Integrating a Discourse Model with a Learning Case-Based Reasoning System

Karolina Eliasson
Department of Computer and Information Science
Linköping University
Linköping Sweden
karel@ida.liu.se

Abstract

We present a discourse model integrated with a case-based reasoning dialogue system which learns from experience. The discourse model is capable of solving references, manage sub dialogues and respect the current topic in a dialogue in natural language. The framework is flexible enough not to disturb the learning functions, but allows dynamic changes to a large extent. The system is tested in a traffic surveillance domain together with a simulated UAV and is found to be robust and reliable.

1 Introduction

For a dialogue in natural language to run smoothly, the participants have to know the history of it. If a computer dialogue system will be able to work properly in such a natural dialogue with a human user, it has to maintain a discourse model of the dialogue so far to be able to interpret the utterances of the user in the right context. The discourse model helps the system to interpret references to utterances earlier in the dialogue. The system also need to know if an utterance shall be interpreted in the earlier discourse or if it is a start of a new dialogue with a new discourse.

In this paper, we will describe a discourse model which is integrated in a case-based reasoning (CBR) system used for dialogue with a robot. Case-based reasoning is a form of machine learning where the system stores problems and their corresponding solutions in a case base. When a new target case enters the system, it searches the case base for similar cases. When the most similar case is found, its corresponding solution is adapted to the new target case and the new solution is returned. The new target case and its solution are then stored in the case base for future use. See for example (Aamodt, 1994) for an overview.

CBR provides our dialogue system with a simple and modular design. New functionality is directly added by writing new cases and storing them in the case base. New domain knowledge similar to existing knowledge can be added to the system in a simple manner. It can directly be used by the system without any additional changes to the case base, due to the flexible and adaptable nature of the CBR design. This provides us with the facility of letting the system incorporate new information, such as new words or knowledge about the physical world, into the system. This knowledge can then directly be used by the cases in the case base, hence giving the system mechanisms for updating its own knowledge and increasing its performance. The new information can be obtained from dialogue with an operator. Because phrase matching is necessary both in CBR and in discourse modeling, in the latter to allocate incoming new phrases to the correct dialogue thread, it makes CBR and discourse modeling a suitable combination without producing any additional overhead.

We have chosen to work on the discourse model presented in (Pfleger et al., 2003) for the SmartKom project. Our structure of the discourse model as described in section 3 is highly inspired by their model. Our contribution to their work is mainly the integration of the model with CBR which is described in section 4 and 5.

2 Dialogue System

CEDERIC, Case-base Enabled Dialogue Extension for Robotic Interaction Control, is a dialogue system designed for dialogue with a physical robot, in particular the WITAS autonomous unmanned aerial vehicle (UAV). The WITAS project focuses on the development of an airborne computer system that is able to make rational decisions about the continued operation of the aircraft, based on various sources of knowledge including prestored geographical knowledge, knowledge obtained from vision sensors, and knowledge communicated to it by data link (Doherty et al., 2000). The UAV used in the project is a Yamaha RMAX helicopter which an operator can control by high level voice commands or by written commands. The operator can ask the UAV to perform different tasks and answer questions.

CEDERIC consists of a *case base*, *domain knowledge*, a *discourse module* and a *case-base manager* as shown in Figure 1. The domain knowledge contains an ontology of the world as the robot knows it, a categorization of the world items, and a grammar. The purpose is twofold. It serves as a world representation which gives CEDERIC knowledge about which buildings there are in the known

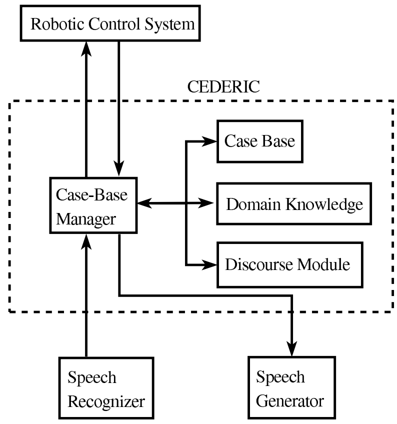

Figure 1: Architecture of CEDERIC.

world, what kind of buildings they are, where they are placed, and their attributes such as color and material. It also gives CEDERIC fundamental knowledge about which items that can be called buildings in the dialogue and which can not and provides CEDERIC with a grammar so that the system can interpret natural language. The ontological information is then used to measure the similarity of two different knowledge items. Items belonging to the same ontological class is considered similar.

The operator can choose to use either speech or text for the input to the dialogue system. The speech recognizer used is the off-the-shelf product Nuance and the speech generator used is one of the off-the-shelf products Festival or Brightspeech. When learning a new word using speech recognition, one can choose between having a considerably bigger grammar for the speech recognizer than the dialogue manager and only consider learning in the dialogue manager, or provide the new word in text form in the learning phase and then compile it into the speech recognition grammar at runtime. We have chosen the second approach where the unknown words are provided in text and the learning phase extends the grammar.

When a new sentence arrives from the operator CEDERIC looks for cases similar to the new target case. The solution to it is either an utterance in return to the user or a request to the robotic control system. The robot acts upon the request and produces a response that is catched by CEDERIC, who searches its case base and returns a message to the user. The system can manage simple cases of dialogue such as a command from the user that directly produces an answer even without a discourse model, but to be able to handle a more natural and sophisticated dialogue such as references to earlier objects and clarifying questions (where?, what?, which?, why?), a discourse model is necessary. This paper is particulary focused on the discourse model implemented in CEDERIC and how it can be used in a case-based system. For a description of the total system, see (Eliasson, 2005).

The following dialogue problems are addressed in the paper:

Anaphora references. The discourse model should be able to solve references to objects which have occurred in an earlier stage of the dialogue.

Sub dialogues. It should be able to recognize if an utterance is a sub dialogue to the present dialogue and hence should be interpreted within the limits of the current discourse or if it is the start of a new dialogue. It should also recognize a dialogue as completed which makes the old discourse no longer applicable. It should be possible to return to older non-completed dialogues which is not presently in focus.

Topic management. The discourse model should be able to figure out if it is a good moment to mention e.g. an observed event or if that utterance should wait for a better occasion when it does not disturb the present dialogue.

3 Discourse Model Design

The discourse model we have chosen to implement in CEDERIC is very similar to the one presented in (Pfleger et al., 2003). It is built up of four different objects, which is linked to one another in a hierarchical manner which constitutes the meaning of the dialogue.

The linguistic objects. These objects are furthest down in the chain of objects and thus most specific on the word level. They contain information of how the nouns in the dialogue where uttered. They could for example have been references by the word `it` or by a noun and a determinant.

The discourse objects. These objects contain the different nouns together with their attributes mentioned in the dialogue. A discourse object can also be composite. An enumeration of several objects can be seen as a discourse object representing the enumeration as such and this object contains the enumerated objects as its children. This gives CEDERIC the opportunity to understand references referring to the order of the enumerations, e.g. `the first one`. The discourse objects have a link to the corresponding linguistic object.

The dialogue objects. These objects groups the sentences and their information together which have the same direct goal. The sentence `fly to the hospital` gives for example, when it is executed, a dialogue object which groups the sentences `fly to the hospital`, `ok` and `I am at the hospital now` together. If any sub dialogues come up, they will be saved in a new dialogue object with their direct goal to clarify some matter in the dialogue. Dialogue objects contain information about the topic of the dialogue, which discourse objects that were created due to the utterances, and which future utterances this dialogue object expects to consider the dialogue or the sub dialogue completed. These expectations on future dialogue are saved in a modified *initiative-response (IR) unit* (Ahrenberg et al., 1991). IR-units in our context can, unlike the original IR-units described by Ahrenberg, contain more than two sub elements. That is because they shall also be able to represent the response from the robot when the system sends a request. The `fly to the hospital` example above shows such an example.

The global focus space. The different objects in the dialogue layer which belongs to the same dialogue, including sub dialogues, are grouped together in a top object called the global focus space. It contains information about the main topic of the dialogue, if it is ok to interrupt the dialogue and which dialogue objects that belongs to it. Each global focus space also keeps track of the discourse object last mentioned, to be able to resolve references such as `it`. This is known as the *local focus stack*. The last mentioned discourse object is said to be in focus.

To keep track of the current dialogue in focus, CEDERIC saves the different global focus spaces in a stack called the *global focus stack*. The global focus space on top of the stack is said to be the one in focus. If every IR-unit belonging to a global focus space is closed, that is, has received all its subelements, the global focus space is marked as closed and removed from the stack. Several dialogues can be open and ongoing at the same time and are thus members of the stack but only one dialogue can be in focus at the same time.

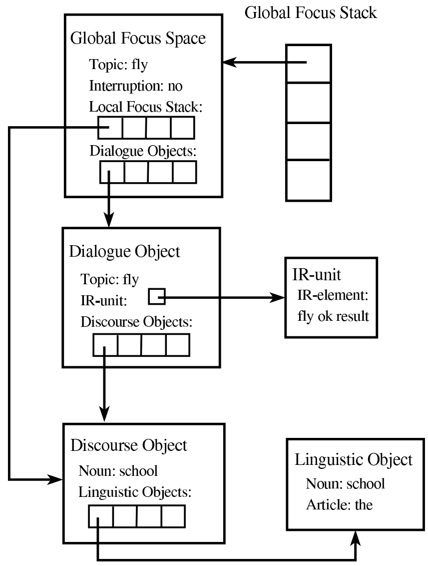

Figure 2: An example of a discourse model.

Figure 2 shows an example of how the discourse model looks like when the utterance `Fly to the school` has been executed.

4 Discourse Information in the Cases

When a new utterance enters the system, it is not only the utterance itself, but also the dialogue discourse, that tells the system how it should be interpreted. The simple answer `yes` to a question is an illustrative example of this. Without knowing the question, the answer carries no information at all. Therefore, to match a case in the case base, not only the utterance by itself but also the discourse needs to match. When a matching case is found, the system knows which information the new utterance carried and the discourse has to be updated accordingly to reflect this new information.

A case in our approach is divided into five different parts:

The problem. The problem is a description of the utterance. It contains the words and their classification according to the grammar in the domain knowledge.

The discourse information. This part describes how the global focus space in focus and its discourse object in focus should look like. It makes sure that utterances such as answers to questions are executed with the correct case.

The update according to problem. Depending on the problem, the discourse model has to be updated with the new information. This information is stored in this part.

The solution. This part contains the reaction to the problem. It can be a request to the robot to perform an action or an answer in natural language to the operator.

The update according to solution. When the solution has been executed, the discourse model has to be updated to reflect it.

If a new dialogue is started, a new global focus space with one or more dialogue objects with corresponding IR-units, one or more discourse objects, and one or more linguistic objects are created. This newly created global focus space is put on top of the global focus stack and the local focus stack of the new global focus space is populated with the new discourse objects. Possible old open global focus spaces on the global focus stack are left in the stack as they are and are still reachable although not in focus.

If the new problem is an expected continuation of an ongoing dialogue, the case returns the newly satisfied elements of the IR-unit and CEDERIC updates the above IR-units accordingly. In case all elements in the IR-unit have been satisfied, the IR-unit is closed and CEDERIC checks if the global focus space of that IR-unit only consists of closed IR-units. In that case the whole global focus space is marked as closed.

In case CEDERIC needs to ask a clarifying question to a given problem to be able to unambiguously interpret the meaning of the operator's utterance, a new dialogue is created. The new dialogue object is created in the same global focus space that matched the case, because the new dialogue is only a sub dialogue to the main one. A new IR-unit is created and possible discourse and linguistic objects are created as well. If a new discourse object is created, it is put on top of the local focus stack.

If the solution to the case is a request to the robot, the discourse model notices it and starts to expect a response from the robot.

5 Case Matching

When a new utterance from the operator or a message from the robot enters the system, it starts by classifying the included words according to the grammar. Then the case base is searched for cases with similar utterances. The current discourse in focus is matched with the discourse information saved in the case, hence a match implies that the utterance can be evaluated in the current discourse in focus.

If no case matches the new problem and the discourse currently in focus, one of the following scenarios has happened:

- The operator or the robot returns to an older open discourse.
- The operator or the robot changed topic and started a new dialogue.
- CEDERIC did not understand the new utterance either because the utterance as such is not represented in the case base or it is totally out of context and no suitable open discourse is found.

The operator is free to change subject of the dialogue at any time by starting a new dialogue or return to an old open one. If no matching case is found using the present discourse in focus and the utterance origins from

O: Fly to the school.
C: I have two schools to choose between. Which one do you mean?
O: Take off.
C: Ok.
O: Which can I choose between.

CEDERIC gets a message from the robot saying that the action take off has been successfully completed

C: You can choose between the one on Harborroad and the one on Mainstreet.
O: Fly to the hospital.
C: Ok.
C: I have taken off now.
O: What is your altitude?
C: It is 20 meters.
C: I am at the hospital now.

Figure 3: An example of dialogue topic changes and topic management between the operator and CEDERIC.

the operator, CEDERIC investigates if it is possible to match the utterance with a discourse from an earlier open dialogue. These dialogues are stored in the global focus stack. If a match is found using an old global focus space the solution to the case will be evaluated in the discourse represented by that old global focus space and it will be put in focus. It will also be updated so it correlates with the result of the evaluation of the solution. If no matching case is found this way either, the operator did not return to an earlier dialogue and the utterance is again matched against the cases in the case base. This time with no discourse in focus at all. If matching, this indicates that the operator has started a new dialogue. If a match is found, the adapted solution is evaluated with an empty discourse and a new global focus space is put on the global focus stack. If there is no match with an empty discourse either, CEDERIC matches with a default error case.

If it, on the other hand, is a message from the robot that does not match with the present discourse in focus, CEDERIC has to take the topic management into consideration. A report of a result of a performed command shall for example not be mentioned right away if the operator waits for an answer to a question. CEDERIC decides what to do by investigating the global focus space currently in focus and checks if it is ok to interrupt in the present discourse or not. If it is ok to interrupt, the same algorithm as the one for an utterance from the operator is performed, but if it is not, the message is put in a queue and is evaluated as soon as it is ok to interrupt or the present dialogue is closed.

6 Result

CEDERIC has been tested connected to a simulated UAV situated in a simulated environment. The simulated UAV can perform high level actions such as flying to a building with a certain unique identity, flying in a certain direction, take off, land, ascend and descend. It reports the result of an action and also report if it observes any buildings as it flies. It is able to answer status questions such as the current altitude, velocity and heading etc. A number of dialogues have been implemented where both references, dialogue topic changes, and topic management during the shifts have been carefully tested. Figure 3 shows an example where the operator switches dialogue topic back and forth. CEDERIC is able to keep the different discourses in mind and use the correct discourse model for every new utterance. It is also an example of a case where CEDERIC gets a message from the robot, but due to the current dialogue, the message is restrained until a suiting moment in the dialogue appears. The operator's utterances are never restrained and he or she is free to lead the dialogue as he or she wishes.

O: Fly to the church.
C: I do not know what the noun church means. Is it a building?
O: Yes.
C: Ok, where is the church you want me to fly to?
O: East of the hospital.
C: Ok, I will start by flying to the hospital.
C: I am at the hospital now.
O: Fly east and look for a white building.
C: Ok, I'll stop when I see such a building.
C: I see the church now and hover over it.
O: Fly to the school on Mainstreet.
C: Ok.
C: I am at the school now.
O: Fly to the church.
C: Ok.
C: I am at the church now.

Figure 4: An example of a teaching situation between the operator and CEDERIC.

More complex examples where the operator teaches CEDERIC new information have also been tested. An example of such a dialogue is provided in Figure 4, where the operator teaches CEDERIC a new word and a domain item, in this case a new building. The result of this dialogue is, besides the movement of the robot, an update of the grammar and the domain knowledge. As seen in the end of the dialogue, CEDERIC can successfully fly to the church after the explaining dialogue.

The tests have proven the discourse model to be robust with ability to cope with various types of operator input as partly shown above. Due to the flexibility of the information stored in the discourse model it is well suited for CBR systems and various learning strategies.

7 Related Work

Dialogue in CBR is mainly used in conversational CBR (CCBR) where the system asks the user questions which guides the search for the most similar case in the case base. Unlike in CEDERIC, the dialogue acts are not by themselves items in the case base, but rather a help to discriminate the cases. For an overview, see (Aha et al., 2001). Some work has been done in integrating a discourse model with CCBR, e.g. Branting's discourse model for conversational CBR (Branting et al., 2004). Brating's discourse model is however not integrated with the cases in the case base.

Because our CBR-system for dialogue with a robot is not a pure conversational CBR system, but has with respect to its use of dialogue more in common with non-learning dialogue systems such as (Allen et al., 2001; Rosset and Lamel, 1999), we have integrated a discourse model built on the traditional principles with CBR.

Within the WITAS project, several dialogue systems with various capabilities have been developed. The first WITAS Dialogue System (Lemon et al., 2001) was a system for multi-threaded robot dialogue using spoken I/O. The DOSAR-1 system (Sandewall et al., 2003) was a new implementation using another architecture and a logical base. This system has been extended into the current OPAS system (Sandewall et al., 2005). Our work takes a rather different approach to discourse modeling, compared to these predecessors, as we are integrating CBR techniques, but it reuses major parts or the OPAS implementation for other aspects of the system. For additional information, please refer to the WITAS web site at `http://www.ida.liu.se/ext/witas/`.

8 Conclusion and Future Work

We present a discourse model called CEDERIC which is integrated with a CBR-system for communication with a robot. We have shown how the cases updates the discourse model which gives scope for learning of new dialogues and dialogue structures within the loose framework the discourse model defines. This way, we can control which cases matches the new problem not just by comparing the problem statements but also by comparing the discourse, which gives us the opportunity to solve problems such as references, sub dialogues and topic management in a learning system.

Our implementation has been tested connected to a simulated UAV operating in a simulated environment. The resulting system is robust and allows the operator to take the initiative in the dialogue at any time without loosing track of the discourse. It has also proven easy to work with and new cases can easily be automatically generated from new target case problem, the adapted discourse description, the adapted solution and the adapted discourse update. In fact, the adapted discourse description is generated per se because it is the same discourse as the one currently in focus.

The integrated discourse model is an aid for our primary goal to design a dialogue system not only capable of learning in a restricted area but to be able to handle a large amount of utterances and advanced dialogue both from the operator and from the robot. The advanced dialogue features provides a platform for further research regarding giving the operator the opportunity to explain new domain and dialogue knowledge to the system and the ability for the system to ask for confirmation to a solution.

Acknowledgement

This research work was funded by the Knut and Alice Wallenberg Foundation, Sweden.

References

Agnar Aamodt. 1994. Case-based reasoning; foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59.

David W. Aha, Leonard A. Breslow, and Hector Munoz-Avila. 2001. Conversational case-based reasoning. *Applied Intelligence*, 14(1):9–32.

Lars Ahrenberg, Arne Jönsson, and Nils Dahlbäck. 1991. Discourse representation and discourse management for a natural language dialogue system. Technical report, Institutionen för Datavetenskap, Universitetet och Tekniska Högskolan Linköping.

James Allen, George Ferguson, and Amanda Stent. 2001. An architecture for more realistic conversational systems. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 1–8. ACM Press.

Karl Branting, James Lester, and Bradford Mott. 2004. Dialogue management for conversational case-based reasoning. In *Proceedings of the Seventh European Conference on Case-Based Reasoning*.

Patrick Doherty, Gösta Granlund, Krzysztof Kuchinski, Erik Sandewall, Klas Nordberg, Erik Skarman, and Johan Wiklund. 2000. The witas unmanned aerial vehicle project. In *Proceedings of the 12th European Conference on Artificial Intelligence*.

Karolina Eliasson. 2005. Towards a robotic dialogue system with learning and planning capabilities. In *Proceedings of the 4th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Oliver Lemon, Anne Bracy, Alexander Gruenstein, and Stanley Peters. 2001. The WITAS multi-modal dialogue system. In *Proceedings of EuroSpeech*.

Norbert Pfleger, Jan Alexandersson, and Tilman Becker. 2003. A robust and generic discourse model for multimodal dialogue. In *Workshop Notes of the IJCAI-03 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Sophie Rosset and Samir Bennacef Lori Lamel. 1999. Design strategies for spoken language dialog systems. In *Proceedings of EuroSpeech*.

Erik Sandewall, Patrick Doherty, Oliver Lemon, and Stanley Peters. 2003. Words at the right time: Real-time dialogues with the witas unmanned aerial vehicle. In *Proceedings of the 26th Annual German Conference in AI*.

Erik Sandewall, Hannes Lindblom, and Björn Husberg. 2005. Integration of live video in a system for natural language dialog with a robot. In *Proceedings of DIALOR-05*.