an ontology is that it favors the system reusability and it also enhance the reasoning capabilities.

In this paper we focus on the use of domain ontologies in order to improve both the adaptability and the collaborative ability of a mixed-initiative web DS we have previously developed for guiding the user when accessing the web. The system was designed to guide the user when accessing different types of web services in several languages. One of the main differences between our work and related work on complex conversational systems is that our study is focused on how the user can be assisted when accessing different types of web services and information. For this purpose, we have studied the most appropriate representation of the different types of knowledge involved in the communication that take place when guiding the user to access the web services: domain-restricted linguistic and conceptual knowledge, service descriptions, general communication tasks, dialogue strategies, as well as information about the user. A complete description of the system design is given in [1].

A prototype of the DS had been previously implemented supporting textual access in Spanish and Catalan. The prototype simulated access to two web services of different type: an informational service on cultural events and a transactional service on large objects collection. The results of the evaluation of the prototype can be found in [2] .

Recent research has been focused on using domain ontologies for improving both the DS adaptability and its collaboration ability. In a previous article [3] we described how the use domain ontologies could be used to reformulate the user's query (in case no results satisfying user's requirements are found) and to summarize web information. This article is an extended description of the improvements on the adaptability and collaborative ability of the system.

II. Previous work

The problem of using ontologies for enhancing the web accessibility has focused many relevant works in several fields. Although our approach is based on the integration on NL, dialogues and ontologies it deals with problems common to approaches using different techniques. In this section we compare our work to other proposals. In the first subsection we relate our work to other approaches using ontologies to enhance web accessibility without using dialogues techniques. The second subsection compares the system we have developed to other DSs integrating ontologies.

A. Using ontologies to enhance web accessibility

There are relevant works on using domain knowledge in order to improve interaction with the user when accessing web sources. Because there are many different aspects to be considered in this field, we have selected several examples of relevant works that deal with problems considered in our proposal: presenting the user relevant domain information to be included in the query [4], reformulating the user's query [5], integrating ontologies and web resources to enhance semantic search capabilities [7] and representing formally complex query and the answer models [8].

The approach proposed in [4] to guide the user when building the query of a search engine consists of an interface presenting a set of menus containing relevant domain data that can be easily adapted. Although this approach can be appropriate for a search engine could present limitations for accessing other applications that need richer interaction.

The problem of query reformulation when no results satisfying users' requirements are found has also focused many works. The approach followed in [5], applied to the discovery of semantic web services consist of extending query terms using ontologies, as in our work. In this work query terms are extended using WordNet, an ontology including descriptions and relations of all words in several languages. In our work (as in [6]) query terms are extended using domain ontologies, because our approach consists of using domain-restricted resources.

The enhancement of semantic search capabilities by facilitating the integration of several web sources is also an active line of research. The work described in [7] in an interesting proposal this line, using common knowledge about city entities in web sources to enhance semantic search in a domain-specific repository. In our system web taxonomies have also been used to improve the interaction with the user when accessing a web service. For example, a furniture taxonomy (obtained from *ikea* web site) has been integrated into the domain ontology to guide the user accessing a web service to state a date for large objects collection.

The formal representation of complex queries dealing with information obtained from different web sources is a difficult problem. The work presented in [8] describes a formal model to represent those complex queries and their answer models. Although our DS currently do not support queries as complex as those described in that work, that involve several complex operations (together with their interrelations), we could integrate that work to foster the our system capabilities.

Our DS provides access to web services previously selected. Future improvements on the DS capabilities could also be achieved by incorporating semantic web techniques, such as those proposed in [9], using ontologies to facilitate query processing and optimization, as well as those on service discovering and composition [10].

B. Related work on Dialogue Systems

There are several DSs using application knowledge model representing domain-specific knowledge ([11]). As mentioned before, the use of the application knowledge model as the basis to define the semantics of information exchanged by the system components facilitates the integration of the knowledge bases used as well as its adptation to new applications. Thus, the use of ontologies representing the application model is especially appropriate in systems supporting several modes of interactions and several languages (such as the SMARTKOM system, [12]). It also facilitates the incorporation of advances artificial intelligence techniques (such as the ACTIVE system [13]).

Ontologies representing the domain-specific knowledge have been used for years in systems supporting textual interactions (as well as in other text processing applications) to facilitate the semantic interpretation of sentences, by relating the lexicon to the concepts in the ontologies. More recently, ontologies in DSs have also been used to achieve a friendlier interaction. By using domain ontologies DSs may infer default and misunderstood values from user intervention as well as provide descriptions of domain concepts. The use of ontologies can also improve dialogue in several other forms: it helps to detect differences in what is expected from the user's interventions, such as under/over specification (corresponding to hyperonym or hypononym) and to improve dialogue coherence by reordering system's questions (as described in [14]).

We also propose the use of the domain ontologies could be used to generate summaries describing the results, in case that too many elements satisfying the user's goal are found. This problem has been studied by different works, following several approaches. Most relevant of these works propose the distribution of the results clustered for all the possible query parameters (restrictions), as described in [15]. Other proposals also include the use of a user model to express differences in an effective way, as in [16]. These strategies could have limitations on web systems, because the amount of information is huge and clustering the set of results could be costly. Instead, we propose the use of the domain knowledge in the DS ontologies to select the most appropriate information to generate a summary of the partial results.

An example of related work using domain ontologies to present the most appropriate information is the DS described in [17], the main difference with our work is that the selection of the ontology knowledge in that system is done using information about the user.

Our DS also incorporates information related to the user's expertise to adapt the dialogue strategies. The DS incorporates an adaptive module that evaluates how well the communication is doing and adapts dynamically the dialogue strategies (initiative and confirmation policies) considering this information. This module could be extended by incorporating additional information on the user profile, useful information to be considered on this line are those described in [18] (on using user and task models to adapt an interface for a complex aplication) and [19] (on user models on Web navigation) and [20] (on personalized web services).

III. The Dialogue System

As mentioned in the introduction the DS we have developed assists the user when accessing the web. The current prototype guides the user to perform two tasks of different type: searching for information on cultural events and stating a date for furniture collection. This section gives an overview of the system.

A. The System Architecture

The DS we have developed follows a modular architecture, as shown in Figure 1. It consists of four independent components: the NL understanding module (NLU), the NL generator (NLG), the DM and the task manager (TM). The user's sentence is first processed by the NLU and the resulting interpretation is passed to the DM. The DM is responsible for controlling dialog to help the user to achieve his goals. For this purpose, the DM firsts infers the user's intention from the semantic interpretation of his intervention, the dialogue context and domain knowledge. Then, the DM determines the next system's actions: generation a system's respond or accessing the web service. The access to the web service is controlled by the TM. The generation of the system's responds is done in two steps: the semantic content is generated by the DM and passed to the NLG, which generates the system's NL sentence. The DM uses the adaptive submodule to determine dynamically the initiative degree of the system's respond. For example, when there is any communication problem the system takes the initiative, in case the user answers giving additional relevant data, then the initiative is given to the user. The adaptive submodule determines the degree of the DS initiative following a dialogue model using relevant cues about how well the communication is doing.

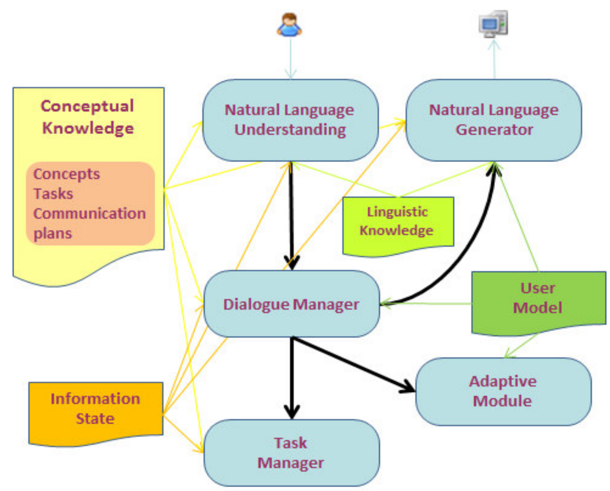

Figure 1. The architecture of the dialogue system

B. The general knowledge bases

The domain-dependent knowledge used by the general modules is represented in separated knowledge bases. Those knowledge bases consists of the domain ontology used across all modules, the communication tasks used by the TM, the general communication plans by the DM as well as the linguistic knowledge bases used by the linguistic modules. In particular, the linguistic knowledge consists of the grammar and lexicon used by the NLU and the patterns used by the NLG.

Adapting the DS to a new service consists of adapting these general knowledge bases. First, considering the operations of the new web service, instances of the corresponding communication tasks are generated and the domain ontology is defined. This first process is done manually. Then, the communication plans are automatically generated by adapting general plans to these communication tasks. Finally, the general linguistic knowledge is adapted to the communication tasks and domain ontology to obtain the domain-restricted linguistic resources. Thus, the adaptation of the data used by all modules to the service specific knowledge facilitates the exchange of information between modules and improves the DS functionality.

Next section gives a more completed description of these general knowledge bases and how they are adapted for a specific web service.

IV. Adapting the general knowledge bases

This section describes the incremental process of acquiring the domain-restricted knowledge involved in communication: the domain ontologies, the communication tasks, the dialogue plans and the linguistic resources.

A. The domain ontology

An ontology representing knowledge related to the specific web service has been incorporated to improve both the adaptability of the DS and its functionality. This ontology consists of top concepts representing general knowledge. For each web service, the domain concepts related to the web services operations and their attributes are incorporated as instances of the general concepts. Knowledge in the ontology is used as the basis to define semantics of information exchanged by the system components.

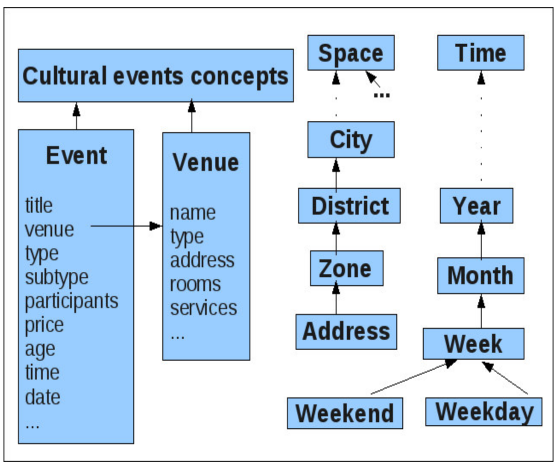

Figure 2. Concepts in the cultural events domain

Let's consider, for example, the domain on cultural events. When applying our prototype to this domain we have studied a web service giving information on the city of Barcelona. This service supported only one operation: consulting information related to the cultural events that take place in the city. The operation input parameters were **what, where, when** and **type**. Advanced search included other optional parameters (I.e., **age**). This web service operation is linked to the concept **Event** in the domain ontology, shown in Figure 2. The parameters of this operation are represented as the attributes **title**, **venue**, **type** and **date**. Other related attributes describing the concept (**participants, price, age** and **schedule**) are also included because they often appear in the dialogues related to this domain. Additionally, the related concept **Venue** is also represented because it is involved in many dialogues in the domain on cultural events. The attributes describing the concept **Venue** are the **name**, the **address**, the **services** and the **rooms**. The attributes **date** and **address** appearing in the description of these concepts are linked to the representation of the general concepts **Time** and **Space**, respectively.

If a new web service on cultural events is accessed by the DS, the same domain representation could be used. In case that additional domain information is needed, the domain ontology would be extended. The domain ontology can thus be used to integrate information obtained from several web services. At run-time, several strategies can be used to select and order the information that is presented to the user.

The domain ontology could also be used to facilitate web service composition. The domain concepts appearing in a web service can be related to other concepts appearing in a different web service, not providing the same operations. For example, the information on cultural events given by informational web services in this domain usually includes the venue where the event takes place. Additional data on how to get there (obtained from other web services) could also be given to the user.

B. The communication tasks

The application-dependent knowledge appearing in communication is related basically to the operations the service can perform, that is, the data needed from the user to perform the task and the resulting information.

In order to facilitate the incorporation of this knowledge for a new web service we have defined general models for the operations appearing in most common web services. For this reason, we have studied the operations performed by several transactional and informational web services. Notice than one web service may support more than one operation. Three different operations have been identified for transactional services: **submission** (controlling the transaction and presenting the results), **cancellation** (controlling the cancellation of a transaction previously done) and **information** (giving related information). For example, it can be able to make a reservation, to cancel a previous reservation and to give information about the service itself. Main operation performed by informational service consists of giving information.

Considering the information that appears in communication when guiding the user to perform these operations we have defined three different types of tasks: **SubmitForm**, **ObtainData** and **FindList**. The first task represents the operations making transactions (submission) and cancellations. The two last task models are related to the operation giving information: **ObtainData** (returning the description of an item satisfying a given specification) and **FindList** (returning a list of items).

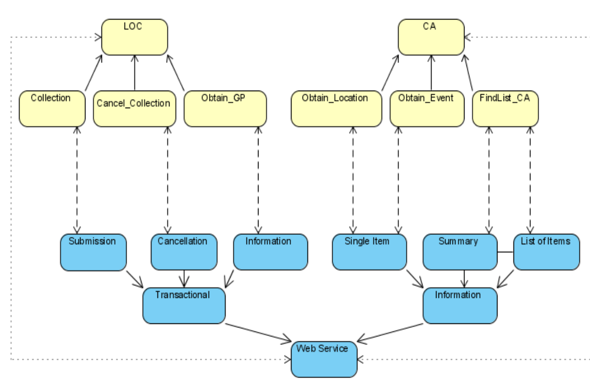

Figure 3. Operations in web services

Figure 3 shows the classification of the operations of the two web services incorporated into the prototype: the informational service giving information on cultural events

(CA) and the transactional service stating a date for large objects collection (LOC).

Each task model includes input/output parameters, constraints and conditions. These constraints apply at different levels: attribute value, attribute relations or overall task. The TM has to consider all these constraints in order to update the parameters.

Each type of task is processed by a specific algorithm. These algorithms are very simple. The algorithms for the **SubmitForm** and **ObtainData** tasks search for a suitable list of parameters and check that no conflicts among the parameters prevent the system from accessing the service. The algorithm for the **FindList** process is only aware of the number of results.

The description of a web service can consists of one or more operations. When a web service is incorporated, each web service operation is described as an instantiation of one of the tree task models. This information is stored in XML files. At run time, those instantiations are used by the TM. Once the user intervention has been interpreted, the system identifies which service and task the user needs to execute. Then, it completes the task information considering the data that appear in the dialogue. Next, the TM accesses the web service and finally, it processes the results.

As mention before, the domain ontology representing service operations and parameters is used as the semantic bases to interchange information between modules. This information is incorporated into the lexical entries used by the NLU. Thus, the semantic interpretations of the user interventions are represented in base to the service task and domain ontology. Using this information the TM can identify the service and the task that has to be accessed.

Let's consider, for example, the user's first intervention is "*Which concerts are there in the Stadium on Saturday?*". The TM has to identify the service to be accessed (**cultural events**), then the task (**find list of events**) and the values of the input parameters (**concert, Stadium, Saturday**). Then, it executes the task. In case no items satisfying the user's restriction are found, several strategies to help the user to find desired information can be followed, as described in next section.

Next dialogue actions depend on the data resulting from the execution of the TM: request the user to give more information, when needed; show the results if the service was correctly accessed; suggest the user new searches, etc. All these actions are controlled by the DM, as explained in next subsection.

C. The dialogue plans

The interaction with the user is controlled by the DM. It controls the information that has to be asked and given to the user and how this process has to be. In our DS, the DM uses models of dialogue based on communication plans to perform these tasks. Those models consider the user's utterances as communication actions that are part of a plan that has to be uncovered by the system.

We consider that the communication taking place when guiding the user to access web services mainly consist on information related to web service tasks and parameters. For this reason, the plans basically describes what and when the information has to be asked and presented to the user to perform service tasks and present results.

In order to provide flexibility, the plans in our DS are not structured as flats lists of actions but as sub-plans that are accommodated considering context information at run-time. Communication plans can be decomposed into actions and subplans. Possible plan actions are those asking the user information about the value input parameters of the service operation as well as giving the value of the output parameters.

For efficient reasons, in our DS (as in many practical DSs) these communication plans are generated statically, when a new service is incorporated, and they are stored in libraries. The dynamic generation of plans during communication increases complexity and time processing.

The main drawback that the generation of the plans for each web service presents is that these plans have to be manually built. To solve this problem we have created general templates that facilitate the plan generation for a new service. Basically, those general templates are related to the general communication task models used in our DS to describe the operations involved in transactional and informational web services (**SubmitForm, ObtainData** and **FindList**). Although the collection of the information the web service needs from the user and the presentation of the results is involved in the all the tasks, these processes differ for each task.

The processes of obtaining information from the user and presenting results in the tasks involved in transactional services are quite simple. The step of collecting the information consists of asking the user to give the values of a set of specific slots, needed to fill a form. Then, the information obtained is passed to the TM and the transaction results have to be presented to the user, usually as a text sentence (i.e., "*The transaction has been successful*").

For informational services those processes are more complicated because different situations have to be considered. The process to obtain information from the user to restrict the search is not always simple. Although usually there is a set of possible parameters to restrict the search, unlike in the case of transactional services, the user may choose to give the value of only a subset of those possible parameters. The user's goal when looking for information on the web is not always clear and can even change during the communication process (even from one turn to the next). Besides, there is a lot of information on the web. For this reason, there is a need for collaborative systems that assist the user when formulating the queries and present the information found in a clear form.

Several considerations have to be done when presenting the results obtained from informational web services. First, the system has to generate content, then its format. Additionally, the system has to apply several strategies to follow when no items found and when too many items found.

The DM has to be decided the specific data about the items found that has to be presented: when a complete or partial description is more appropriate, as well as the specific data and the format. For instance, when the DS assists the user accessing the web service giving information about cultural events, different data about a particular event can be given to the user: the event name, the location, the data, the time, the price, etc.

In the specific case that the DS has found no items satisfying the user's need, the system could guide the user to reformulate the query or, alternatively can do it using different types of knowledge, as described in next section.

Several strategies can also be followed in the specific case that the number of results obtained is higher than a predefined threshold. Simple responses could present all the results in several steps. But more cooperative responses could present a summary with partial results and suggest the user some additional information to limit the search.

We propose the use of domain ontologies in the two specific situations mentioned above: when no results satisfying user's requirements are found and when there are many results.

D. The linguistic knowledge

The language modules in our DS (the NLU and NLG) use specific domain-restricted linguistic resources. The use of domain-restricted linguistic resources limits the space of possible interpretations in language processing but increases robustness and reduces the run-time processing. In our DS, the domain-restricted linguistic resources consist of linguistic structures and terms needed to express questions and answers related to the specific web services and their parameters.

The use of a general description of web services plus an ontology representing domain concepts and relations favors the semi-automatic adapting of general lexicons and linguistic structures. Domain taxonomies have also been used in DSs to obtain all possible terms that can appear in the user intervention when giving specific information. Existing general lexical ontologies, such as *EuroWordnet*, could also be used to obtain related terms, but a lot of work is necessary to choose the appropriate terms for a particular domain among all possible related terms.

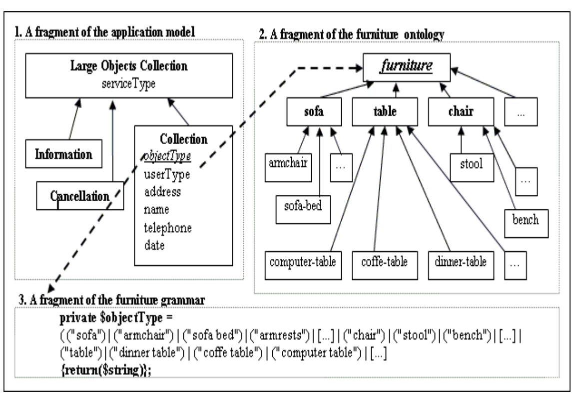

Figure 4. Using ontogies to obtain linguistic resources

Let us consider, for instance, the service for collecting large objects from houses. This service needs information about the specific type of object the user wants to get ride off. A furniture taxonomy has been used to obtain the lexical entries representing the different types of objects. For this specific service, the objects related by the *is-a* relation have been represented as synonyms, linked to the same concept furniture, because is no need for more specific distinctions considering the specific type of furniture. If needed, the DS could use the taxonomy to obtain more precise information. For example, it could detect a hyperonym in the user's intervention describing the object. In case the user says *I want to throw out some furniture* and the concept *furniture* is classified in the objects ontology as a hyperonym of the information the service needs, the DS would ask the user to be more specific.

V. Using the domain ontology to enhance the collaborative ability

A. Reformulation of the user's query

We have defined a set of rules to reformulate the user's query to obtain a more general one (to relax the query constraints) and thus obtain more results. The main advantage of this approach is that those general rules can be reused across domains. One example of these rules is the following:

Rule1. *A conceptual class can be replaced by its upper class.*

U1: I want to go to the Opera tonight.
S1: Sorry there are no opera concerts tonight. But there is another classical concert: The Requiem of Brahms in the Auditori

Dialogue 1. Replacing *Opera* by *classical concerts*

Dialogue 1 illustrates the use of Rule1 in the domain of cultural events. In this scenario the user is looking for opera concerts that takes palce at a specific date (tonight) and the system does not find any. If the system would have asked the user to give a more general description of the object to be found it could have taken the user several turns to redefine his goal. In order to achieve a friendlier interaction, our DS is able to perform this process automatically.

As described in previous section, in the domain ontology the web service operation is linked to the concept **event** and the operation parameters are represented as the attributes of this concept. In Dialogue 1, the user query includes two restrictions (operation parameters): the **event type** (*opera concerts*) and the **event date** (*tonight*). Applying Rule 1, the system extends the value of the query restriction **event type** (*opera concerts*) to its upper class (*classical concerts*).

Specific rules adapting the general Rule1 have been defined for the general concepts **address** and **date**, appearing in many web services. Those rules are following:

Rule 1.1. *The attribute* ***address*** *can be replaced by the more general attributes* ***zone*** *and* ***district****.*

Rule 1.2. *The attribute* ***date*** *can be replaced by the more general* ***week day****,* ***weekend****,* ***week*** *and* ***month****.*

U1: I want to see Opera Aida next Sunday.
S1: Sorry. There is performance of Opera Aida next Sunday. But there is one next Saturday.

Dialogue 2. Replacing *Sunday* by *weekend*

Dialogue 2 is illustrates the application of Rule 1.2. There are no results satisfying the restrictions of **title** (*Aida*) and **date** (*next Sunday*), but a result is found when the restriction *Sunday* is replaced by the more general *weekend*.

Notice that Rule 1 can be applied to any of the restrictions in the query. For each specific domain the predefined order of relevance of the operation parameters (possible restrictions of the query) is defined and the restrictions in the query are extended considering this order (starting by least relevant). The predefined ordered set of parameters (represented as attributes) in thedomain of cultural events is: {**title**, **age, date, type, venue**}.

The following rule is another example of a general rule that can be adapted to each specific domain:

Rule 2. *If the query includes two or more restrictions then one of the restrictions is deleted until results are found.*

Dialogue 3 illustrates the use of Rule 2. In this example, a set of results partially satisfying the description given by the user is presented, there are no theater plays for children on Sunday, but other events for children on the same date are presented. The user's query includes four restrictions, that is, the value of four attributes: **type** (*theater*), **age** (*for children*), **date** (*on Sunday*) and **venue** (*Auditori*). Then, the system applies the Rule 2 and eliminates the attribute **venue** (notice that the attributes **age, date** and **type** are more relevant in the cultural domain). Results are found for the resulting the updated query: "*Events for children on Sunday*".

U1: Are there musical concerts for children on Sunday in the Auditori?
S1: Ok. I'm searching. I'm sorry, there are no for children on Sunday. But, there are two events for children on Sunday.
Results from 1 to 2 follows:
1: Musical festival on Ciutadella
2: Sónar Kids

Dialogue 3. Deleting the venue restriction (Auditori)

Specific rules for specific domain concepts could also be incorporated. The advantage of defining relaxation rules adapted to a particular domain (and not to a particular service) is that they can be reused across web services related to the same specific domain.

B. Generation of summaries

In case the DS finds many results satisfying user's requirements, it presents a summary of them. In order to generate the appropriate summary we have defined an ordered set of the attributes of the ontology concepts representing the service operation that could be considered to ellaborate a summary. This general ordered set of attributes that can be adapted for each specific domain. This set is not the same that the one used in the relaxation rules. The ordered set consists of the following attributes: first **types** (i.e., types of event, types of restaurant), second **subtypes**, third **locations** and finally, **time.** Attributes in this set can be replaced by their upper classes. Additionally, for each domain this set attributes can be extended. For example, the information on cultural events can be summarized using the following ordered set of attributes: { **event type, event subtype, zone, district}**.

We have also stated several restrictions on attributes that could be considered to produce a proper summary. Examples of attributes that could not be considered would be those appearing in the query constraints (explicitly described by the user) and those having an open set of possible values.

U1: cinema today
S1: OK. I am searching. There are many films today.
6 in the dramatic genre,
12 in the action and terror genre,
8 in the childlike and adventure genre and
6 in other genres.
Which genre are you interested in?

Dialogue 4. An example of summary

. Dialogue 4 shows an example of interaction that takes place when the DS assists the user to access information on cultural events. In this example, a summary has been generated because the results satisfying user's goal were (*movies for today*) were too many to be presented in one turn. Using the ordered set of attributes information has been summarized considering the **event subtype** (being the **event type** *cinema*).

Conclusion

One of the main limitations of adapting mixed-initiative DSs to assist the user when accessing the web is the high cost of adapting complex DS to different types of web sources (different domains, languages, presentations, etc.). In this paper we propose a general organization of the different types of knowledge involved in a DS assisting the user when accessing web services: service modeling, domain concepts, dialogue plans and linguistic resources. This organization in separated, related general knowledge bases favors the adaptability of the DS to new web services, languages and users. We also describe how the use of domain ontologies enhance both the adaptability of our DS and its collaborative abilities.

Although the prototype implemented only supports text access in Spanish and Catalan to two web services, the DS design facilitates its extension to access other web services, languages and modes of communication (such as voice).

Future work will also include the study of how our DS could assist the user in more complex tasks, implying, for example, the composition of several transactional web services.

References

[1] M. González, "Digui: A flexible dialogue system for guiding the user interaction to access web services," Ph.D. dissertation, Technical University of Catalonia, Barcelona, Spain, 2010. [Online]. Available: http://www.lsi.upc.edu/http://www.lsi.upc.edu/ _mgonzalez/thesis_meritxell gonzalez.pdf.

[2] M. Gatius and M. González, "A flexible dialogue system for enhancing web usability," in Proceedings of the 18th international conference on World wide web, WWW '09. New York, NY, USA: ACM, 2009, pp. 1167–1168.

[3] M. Gatius and M. González. Using Domain Knowledge for Fostering the collaborative Ability of a Web Dialogue System. In the proceedings of the 7th International Conference on Next Generation Web Services Practices (NWeSP), pp.129-134, Oct. 2011.

[4] Susmitha Dey, Siby Abraham. User Interface For A Search Engine: A Customized and Multi-domain Approach. International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Volume 4 (2012) pp. 169-179.

[5] Dalila Mekhzoumi, Hassina Aliane, Omar Nouali. Queries reformulation for the discovery of semantic web services. In the proceedings of the 7th International Conference on Next Generation Web Services Practices (NWeSP), October . 2011.

[6] S. Varges, F. Weng, and H. Pon-Barry, "Interactive question answering and constraint relaxation in spoken dialogue systems," Natural Language Engineering, vol. 15, pp. 9–30, January 2009.

[7] Weisen Guo and Steven B. Kraines. Integrating Knowledge of City Entities Extracted from DBpedia and GeoLite into the EKOSS Failure Cases Repository to Enhance Semantic Search Capabilities. International Journal of Computer Information Systems and Industrial Management Applications.ISSN 2150-7988 Volume 3 (2011) pp. 780 -787.

[8] Brahim Batouche Yannick Naudet Frederic Guinand. Web Service Complex Request Ontology and it Answers Models. In the proceedings of the 7th International Conference on Next Generation Web Services Practices (NWeSP), Oct. 2011.

[9] Sanjay Kumar Malik, SAM Rizvi. A Framework for SPARQL Query Processing,Optimization and Execution with Illustrations. International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Volume 4 (2012) pp. 208 -218.

[10] Hong Qing Yu, Stefan Dietze, Carlos Pedrinaci and Dong Liu. A Linked Data compliant Framework for Dynamic and Web-scale Consumption of Web Services. International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Volume 3 (2011) pp. 796-803.

[11] J. Allen, D.K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. Toward conversational human-computer interaction. AI magazine, 22(4): 27-38, 2001.

[12] D. Sonntag, R. Engel, G. Herzog, A. Pfalzgraf, N. Pfleger, M. Romanelli, and N. Reithinger, "Smart web handheld –multimodal interaction with ontological knowledge bases and semantic web services," T.S. Huang et al. (Eds.): AI for Human Computing, LNAI 4451, pp. 272-295, 2007. Springer-Verlag Berlin Heidelberg.

[13] D. Guzzoni, C. Baur, and A. Cheyer, "Active: A unified for building intelligent web interaction assistants," in IAT Workshops, 2006, pp. 417–420.

[14] D. Milward, "Ontologies and the structure of dialogue," in theProceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (Catalog), 2004, pp. 69–77.

[15] J. Polifroni and M. Walker, "Intensional summaries as cooperative responses in dialogue: Automation and evaluation," in Proceedings of ACL'08 - HLT. Columbus, Ohio: Association for Computational Linguistics, Jun. 2008, pp. 479–487.

[16] A. K. Winterboer, M. I. Tietze, M. K. Wolters, and J. D.Moore, "The user model-based summarize and refine approach improves information presentation in spoken dialog systems," Comput. Speech Lang., vol. 25, pp. 175–191, April 2011.

[17] M. E. Foster and M. White, "Assessing the impact of adaptive generation in the comic multimodal dialogue system," in Proceedings of the IJCAI 2005 Workshop on Knowledge and Reasoning in Practical Dialogue Systems, 2005

[18] Victor Alvarez-Cortes, Víctor H. Zárate Silva, Benjamín E. Zayas Pérez, Jorge A. Ramírez Uresti and Agustín Quintero Reyes. User and Task Models Impact on an Adaptive User Interface for the Startup of a Power Plant. International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Volume 4 (2012) pp. 300-30.

[19] Gustavo de la Cruz Martínez and Fernando Gamboa Rodríguez. User Interaction to Model User Comprehension on the Web Navigation. International Journal of Computer Information Systems and Industrial Management Applications. ISSN 2150-7988 Volume 3 (2011) pp. 878-885.

[20] Z. Jarir, M. Quafafou and M. Erradi. Personalized Web Services for Web Information Extraction. International Journal of Web Services Practices, Vol. 5, No.1 (2010), pp. 22-31.ISSN 1738-6535.

Author Biographies

Marta Gatius as born in Barcelona. She is an assistant professor in the Software Department at the Technical University of Catalunya (Spain). Her research interests include natural language processing and generation, natural language interfaces, dialogue systems, knowledge representation, ontologies and information extraction from the Web. She received her Ph.D. from Software Department at the Technical University of Catalunya in 2001.

Meritxell Gonzalez was born in Barcelona**.** She is an assistant professor in the Software Department at the Technical University of Catalunya (Spain). Her research interests include natural language processing and generation, natural language interfaces, dialogue systems and translation. She received her Ph.D. from Software Department at the Technical University of Catalunya in 2011.