

A Conversant Robotic Guide to Art Collections

D. Vogiatzis*, D. Galanis†, V. Karkaletsis*, I. Androutsopoulos†‡, C.D. Spyropoulos*

*Institute of Informatics and Telecommunications, NCSR “Demokritos”
Athens, Greece

† Department of Informatics, Athens University of Economics and Business
Athens, Greece

‡ Digital Curation Unit, Research Centre “Athena”
Athens, Greece

Abstract

We present the dialogue system of a robot that has been developed to serve as a museum guide. The robot interacts with human visitors in natural language, receiving instructions and providing information about the exhibits. Moreover, being mobile, it physically approaches the exhibits it provides information about. Although the robotic platform contains many modules, including navigation, speech recognition and synthesis, our focus in this paper is the dialogue system, which supports the sessions between humans and the robot, as well as the natural language generation engine, which generates the text to be spoken. Both modules are closely intertwined and depend on an ontology represented in OWL. The robot supports dialogues in both English and Greek.

1. Introduction

This paper presents the natural language interaction capabilities of a mobile robotic tour guide developed in the Greek project Xenios.¹ The robot adopts a simple finite-state architecture for dialogue management, which is complemented by multilingual natural language generation from OWL ontologies; the latter is used to produce personalised descriptions of exhibits, currently in English and in Greek. The robot guides the visitors to the premises of the Hellenic Cosmos, the cultural centre of the Foundation of the Hellenic World (FHW).² The vision and navigation components of the robot were developed by the Foundation of Research and Technology Hellas.³

Robots are being deployed experimentally as tour guides in science parks and museums (Bennewitz et al., 2005; Chiu, 2004; Thrun et al., 1999); however, the issue of natural language has been somewhat neglected. Most of the robotic platforms acting as guides rely on prerecorded texts, which are spoken by a speech synthesiser. In contrast, in our work the focus is on dynamic text generation from domain ontologies, in the context of spoken dialogues. Spoken Dialogue Systems (SDS) typically involve speech recognition, dialogue management, language interpretation, language generation, and speech synthesis. In addition, they require domain specific knowledge and linguistic resources (e.g., domain lexicons). CSLU⁴, Dipper⁵ and Trindikit⁶ are three representative examples of software platforms that support the development of SDSs.

SDSs are divided into three major categories, depending on who controls the dialogue (McTear, 2004). In system initia-

tive SDSs, the system asks a sequence of questions to elicit the required parameters of the task from the user. In user initiative SDSs, the dialogue is controlled by the user, who asks the system questions in order to obtain information. In mixed initiative SDSs, the dialogue control alternates between the two participants; the user can ask questions at any time, but the system can also take control to elicit required information or to clarify ambiguous information. In all SDSs, a dialogue management module is required to determine what questions the system should ask, in what order and when, as well as to anchor user utterances to the discourse context (e.g., resolve referring expressions). There are three main categories of dialogue management techniques: finite state based, frame based, and plan based. The most common category are the state based ones, which use graphs. Dialogue is modeled as a network of states connected by edges. At each state, the system can perform one of the following:

1. ask the user for specific information, possibly listing expected answers (in effect offering menu options),
2. generate a response to the user, or
3. access an external application.

In Xenios, we use system initiative dialogues, with finite state dialogue management, and particular options offered to the user at each state. The advantages of this technique are faster development and more robust systems, at the expense of limited flexibility in the dialogue structure.

Recent research has started to examine the efficient use of domain ontologies in dialogue systems (Milward and Beveridge, 2003). For instance, the domain specific lexicon and the grammar of the automatic speech recognition (ASR) component can be partially derived from the ontology. Furthermore, the natural language generation component (NLG) can generate descriptions of the ontology’s instances or classes, as in our case. The ontology can also be

¹<http://www.ics.forth.gr/xenios/description.html>

²http://www.fhw.gr/index_en.html

³<http://www.forth.gr/>

⁴<http://cslu.cse.ogi.edu/toolkit/>

⁵<http://www.ltg.ed.ac.uk/dipper/>

⁶<http://www.ling.gu.se/projekt/trindi/trindikit/>

used during language interpretation, for example to locate properties the user's utterances may refer to.

An entire strand of work in Natural Language Generation (NLG) has focused on producing textual descriptions of classes and instances (entities) of domain specific ontologies (Reiter and Dale, 2000). A representative example is ILEX (O'Donnell et al., 2001), which was mainly demonstrated with museum ontologies. In more recent work, a multilingual extension of ILEX was developed in the context of the M-PIRO project (Isard et al., 2003; Androutsopoulos et al., 2007). The system was tested in several domains including museum exhibits and computing equipment. Similar systems have also been built to automatically produce medical reports (Bontcheva et al., 2004) and drug descriptions (Bontcheva and Wilks, 2004).

The rest of this paper is organised as follows: the architecture of the dialogue system is presented in section 2; in section 3, we present the resources that are used in the dialogue system; then, in section 4, we present the natural language generation module; in section 5, we provide samples of a visitor's interaction with the robot; the paper ends with an overview and conclusions in section 6.

2. Architecture of the Dialogue System

The dialogue system consists of *resources* and *modules* (see Fig. 1). The modules are the *Dialogue System Manager (DSM)*, the *Natural Language Generation Engine (NLG)*, the *Automatic Speech Recognition (ASR)* engine, the *Text To Speech (TTS)* synthesiser and the *Gesture Recogniser*. The latter can detect a set of three human gestures (yes, no, quit), and it is not described here. As TTS we have used the Demosthenes speech synthesiser (Xydas and Kouroupetroglou, 2001). As ASR we have used the BabEar system from Acapela⁷, which also provides natural language understanding facilities; the latter are based on a domain specific lexicon and a grammar.

The dialogue system manager (DSM) is the "actor" of the whole dialogue system, in the sense that it is the module that invokes and coordinates all the other modules. The DSM initiates the dialogue, and expects the visitor to select an option from the ones offered. In order to decide the next dialogue state and the text it will utter (through the TTS unit), it takes into account the *dialogue model*, the *interaction history* of the user, as well as the *location* of the robot. All the above contribute into creating more natural dialogues. Finally, there is the communication server, which enables the inter-module communication.

The resources are: the *dialogue model*; the *resources of the NLG*, which are discussed separately below; user models, including databases recording the interaction history of each user; and some databases that hold canned text and other information used during the dialogue. The modules are not domain specific; consequently, they can be easily transferred to another domain, with the exception of the ASR, which needs adjustments. On the other hand, the resources are domain specific and must be designed from scratch for each domain.

3. Resources of the Dialogue System

Dialogue Model

The dialogue model is based on a finite state automaton. At each state, the robot utters a particular question or statement. The transitions from a state represent the possible options offered to the human at that state. The dialogue model supports three sessions: *visitor welcome*, *tour* and *dialogue termination*. The dialogue model was developed especially for museums that host art collections, and are equipped with show rooms, where informative movies can be played.

In Fig. 2 we present an overview of the dialogue model. The first session (topmost node), captures the part of the dialogue session from the moment the visitor approaches the robot, till the moment he validates his ticket (validation occurs through a bar code reader, carried by the robot). The bottommost node represents the termination of the dialogue session. The second session (mapped to nodes 2, 3 and 4) concerns the presentation of the buildings and programmes of FHW to a visitor. The robot can talk about particular programmes, offer tour guidance, or do both.

The middle level nodes do not correspond to individual states; instead each one corresponds to approximately 10 states. Moreover, in each state there is a template which combines canned text with calls to the NLG in order to produce the final text to be uttered by the TTS.

Domain Ontology

The ontology represents the buildings and rooms of FHW, as well as the educational programs that are offered. The information about the buildings and the rooms is essential to the robot's navigation, to provide guidance instructions to visitors, and to relate educational programmes with their physical location. For instance, the following ontology extract includes the class *Area*, as well as the entities that belong to that class. For instance, *first-dome* and *ground-dome* represent the first and second floor of the dome (a semi-spherical movie projection hall).

```
Area
  ground-dome
  first-dome
  screening-room
  virtualReality-area
  basement1-building1
  basement2-building1
  first-building1
  ground-building1
```

The next ontology extract represents two types of programs that are offered, in particular there are documentaries about Miletus and the Black sea, as well as educational programs about the Ancient Agora and the ancients' contribution to mathematics.

```
Programme
  Documentary
    miletus
    black sea
  Educational
    ancientAgora
```

⁷<http://www.acapela-group.com/index.asp>

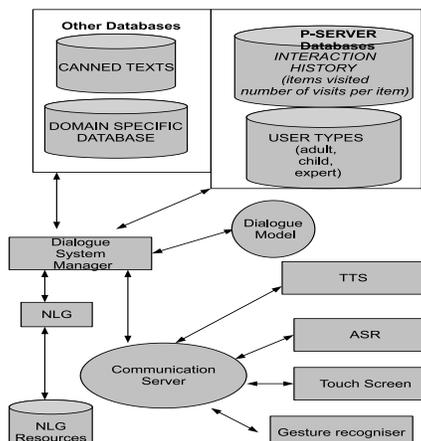


Figure 1: System Architecture

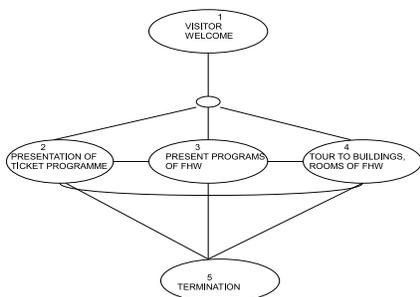


Figure 2: Dialogue Model

mathematics

The property *takes-place-at*, which is assigned to every programme, takes as its values entities of the *Area* class; for instance, $\langle \text{miletus, takes-place-at, screening-room} \rangle$. Summarising, the ontology describes all the buildings and rooms where educational activities take place, as well as a significant part of the programmes that are currently offered.

The ontology can be updated by adding visiting areas, new exhibits, or by updating information on already existing areas and exhibits. The use of the ontology enables the system to describe the newly added or updated objects without further configuration. The ontology is authored in OWL,⁸ the Semantic Web standard language for specifying ontologies, using the ELEON editor (Bilidas et al., 2007). Furthermore, ELEON allows annotating the ontology with the linguistic resources required by the NLG engine. User types (e.g., adult, child, expert) can also be defined with ELEON; we discuss user types below. The linguistic re-

sources of the OWL ontology are exported by ELEON in the RDF format the NLG module requires.

The ontology was authored by a group of people from FHW. The group involved archaeologists that provided the initial information in the form of a collection of documents; this collection had been used before to produce paper leaflets and other promotional material. The same group extracted the most important paragraphs from the collection. The information of the paragraphs was then entered in ELEON, with additional information obtained from an architectural plan. Furthermore, FHW edited the contents of the canned text database, by using a simple editor developed for this purpose.

User Types

User types such as adult, child, expert, are defined with the aid of the ELEON authoring tool. The user types are defined when creating the domain ontology, and do not change afterwards. User types are quite useful, since they permit extensive personalisation of the information that users receive (Androutsopoulos et al., 2007). Thus, user types determine the interest of the ontology facts (e.g., some facts about painting techniques may be too elementary for experts), as well as how many times a fact has to be repeated before the system can assume that a user of a certain type has assimilated it (e.g., how many times we need to repeat the duration of the Roman period). In addition, user types specify the appropriateness of linguistic elements (e.g., lexicon and microplans, see below), as well as parameters that control the maximum desired length of an exhibit description. Finally, different synthesiser voices can be chosen for different user types. Orthogonal to the user types is the multilingual support (English and Greek) for the lexical items and the microplans.

Interaction History and Application Databases

The interaction history records information showing the exhibits each user has seen, the information the system has conveyed, as well as the natural language expressions the system has used to convey the information. This allows the system to avoid repeating the same information and expressions. The interaction history is kept in a database that is controlled by the Personalisation Server, which can also produce statistical data. Moreover, it can act as a recommender system (based on past usage data) upon the arrival of a new user (Paliouras et al., 2006). There is also a *Canned Text Database*, which contains fixed text that will be spoken at the commencement, at the end, or at an intermediate stage of the visitor's interaction with the dialogue system. Canned texts also contain some string variables that are instantiated during the dialogue session. Finally, there is a *Domain Specific Database*, which in effect contains instances of the ontology, for example, particular buildings, programs and rooms. This information is extracted from the ontology that the NLG module uses (Galanis and Androutsopoulos, 2007).

4. The Natural Language Generation engine

The natural language generation engine of Xenios, called NaturalOWL, is heavily based on ideas from ILEX and M-

⁸<http://www.w3.org/TR/owl-features/>

PIRO.⁹ Unlike its predecessors, NaturalOWL is simpler (e.g., it is entirely template-based) and it provides native support for OWL ontologies. Currently, the system supports English and Greek.

NaturalOWL adopts the typical pipeline architecture of NLG systems (Reiter and Dale, 2000). It produces texts in three sequential stages: *document planning*, *micro-planning*, and *surface realisation*. In document planning, the system first selects the logical facts (OWL triples) which will be conveyed to the user and it specifies the document structure. In micro-planning, it constructs abstract forms of sentences, then it aggregates them into more complex periods, and finally it selects appropriate referring expressions. In surface realization, the abstract forms of sentences are transformed into real text, and appropriate syntactic and semantic annotations can be added, for example to help the TTS produce more natural prosody. The system is also able to compare the described entity to other entities of the same collection (e.g., “Unlike all the vessels that you saw, which were decorated with the black-figure technique, this amphora was decorated with the red-figure technique.”).

4.1. Document planning

Content Selection

When NaturalOWL is instructed to produce a description of a class or an entity, it attempts to locate all the logical facts of the OWL ontology (OWL triples of the form $\langle S, \text{Property}, O \rangle$, where S and O are the semantic subject and object of the fact, respectively) that are relevant to that entity or class. First, it selects all the facts that are directly relevant. For example, in the case of the entity whose identifier is `exhibit24`, it first selects, among others, the following facts, which associate `exhibit24` with class `aryballos` and the entities `archaeological-delos`, `iraion-delos` and `archaic-period`.

```
<exhibit24,rdf:type,aryballos>
<exhibit24,current-location,
  archaeological-delos>
<exhibit24,location-found,iraion-delos>
<exhibit24,creation-period,archaic-period>
```

Then, the system selects iteratively facts that are indirectly relevant to the described entity or class. In our example, the second iteration would add facts like the following:

```
<archaic-period,covers,
  archaic-period-duration>
<aryballos,rdfs:subclassOf,vessel>
```

The final set of selected facts results by removing the already assimilated facts and then the facts with the lowest interest. The interest of each fact and the number of times a fact has to be mentioned in order to be considered as assimilated is specified in the user types (Androustopoulos et al., 2007).

⁹See <http://www.ltg.ed.ac.uk/methodius/> for information on METHIDIUS, another descendant of M-PIRO’s generator.

Text Planning

The selected facts of the first iteration are ordered based on a domain-specific partial order of their properties (e.g., `current-location` must be mentioned after `creation-period`). More specifically, the partial order is created with the help of the ELEON tool, which allows the users to assign to each property an order score, an integer from 1 to 10. Smaller order scores indicate that facts with those properties should be mentioned earlier in the generated description. The order is stored in the RDF annotations of the ontology. The selected facts of the second or subsequent iterations are placed immediately after the corresponding facts of the first iteration.

4.2. Micro-planning

Abstract sentence forms

The RDF annotations of the ontology map each OWL property to one or more natural language templates. The templates consist of a list of slots and instructions showing how to fill them in. Each slot can be filled in with any of the following:

- A referring expression pointing to the semantic subject of the fact.
- A fixed string. If the string is a verb, it is specially tagged along with tense and voice. Prepositions are also tagged. These tags are used in aggregation.
- A referring expression pointing to the semantic object of the fact.

The final sentences are constructed by concatenating the slot values. For example, for the `current-location` property, the following microplan is defined:

```
Microplan(MicroName: "templ1",
  ForProperty="current-location") ->
Slot(Type:String, Value: "today")
Slot(Type:Owner, Case: nominative)
Slot(Type:Verb, Value: "is exhibited",
  Voice: Passive, Tense: present)
Slot(Type:Prep, Value: "in")
Slot(Type:Filler, Case: accusative)
```

If the above microplan is applied to the fact `<exhibit24, current-location, archeological-delos>`, the abstract sentence “*today* Ref(Entity: `exhibit48`, Case: `nominative`) *is exhibited in* Ref(Entity: `archaeological-delos`, Case: `accusative`)” is produced. The Ref tags represent the referring expressions which will be produced in a following stage.

Aggregation

The produced abstract sentences are then aggregated into longer sentences using a set of aggregation rules similar to M-PIRO’s (Melengoglou, 2002). For example, NaturalOWL’s rules aggregate phrases that share the same semantic subject (e.g., by adding the connective “and”), or pairs of phrases where the semantic object of the first one is the same as the semantic subject of the second one and the second phrase’s verb is “to be”, as illustrated below.

This aryballos was found in the Heraion of Delos. It is currently exhibited in the Archaeological Museum of Delos.

->

This aryballos was found in the Heraion of Delos and it is currently exhibited in the Archaeological Museum of Delos.

This is an aryballos. An aryballos is a kind of vessel.

->

This is an aryballos, a kind of vessel.

Referring expressions generation

In this stage, the system generates the appropriate referring expressions. When an entity or class is introduced for the first time, its name (e.g., "*Bancroft Chardonmay is a ...*") or the name of its class is used (e.g., "*this is an aryballos*"). In all subsequent references, an appropriate pronoun is used (e.g., "*it was found in the Heraion of Delos*"). If the focus (semantic subject of facts) shifts to another entity or class, again the name or the class of the new focus is first used, followed by pronoun references. When the focus returns to the original entity, a demonstrative (e.g., "*this aryballos is made of ...*") or the name of the original entity is used.

4.3. Surface Realization

Finally, in surface realization the system simply concatenates the slot values of the sentence specifications. In addition, syntactic or semantic markup can be added, as already mentioned.

5. Sample Interaction

Below we present an excerpt from a dialogue between the robot and a visitor. In the robot's utterances, the words that appear in italics are generated by the NLG module, whereas the other words are canned texts, with some intervening values of variables. As it can be seen, each robotic utterance is a mixture of dynamically generated and canned text, and as the dialogue progresses, the robot traverses the states of the dialogue model.

Robot Ok. Let us talk about the programmes of the Hellenic World. There are 6 programmes. The following: *a programme about the Ancient Agora of Athens, a documentary about the Olympic Games, ...* Please tell me which programme you are interested in.

Human I am interested in the Ancient Agora of Athens.

R Right. *This is an educational programme. Its title is: "In the Ancient Agora of Athens". It uses multimedia technologies and it is hosted in building 1. Its subject is history and it was established in 2007. It is addressed to children from 9 to 15 years old.* Would you like me to guide you to the place of the programme?

The following excerpt illustrates a case where the robot does not understand the user's utterance:

R Which building would you like to visit? I remind you that the Hellenic World has the following buildings: *building 1, building 5, building 7, Tholos, Research Centre.*

H The first one.

R I am sorry! I did not understand. I asked you which building you would like to visit. The answers I expected were: *building 1, building 5, building 7, Tholos and Research Centre.* Please select one or use the touch screen.

6. Conclusions and Future Work

We have described the dialogue system of the Xenios robotic guide. The dialogue system comprises many modules and resources. The central module is the dialogue manager that supports robot initiative dialogues. During human-robot interaction, the robot's utterances are a mixture of canned and dynamically generated text. Dynamic text generation proceeds in three stages: document planning, microplanning, and surface realisation. One of the most important resources of the system is the enriched domain ontology, which is authored by using the ELEON ontology editor. The ontology comprises descriptions of the FHW exhibits in OWL, as well as linguistic and user type information in RDF. The system supports both English and Greek.

Although an evaluation of the robotic guide has not been performed yet in real conditions, an initial version of the dialogue system has already been examined by FHW. In particular, the dialogue model, the richness of the generated text by the NLG, and the ontology are currently under revision based on user interaction data.

In the future we envisage the integration of an affective unit into the dialogue system. There is considerable research that suggests that user emotion recognition and robotic emotion expression may lead to more natural forms of communication (André et al., 2004).

Acknowledgements

This work was carried out in the context of project XENIOS, which was co-funded by the Greek General Secretariat and the European Union.

7. References

- E. André, L. Dybkjær, W. Minker, and P. Heisterkamp, editors. 2004. *Affective Dialogue Systems*. Springer.
- I. Androutopoulos, J. Oberlander, and V. Karkaletsis. 2007. Source authoring for multilingual generation of personalised object descriptions. *Natural Language Engineering*, 13(3):191–233.
- M. Bennewitz, F. Faber, M. Schreiber, and S. Behnke. 2005. Towards a humanoid museum guide robot that interacts with multiple persons. In *Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots*.
- D. Bilidas, M. Theologou, and V. Karkaletsis. 2007. Enriching OWL Ontologies with Linguistic and User-related Annotations: the ELEON system. In *Proceeding of International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE Computer Society Press.
- K. Bontcheva and Y. Wilks. 2004. Automatic report generation from ontologies: the MIAKT approach. In *Proceedings of the 9th International Conference on Applications*

- of *Natural Language to Information Systems*, pages 324–335, Manchester, UK.
- K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. 2004. Evolving GATE to meet new challenges in language engineering. *Natural Language Engineering*, 10(3/4):349–373.
- C. Chiu. 2004. *The Bryn Mawr Tour Guide Robot*. Ph.D. thesis, Bryn Mawr College.
- D. Galanis and I. Androutsopoulos. 2007. Generating multilingual descriptions from linguistically annotated OWL ontologies: the NATURALOWL system. In *Proceedings of the 11th European Workshop on Natural Language Generation*, Schloss Dagstuhl, Germany.
- A. Isard, J. Oberlander, I. Androutsopoulos, and C. Matheson. 2003. Speaking the users’ languages. *IEEE Intelligent Systems*, 18(1):40–45.
- M. F. McTear. 2004. *Spoken Dialogue Technology. Toward the Conversational User Interface*. Springer.
- A. Melengoglou. 2002. Multilingual aggregation in the M-PIRO system. Master’s thesis, School of Informatics, University of Edinburgh, UK.
- D. Milward and M. Beveridge. 2003. Ontology-based dialogue systems. In *3rd Workshop on Knowledge and Reasoning in Practical Dialog Systems*.
- M. O’Donnell, C. Mellish, J. Oberlander, and A. Knott. 2001. ILEX: an architecture for a dynamic hypertext generation system. *Natural Language Engineering*, 7(3):225–250.
- G. Paliouras, A. Mouzakidis, C. Ntoutsis, A. Alexopoulos, and C. Skourlas. 2006. PNS: Personalized Multi-Source News Delivery. In *Proceedings of the 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems (KES), UK*.
- E. Reiter and R. Dale. 2000. *Building natural language generation systems*. Cambridge University Press.
- S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. 1999. MINERVA: A second generation mobile tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.
- G. Xydas and G. Kouroupetroglou. 2001. The DEMOSTHENES Speech Composer. In *Proceedings of the 4th ISCA Tutorial and Workshop on Speech Synthesis*, Perthshire, Scotland.