Using Domain Knowledge for Fostering the Collaborative Ability of a Web Dialogue System

Marta Gatus  Meritxell González
Software Department
Technical University of Catalonia
Barcelona, Spain
e-mail: {gatius,mgonzalez}@lis.upc.edu

Abstract—In this paper we describe the use of the domain ontologies for improving both the adaptability and the collaborative ability of a flexible web DS. We have developed a multilingual DS for guiding the user when accessing several types of web services. The use of domain ontologies facilitates the adaptation of the system to new web services as well as the generation of more collaborative responses.

Dialogue systems; domain ontologies; communication task modeling

I. INTRODUCTION

Friendly natural language (NL) interfaces that assist the user when accessing the web seem a good opportunity to enhance web usability and accessibility. NL systems can support friendly interaction in different languages through several channels. Additionally, standard languages and architectures, such as those defined by the Voice Browser Activity (http://www.w3.org/Voice/) and the Multimodal Interaction Activity (http://www.w3.org/2002/mmi/) groups facilitates the development and portability web NL interfaces.

There are several research areas focused on improving web usability, such as web interaction management, question answering systems and intelligent web Dialogue Systems (DSs).

Web question/answering systems are focused on answering open domain users questions ([14]). Those systems usually combine NL and information retrieval techniques. To answer to a user's particular question they access a preprocessed data set, including, in some cases web services annotated with their semantic content.

Web interaction management is typically concerned with building (semi)automatical dialogues from static and dynamically generated web pages. There are several research lines in this area, examples of them are works on the adapting of web contents to different impairments, the definition of transformations rules for creating mixed-initiative dialogues and frameworks for incorporating multimodal interfaces to existing web applications.

DSs are focused on achieving a friendly conversation when guiding the user accessing a specific application or domain. In order to achieve a friendly interaction most research DSs support user's initiative, allowing the user take the conversation control. However, inferring the user's intention becomes complex when the user controls the dialogue (users can change easily the conversation topic,...). The procedures developed for modeling simple interaction follow pattern-based approaches that model the sequencing regularities in dialogue (such as questions being followed by answers, proposals followed by acceptances, etc.) and have limitations for modeling user's initiative. For this reason, many flexible DSs use more complex approaches, such as semantic Bayesian networks([7]), communication plans ([1]), collaborative problem-solving ([12]).

One of the main limitations the current complex DS present is that they have been usually developed for a specific type of application and their adaptation to other type is not easy because it implies the modification of heterogeneous knowledge sources. In order to solve this problem many DSs use the application knowledge model as the basis to define the semantics and the content of information exchanged by multiple components of the system. Increasingly, DSs incorporate ontologies to represent the application knowledge.

The main advantage of organizing conceptual knowledge according to an ontology is that it favors reusability as well as reasoning and therefore it facilitates the development of more intelligent web DSs. Relevant examples of intelligent web assistance systems based on ontologies are the Active platform ([6]) and the Smartweb system ([11]).

In this paper we describe our recent research on the use of domain ontologies in a DS we developed for guiding the user when accessing the web. We have previously developed a prototype of the DS supporting textual access in Spanish and Catalan to two web services of different type: an informational service on cultural events (http://www.bcn.es/) and a transactional service on large objects collection (LOC).

The results of the evaluation of the prototype can be found in ([4]) and a more complete description of the system design is given in([5]). In this paper we describe the use of the domain ontologies for improving both the adaptability and the collaborative ability of the DS.

The next section gives an overview of the DS. Then, in Section 3 we propose several dialogue strategies that use domain ontologies to generate cooperative system's responses. Finally, last section draws some conclusions.
II. The Dialogue System

A. The System Architecture

In order to facilitate the adaptability of our DS to different communication modes (voice and text), applications, users and languages it was designed following a modular architecture, as shown in Figure 1. The DS consists of domain independent components plus the knowledge bases. There are two language components: the NL understanding (NLU), that interprets user's sentence and the NL generator (NLG), that generates the system's response. The dialogue control is performed by the dialogue manager (DM), that uses the adaptive submodule to detect how well the communication is doing and adapt the degree of the system dialogue initiative. The access to the web services is performed by the task manager (TM). All these system components are described in more detailed in the following section.

The knowledge bases consist of general conceptual and linguistic knowledge that is adapted to the specific domain. The conceptual knowledge consists of an ontology describing general and domain concepts, a taxonomy of communication tasks and general communication plans. During communication, an additional data structure is used to share information on the dialogue context among all modules, the information state.

B. The Language Components

The NLU uses specific domain-restricted grammars and lexicons to perform deep syntactic and semantic analysis of the user interventions. The resulting semantic interpretation is represented in base to the domain ontology. The use of domain-restricted linguistic resources limits the space of possible interpretations but increases robustness and reduces the run-time processing. The grammars used could be transformed easily to the VOICEXML standard formalism for voice grammars.

The NLG uses patterns (sentences) containing variables that would be instantiated at run-time considering dialogue context. For this purpose, we have used a general syntactic-semantic taxonomy (we had previously developed) that relates the attributes describing the domain concepts in the domain ontology to the linguistic structures needed for its realization. The taxonomy is common to several languages (we have studied its use in English, Spanish and Catalan).

C. The Dialogue Components

The DM is responsible for controlling dialog to help the user to achieve his goals. The DM has three main tasks: interpreting user's intention (using the semantic representation of the user's intervention passed by the NLU), determining next system's action and generating the content of the system's respond.

The design of the DM follows the information state update (ISU) approach ([13]). This approach is based on a rich representation of the dialogue context (the information state) and a clear definition of the rules that update this information, basically considering the dialogue moves. The ISU approach concentrates on abstracting out the dialogue state. For this reason, even when this approach supports complex acts (such as confirmation, clarification and indirect answers), it allows simpler and more generic dialogue components, specially in systems like ours, where the application knowledge model is used as the basis to define the content of information exchanged between components.

The DM uses communication plan-based models of dialogue. Those models consider the user's utterances as communication actions that are part of a plan that has to be uncovered by the system. In our system, as in many DSs, communication plans are not generated dynamically for efficient reasons. Plans are defined and stored in a library when a new service is incorporated into the system. 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.

TM is responsible for several tasks: identifying the web specific service operation that has to be accessed, determining the possible additional information needed from the user, accessing the service and processing the results.
obtained. To perform these tasks the TM uses the description of the web service communication tasks.

A step has been made towards the adaptability of the DS by representing, in a declarative form, general communication tasks involved in communication when accessing different types of web services. Including a detailed description of all possible communication tasks facilitates the obtaining of the service restricted resources. Communication tasks are described by the input and output parameters as well as preconditions.

The general tasks have been defined by considering several web services of different type as well as examples of dialogues obtained when the system guides the user accessing them. For example, three different communication tasks have been defined for transactional services: transaction (asking for the parameters and presenting results), cancellation (asking data to cancel a transaction previously done) and information (giving related information). For informational service four different tasks have been defined: list of items (give the list of elements that satisfy user's restrictions), description of an item (give details of an element required by the user), summary (give a summary of a list of elements), alternative results (giving list of elements that satisfy partially the user's restriction).

D. Adapting the Dialogue System to a new Domain

Adapting the DS is an incremental process of acquiring the domain-restricted knowledge involved in communication, that is, the domain ontologies, the communication tasks, the dialogue plans and the linguistic resources. First, the domain ontologies are defined manually. Then, considering the web service type (such as informational, transactional,...) instances of the corresponding communication tasks are generated. The communication plans are automatically generated by adapting general plans to these communication tasks. Finally, the general linguistic knowledge is adapted to the conceptual knowledge to obtain the domain-restricted linguistic resources.

The domain ontologies are defined considering the communication that takes place when accessing one or more service in the same domain. Service operations are represented as concepts and operation parameters as attributes describing them. Related information appearing in communication is also incorporated.

Let's consider the example of the web service giving information on cultural events in the city of Barcelona. The DS process consists of only one operation: the consulting on information related to the events that take place in Barcelona. The operation input parameters are title, venue, type and date. This web service operation is represented in the domain ontology by the concept Event, shown in Figure 2. The parameters of this operation are represented as attributes. Other related attributes describing the concept (participants, price, age and schedule) are also included because they can appear in the communication. Additionally, the related concept Venue is also represented because it is involved in many dialogues in the domain of 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 incorporated into the system, the same domain representation could be used, only if additional domain information is needed the domain ontology would be extended.

Existing domain taxonomies could also be integrated into the DS domain ontologies. For example, when adapting our DS to the LOC service, a furniture taxonomy (from http://www.ikea.com/) has been used. The lexical entries representing the different types of objects the user could want to get ride off have been obtained from it.

Domain taxonomies are especially useful to obtain all possible terms that can appear in the user intervention when giving a specific information. For example, those related by the is-a relation can be used as synonyms. 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.

III. USING DOMAIN KNOWLEDGE TO GENERATE COOPERATIVE SYSTEM RESPONSES.

Ontologies in DSs can be used at run-time to improve the interaction in different forms: to detect differences in expectation in the user's interventions, such as under/over specification (corresponding to hyperonym or hyponymy),
to infer default and misunderstood values from users intervention, to provide descriptions of domain concepts, to improve dialogue coherence by reordering system's questions ([8]), to select conceptual attributes of concepts when presenting descriptions([3]).

Our work has focused on making the system's responses more collaborative when guiding the user accessing information services. The user's goal when looking for information on the web is not always clear and can even change during the communication process (i.e., from one turn to the next). Besides, there is a lot of information on the web and presenting the results of informational services can be a difficult task when too many results and when no results are found. For these reasons, there is a need for collaborative systems that guide the user to do the appropriate queries and present the results found in a clear form.

We propose the use of domain ontologies for fostering the collaborative ability of the DS when guiding the user to find web information. Our work on cooperative system's responses has been focused in two different situations: when no results are found and when there are too many results. In the first situation, the system uses the domain ontologies to reformulate the query giving a more general description of the user's goal (what is also known as query constraints relaxation). In case too many objects satisfying the user's goal are found, the domain ontologies are used to generate summaries describing the results. This section describes the dialogue strategies and their application to guide the user find information on the domain of the cultural events.

A. Suggesting alternatives

UI: Are there theater plays for children on Sunday?
S1: Ok. I'm searching. I'm sorry, there are no theater plays for children on Sunday. But, there are two events for children on Sunday.
   Results from 1 to 2 follows:
   1: Musical festival on transports
   2: Sónar Kids

Dialogue 1: An example of query reformulation

Dialogue 1 is an example of the specific case in which there are no results satisfying the user's query. 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. 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.

We have designed a set of relaxation rules (similar to the ones proposed in ([15])) that are applied over the domain concepts represented in the ontology. An example of these rules is the following:

A conceptual class can be replaced by its upper class.

The main advantage of this approach is that those general rules can be reused across domains. Specific rules adapting these general rules to the time and space representations are defined and can also be reused in many domains. Examples of those rules are following:

- The attribute address can be replaced by the more general attributes zone and district
- The attribute date can be replaced by the more general week day, weekend, week and month.

General relaxation rules can be adapted for a specific domain. For example, the following is an example of general rule:

If the query includes two or more attributes then one of the attributes is deleted until results are found.

The attributes in the query are deleted considering the predefined order of relevance. This order is established for each domain. 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.

Dialogue 1 illustrates the use of these rules. In this scenario the user is looking for childlike theater plays on Sunday and the DS did not found any play matching these conditions. The query included three attributes: type (theater), age (for children) and date (on Sunday). Then, the system applies the rule If the query includes two or more attributes then one of the attributes is deleted until results are found. Using the predefined ordered set of attributes, the attribute type is then eliminated and the resulting updated query is "Events for children on Sunday".

B. Generation of summaries

Several strategies can 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 attributes to limit the search.

Several works have studied how to build a summary and guide the user to give additional information. Most relevant of these works propose the distribution of the results clustered for all the possible attributes, as in the work of ([10]) and in ([2]). Other proposals also include the use of a user model and discourse makers (erg, connective and discourse adverbials) to express differences in a clear succinct and effective way, as in ([16]). These strategies could be integrated into our system. However, our work has been focused on providing access to the web where the amount of information and attributes is huge; we do not consider clustering the set of results because it could be costly. Instead, we propose the use of the domain knowledge in the DS ontologies to generate a summary of the partial results.
Our approach consists of the definition of an ordered set of attributes that could be considered to produce a proper summary. We have defined a general ordered set of attributes that can be adapted for each specific domain. This ordered set is not the same that the one used in the relaxation rules. The initial order set of attributes for summaries is the following: first types (for instance, types of event, types of restaurant), second subtypes, third locations and finally, time.

Attributes in this set can be replaced by their upper classes. 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 and 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.

Dialogue 3 shows an example of interaction that takes places when the user access to the service giving information on cultural events. In this example a summary using the event subtype has been generated because the results satisfying user’s goal were too many.

U1: cinema on Sunday
S1: OK. I am searching. There are too many films on Sunday.
6 of them 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 2: An example of summary

CONCLUSIONS

In this paper we describe the use of domain ontologies in a web DS for both facilitating the adaptation of the DS to new services and fostering its collaborative ability. Although the prototype implemented only supports text access in Spanish and Catalan to two web services, it design facilitates its extension to support other languages, other modes of communication (such as voice) and other web sources. Another interesting extension of the DS would be the integration of the information provided by several web services. The results obtained from web services giving similar information (i.e., about cultural events) can be easily integrated in the same ontology domain.

Future work will include facilitating the user the access to several related web services, for instance when the an event venue has been described web information on how to go there, obtained from other web sources will be presented to the user.

REFERENCES

[16] Andi K. Winterboer, Martin J. Tietze, Maria K. Wolters, and Johanna