

# Towards Ontology-Based Natural Language Processing

**Dominique Estival, Chris Nowak and Andrew Zschorn**

Human Systems Integration Group  
Defence Science and Technology Organisation  
PO Box 1500, Edinburgh SA 5111  
AUSTRALIA

{Dominique.Estival,Chris.Nowak,Andrew.Zschorn}@dsto.defence.gov.au

## Abstract

Conceptualising a domain has long been recognised as a prerequisite for understanding that domain and processing information about it. Ontologies are explicit specifications of conceptualisations which are now recognised as important components of information systems and information processing. In this paper, we describe a project in which ontologies are part of the reasoning process used for information management and for the presentation of information. Both accessing and presenting information are mediated via natural language and the ontologies are coupled with the lexicon used in the natural language component.

## 1 Introduction

*Ontologies* are now being recognised as important components of information systems and information processing. It is commonly accepted that an ontology is an *explicit specification of a conceptualisation* (Gruber, 1995). In the areas of knowledge representation and reasoning (KR) and of conceptual modelling, it has long been recognised that conceptualising a domain is a prerequisite for understanding the domain and processing information about the domain, especially in the case of large, non-trivial domains. Nowadays, there is no clear-cut border between large and small domains, simply because information systems are no longer isolated but are parts of the *global information system* and need to be *interoperable*. Hence, conceptualisations and ontologies are required for all kinds of information

systems and information processing. In some cases it is not clear yet what functions and advantages ontologies can offer, but there is no doubt that in *every* case ontologies do offer something: at the very least they offer a way to address *meaning of terms (concepts, relations)* required for information processing.

This paper attempts to provide some suggestions on how natural language processing can benefit from using ontologies. We present a large-scale research project in which ontologies are part of the reasoning process used for information management and for the presentation of information. Users' access to information and the presentation of information to users are both mediated via natural language, and the ontologies used in the reasoning component are coupled with the lexicon used in the natural language component.

In Section 2, we describe the FOCAL (Future Operations Centre Analysis Laboratory) project: both the ontological processing and the natural language processing work presented here are based on the relevant aspects of FOCAL. In Section 3, we present ontology-related work for FOCAL and in Section 4, the NLP-related aspects of FOCAL. In Section 5, we show how ontologies and NLP are combined. Section 6 summarises the current state of this work and indicates directions for future research.

## 2 Future Operations Centre Analysis Laboratory (FOCAL)

The Future Operations Centre Analysis Laboratory (FOCAL) is a research project whose goal is to "pioneer a paradigm shift in command environments through a superior use of capability and greater situation awareness" (FOCAL Task Plan). In part, this involves building a high-level

information fusion system for the military domain (Lambert, 2003; FOCAL, 2002).

To support this goal, the FOCAL facility was designed to experiment with innovative technologies. FOCAL contains a large-screen (150°) semi-immersive virtual reality environment as its primary display, allowing vast quantities of information (real or virtual) to be displayed. Spoken dialogue with virtual characters known as VAs (Virtual Advisers) is one of the means of delivering information (Estival et al., 2003).

Within the FOCAL project, the Natural Language Processing (NLP) and the Knowledge Representation and Reasoning (KR) work packages are tasked with providing appropriate NLP and KR functionalities, including processing natural language queries and providing a formalisation of the domain and reasoning capabilities. These two work packages are closely related in that a natural language query is to be processed, mapped to its formal representation and answered by a reasoning subsystem, and then a natural language answer is returned to the user.

Current FOCAL work is focused on implementing a scenario, which is located within a particular military situation and describes a military domain, a limited (in space and time) region of the world, and other relevant elements of that situation. Among other things, the domain description requires dealing with geography, logistics and planning.

The FOCAL architecture is agent-based and uses the CoABS (Control of Agent Based Systems) Grid as its infrastructure (Global InfoTek, 2002). The CoABS Grid was designed to allow a large number of heterogeneous procedural, object-oriented and agent-based systems to communicate. FOCAL agents process information, communicate and collaborate. Most agents are implemented in ATTITUDE and communication between agents is accomplished via string messages (Wark et al., 2004).

Humans are also involved in FOCAL, as the end users who interact with the system to perform their work and achieve their goal: successfully planning and conducting an operation. The current scenario provides a testbed for the system. Extensions of the scenario and new scenarios for different domains will ensure that FOCAL functions as expected outside of the limited domain of the current scenario.

There are many aspects of FOCAL which are not directly related to NLP and KR activities, and which are therefore excluded from this discussion. In the rest of this paper, only aspects relevant to NLP and KR are considered.

### 3 Ontological Reasoning for FOCAL

The main task of the KR work package within the FOCAL project is to provide the FOCAL users with automated knowledge management and with automated reasoning capabilities about a complex domain. Ontologies have been chosen as the type of representation most suited for this task, and the provision of *ontological reasoning* capabilities has been one of the main thrusts. An *ontology* for FOCAL has been built and a number of reasoning activities are now ontology-based.

#### 3.1 Conceptualisation

Lambert (2001) advocated Dennett's Intentional Stance framework (Dennett, 1987). Dennett identified three means by which people predict and explain outcomes.

1. The first is the Physical Stance, where one engages principles of Physics to predict outcomes. People employ this when playing snooker or assessing the trajectories of projectile weapons.
2. The second is the Design Stance, where one engages principles of design to predict and explain outcomes. People employ this when troubleshooting an automobile fault or coding and maintaining computer programs.
3. The third is the Intentional Stance, where one engages principles of rationality to predict outcomes. People employ this when forecasting the actions of a fighter pilot or when competing with an advanced computer game.

The Design Stance is used whenever the physics of the situation is too difficult or laborious. The Intentional Stance is used whenever the design underpinning the situation is too difficult or laborious.

Lambert (2001, 2003) adopts Dennett's framework for representing knowledge about the world, but adds two other layers: a *metaphysical layer* below the physical layer, and a *social layer* above the intentional layer. Therefore, *formal*

*theories* that allow one to represent and reason about the world, would be assigned to the following levels:

1. Metaphysical theories, for *what* there is, *where* and *when*.
2. Physical theories, for the operation of aspects of the *environment*.
3. Functional theories, for the operation of *designed artefacts*.
4. Intentional theories, for the operation of *individual minds*.
5. Social theories, for the operation of *groups of individuals*.

This five level framework proposed by Lambert suggests a way to *conceptualise* the domain in terms of *processes*, namely *metaphysical*, *physical*, *functional*, *intentional* and *social* processes (M, P, F, I, S processes). The resulting conceptualisation is referred to as a *Mephisto* conceptualisation (Nowak, 2003) and is the basis for the ontologies we are constructing for FOCAL.

### 3.2 Ontological languages

Ontologies are concerned with the *meaning* of terms. It is therefore appropriate when selecting an *ontological language* to choose a language which is equipped with a formal semantics. This requirement excludes XML from the list of possible candidates, as XML does not offer semantics, but only syntax. RDF provides some semantics, but proper, formal semantics requires languages based on logics. Description logics (DL) provide some frameworks, and several languages used for building and processing ontologies are DL-based, e.g. DAML and DAML+OIL languages, including such languages as SHF and SHIQ, and the OWL language (Horrocks et al., 2003).

A commonly used view of an architecture for the Semantic Web is a layered architecture, with XML as the bottom layer, RDF as the middle layer, and logic (e.g. DL) as the top layer (sometimes the top layer distinguishes ontological vocabulary, logic, proof; on top of the logic layer a trust layer is sometimes placed). The logic layer is a necessary component if the Semantic Web is to be equipped with a formal semantics; this logic layer can be based on a description logic (such as SHIQ or OWL), on first-order logics, KIF or

CyclL, and whichever logic is used determines the expressibility and tractability of the framework, but in every case a formal semantics is added. Frameworks based on DL (description logics) are most successful, because they provide expressive languages with practical tractability. SHIQ is one such language, another is the closely related language OWL

The ontological language chosen for FOCAL is SHIQ, a DL language of the DAML+OIL project (<http://www.daml.org/>), a successor of the OIL project (<http://www.ontoknowledge.org/oil/>). FaCT (<http://www.cs.man.ac.uk/~horrocks/FaCT/>) is a reasoner for the SHIQ logic employed in the OilEd ontology editor (<http://oiled.man.ac.uk/>). The logic SHIQ has also been implemented in the ([www.cs.concordia.ca/~faculty/haarslev/racer/](http://www.cs.concordia.ca/~faculty/haarslev/racer/)) RACER project.

SHIQ is closely related to OWL (Horrocks et al., 2003). In fact, there are a few variants of OWL, namely OWL Lite, OWL DL and OWL Full. OWL Lite is similar to a description logic SHIF(D), while OWL DL is similar to a description logic SHOIN(D). The language implemented in the RACER framework is a version of SHIQ, which provides some functionalities for dealing with individuals, and dealing with concrete domains; this makes the RACER's version of SHIQ very close to OWL Lite. A proper discussion on these languages is beyond the scope of the paper, but clearly the RACER language is an implemented language and reasoner for a logic very close to OWL DL. References related to OWL, SHIQ and OIL include (Horrocks et al., 2003), (Bechhofer and Horrocks, 2003) and (Horrocks, Sattler and Tobies, 2000).

### 3.3 Ontological frameworks

Ontology frameworks provide formalisms for building ontologies, but do not provide the contents. Therefore, they should do at least two things:

- provide a *formal language* in which the ontologies can be expressed or specified, and
- provide some reasoning capabilities, so that an ontology can be demonstrated to be consistent (i.e. free of contradictions, assuming that contradictions indicate modelling mistakes or errors).

Given this standpoint, frameworks that do not provide reasoning capabilities are unsatisfactory. Note also that a *formal language* is usually a *logical language*, with clearly specified syntax and semantics, and the logic should be sound, complete, decidable, and hopefully tractable (or tractable in practice). These properties of the logical framework are necessary to obtain reasoning facilities. The most attractive ontology frameworks seem to be the following (see Table 1 for a more detailed comparison of the different frameworks):

1. the OIL framework based on *description logics*,
2. the OntoEdit/OntoBroker framework (F-logic),
3. the Ontolingua framework based on the KIF logic.

| framework | logic          | reasoner   | editor    |
|-----------|----------------|------------|-----------|
| Ontoprise | F-logic        | OntoBroker | OntoEdit  |
|           | $AGCQHLR+(D)-$ | RACER      | RICE      |
| OIL       | $SHF$          | FaCT       | OilEd     |
|           | $SHTQ$         |            |           |
| DAML+OIL  | $SHOQ(D_n)$    |            | Protégé   |
| OWL       | $SHTF$         | Cerebra    | Construct |

Table 1: Ontology frameworks, logics, reasoners and editors.

For FOCAL, we have chosen to employ the OIL and RACER frameworks. Ontologies are built using the OilEd ontology editor and verified using FaCT. At run-time, a RACER agent is initialised with the ontology (see section 3.4).

### Higher order relations and Description Logic

Although description logics on which OIL and RACER are based allow only binary relations, we use OIL and Racer in a way that also allows us to employ arbitrary n-ary relations and higher-order relations. Given that a ternary relation can be represented as a binary relation that takes another binary relation as one of its argument, any n-ary relations can be represented via higher-order relations, i.e. relations which take other relations as arguments. Suppose that we want to implement a second-order relation that takes as its first argument a binary relation- more precisely, the second order relation takes as its first argument instances of that binary relation- rather than instances of a concept. The instances of the binary

relation can be mapped to instances of a newly created concept, i.e. the concept of individuals which are single entities but correspond to (and are linked to) the instances of the binary relation. There is an exact correspondence between the second-order relation taking a binary relation instance as its first argument and its implementation in terms of a binary relation that takes as its first argument an instance of the concept which has instances of the other binary relation as its individuals. The approach we described here has now been used to implement in the FOCAL ontology information which extends beyond the binary relation based language. Multiple facts involving n-ary relation and higher-order relation are present in the current version of the FOCAL ontology. ATTITUDE agents are currently being built to allow automated reasoning with this extended language.

### Implemented Ontology

As mentioned in section 1 the FOCAL scenario, which is based on real material for training exercise, provides background information in a number of domains, including geography, political situation, logistics, weather. For now, the scenario also specifies what kinds of questions can be asked by FOCAL users, to be answered by the FOCAL agents. The ontology serves as a formal, clearly specified knowledge base containing the background information and allowing the agents to query that knowledge base and to get replies helping them to answer the queries.

An initial version of the FOCAL ontology has been created manually using OilEd and verified using FaCT.<sup>1</sup> There are in fact several ontologies, for the different domains covered in the scenario, and an important research issue is that of the combining (or merging) of the ontologies in the larger FOCAL one. Another issue is that the manual creation of the ontologies is a time consuming and tedious process, but the existence of tools such as FaCT ensures that the result is consistent and free of mistakes due to user input errors.

### 3.4 Ontological reasoning

<sup>1</sup> The FOCAL ontology currently contains over 300 concepts, about 80 relations and over 100 individuals (plus a large number of facts connecting all of these).

Both the FaCT and RACER reasoning agents provide reasoning facilities, FaCT during the building of the ontologies to ensure coherence and consistency, and RACER at run-time. When integrated within the FOCAL system, the RACER server can be initialised with a given ontology and there is a RACER client wrapped as a CoABS agent on the grid, which can connect to the server. Other FOCAL agents, e.g. the Dialogue Manager (see section 4.1), can then communicate with the RACER server (via the RACER client agent) and receive answers using the ontology.

The ontology can be also be accessed and queried outside of the FOCAL system, still using a client-server connection.

- Using OilEd, the ontology "focal.daml" can be saved in the DIG format as a file named "focal.dig".<sup>2</sup>
- The RACER server can be started and initialised to the "focal.dig" ontology.
- A java package called jracer includes a test client (<http://www.lsi.upc.es/~jalvarez/>) which can be used to connect to the RACER server.

At the ">" prompt, queries can be entered. The queries are received and replied to by the server. For instance, we show in (1) an example of a query as to whether (the individual) **AUSTRALIA** is an instance of (the concept) **nation**, and give the server's answer to that query, i.e. **T** (for *true*).

(1) > (individual-instance? AUSTRALIA nation)  
T

### 3.5 Hierarchies of concepts and relations

A DL-based ontology, such as our OilEd "Focal" ontology, is a knowledge base (KB) expressed in a DL language. Every DL language provides facilities for defining concepts, with the relation of *subsumption* between the concepts being the core relation and the basis for building the definitions. The set of concepts can be seen as an ordered set, the subsumption relation being the ordering relation; hence, we have a hierarchy of concepts. There is also a hierarchy of relations ordered by the subsumption relation. These two hierarchies, together with the concepts' definitions, can be taken to form a *lexicon*, i.e. a list of *words* (for

<sup>2</sup> OilEd can export to SHIQ, OWL and other formats.

*concepts* and *relations*) with well-defined meanings for those words.

These two hierarchies of concepts and relations thus provide a basis for a domain specific lexicon and one of the advantages which ontologies can offer NLP systems is that a properly built knowledge base (as an ontology) will allow the semi-automatic creation of a lexicon.

## 4 NLP in FOCAL

The underlying architecture for dialogue management has been developed using ATTITUDE agents (Estival et al., 2003). Input from FOCAL users can be either spoken or typed and is processed by the same NLP component. We use Nuance for speaker-independent speech recognition (Nuance, 2002) and the open source Regulus NLP package (Rayner et al., 2001) for grammar development.<sup>3</sup> We are in the process of integrating language input with input from other devices, e.g. pointing devices such as mouse or wand, gesture tracking device and, in the future, gaze tracking.

### 4.1 Dialogue Agents

The FOCAL Dialogue Agents can be divided into 3 categories: Input Agents, Internal Reasoning Agents and Output Agents. The Input Agents comprise:

- Speech Input

The Speech Input agent is a straightforward wrapper around a Nuance Client implementation. It forwards on to the Input Fuser the interpretations of speech recognition results (in the form of lists of Attribute-Value pairs), notifications of failed recognition events and the interpretations of typed input. It also passes on instructions to activate and de-activate the recogniser.

- Input Fuser

The Input Fuser (IF) is responsible for receiving and combining user input. This input can be via speech (Nuance), keyboard (typed input), gesture, gaze etc. The IF turns streams of input events into a Bayesian network of discrete communicative acts

<sup>3</sup> The existing grammar was developed using Regulus 1, but we are currently developing a larger, more flexible grammar with Regulus 2 (Rayner et al., 2002) which will provide a broader coverage, allowing the more naïve users to be recognised more easily.

which are then interpreted by the Dialogue Manager.

The Internal Reasoning Agents comprise:

- Reference Resolver

This is currently a stub, but the Reference Resolver is meant to assist other agents (particularly the Input Fuser and the Dialogue Manager) resolve anaphoric references found in user communicative acts by maintaining context and linking dialogue variables to referents.

- Dialogue Manager

The Dialogue Manager (DM) is activated by a message that includes an activation context symbol. The DM receives the Bayesian network of interpretations of user(s) communicative acts from the IF and it finds the interpretation with the highest probability that unifies with the current dialogue context. The DM then informs the IF of which interpretation of the communicative act was chosen, so the IF can forward the full information on to the Transcriber. At the same time, the DM requests that the Presentation Planner present the response to this communicative act; this request is termed a *communicative goal*.

- Presentation Planner

The Presentation Planner (PP) receives requests from the DM to achieve communicative goals. For now a communicative goal will succeed if there is a presentation clip which is marked-up with the conjunction of the DM's activation context and the meaning representation for the query, but current work is extending the PP agent along the lines given in (Colineau and Paris, 2003).

The Output Agents comprise:

- Transcriber

The Transcriber agent receives notification of user's communicative acts from IF and of the system's communicative acts from DM. It produces an HTML listing of these communicative acts, which includes speech recognition results and a link pointing to the audio recording.

- Text-to-Speech

If the output is to be presented verbally by the Virtual Advisers, it is sent to the Text-to-Speech (TTS) component. We use the *rVoice* TTS system, which gives us a choice of voices for the different VAs (rVoice, 2002).

## 4.2 Lexicon for NLP

As described above, language processing is performed by the Nuance/Regulus grammar. Regulus is an Open Source environment which compiles typed unification grammars into context-free grammar language models compatible with the Nuance Toolkit.<sup>4</sup>

The lexicon for Regulus 2 is of the form shown in (2) and (3), where the macro in (2) defines the properties of a noun class, and the instances in (3) specify the lexical items belonging to that class, in this case *result*, *results*, *outcome*, *outcomes*.

(2) macro defining noun class

```
macro(noun_like_result(Words,Sem),
      @noun(Words, [sem= @noun_sem(abstract, Sem),
                    sem_n_type=abstract, takes_det_type=def/null,
                    n_of_mod_type=_])).
```

(3) examples of nouns for that class:

```
@noun_like_result([result, results], result).
@noun_like_result([outcome, outcomes], result).
```

## 4.3 Meaning representation

The Meaning Representation produced by the NLP component, and passed on by the Speech Input agent, is translated into an ATTITUDE expression.

For example, if a user can ask the question given in (4.a), it will first be translated into the (simplified) list of attribute value pairs given in (4.b) and sent to the Speech Input agent. Speech Input then translates these attribute value pairs into the (simplified) ATTITUDE expression given in (4.c) and forwards it on to the Input Fuser agent.

(4) a. What is our relationship with PNG?

b. (question whquestion concept relationship obj1 Australia obj2 Papua\_New\_Guinea)

c. (comm\_act (?relationship Australia Papua\_New\_Guinea) from speaker type whquestion )

## 5 Natural Language & Ontological Processing for FOCAL

There are at least two ways that ontologies can facilitate language processing. Firstly, an ontology can be used directly when building the lexicon, defining the *terms* (*concepts* and *relations*) for content words. Secondly, an ontology is a knowledge base (KB), expressed in a formal language, and therefore it provides (formally

---

<sup>4</sup> Regulus is described in detail in (Rayner et al., 2001).

expressed) *knowledge* for more complex language processing.

## 5.1 Ontology and the lexicon

We view an ontology as a knowledge base, consisting of a structured list of concepts, relations and individuals. The ontology provides partial definitions for these, through the taxonomy relation between the terms and the properties specified for them. An example of how a fragment of a lexicon, for the content words in the domain, can be obtained from an ontology is presented below.

We give in (6) an ontology fragment, where every concept is listed in the format shown in (5).

- (5) ( concept\_n  
       list-of-parents\_of\_concept\_n  
       list-of-children\_of\_concept\_n )
- (6) ( ( |ship|  
       (|platform|  
       (|frigate|) )  
       (|platform|  
       (|asset|  
       (|aircraft| |ship|) )  
       (|frigate|  
       (|ship|  
       (|ffg|) ) )

For completeness, we give in Figure 1 the actual OWL format for this fragment.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<owls:Ontology xmlns:owls=http://www.w3.org/2002/OWL-XMLSchema
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2002/OWL-XMLSchema
http://potato.cs.man.ac.uk/owl/schemas/owl1-dl.xsd">
<owls:Class owls:complete="false"
owls:name="file:/D:/ontology/focal.daml#platform">
<owls:Class
owls:name="file:/D:/ontology/focal.daml#asset">
  </owls:Class> <owls:Class owls:complete="false"
owls:name="file:/D:/ontology/focal.daml#ffg">
<owls:Class
owls:name="file:/D:/ontology/focal.daml#frigate"/>
  </owls:Class>
<owls:Class owls:complete="false"
owls:name="file:/D:/ontology/focal.daml#ship">
<owls:Class
owls:name="file:/D:/ontology/focal.daml#platform">
  </owls:Class>
```

```
<owls:Class owls:complete="false"
owls:name="file:/D:/ontology/focal.daml#frigate">
</owls:Class>
<owls:Class owls:complete="false"
owls:name="file:/D:/ontology/focal.daml#aircraft">
<owls:Class
owls:name="file:/D:/ontology/focal.daml#platform"/>
  </owls:Class>
<owls:Class owls:complete="false"
owls:name="file:/D:/ontology/focal.daml#asset"/>
</owls:Class>
</owls:Ontology>
```

Fig. 1: FOCAL ontology fragment in OWL format

Simplified lexical entries for the words *aircraft*, *airplane*, *airplanes*, *plane*, *planes*, *ship*, *ships*, *frigate*, *frigates* and *FFG* are shown in (7) and (8).

(7) macro for noun class "platform":  
 macro(noun\_like\_platform(Words,Sem),  
       @noun(Words, [sem= @noun\_sem(platform, Sem),  
       sem\_n\_type=platform, takes\_det\_type=def/null,  
       n\_of\_mod\_type=\_])).

(8) examples of nouns for class "platform":  
 @noun\_like\_platform([frigate, frigates], ship).  
 @noun\_like\_platform([ffg], ship).  
 @noun\_like\_platform([ship, ships], ship).  
 @noun\_like\_platform([airplane,airplanes,plane,planes],  
 aircraft).  
 @noun\_like\_platform([aircraft], aircraft).

This example shows how synonyms are handled in our system, with the same semantic interpretation, and the same parent class, given to a number of lexical items.

## 5.2 Ontology as knowledge

Since an ontology is a knowledge base expressed in a formal language, it provides formally expressed *knowledge* for language processing. Although at this point not all this knowledge can be used directly by the speech recognition system which processes the speech input, nor by the grammar which builds the meaning representations, some of this knowledge can already be used by the other Dialogue agents, in particular the Dialogue Manager, and later by the Reference Resolver.

The best example is the resolution of ambiguity, such as the polysemy of some terms. For instance the name *Adelaide* can refer to a city (Adelaide in South Australia), a ship ("HMAS Adelaide"), a



river (the Adelaide River in the Northern Territory of Australia), or even a person, (e.g. "Queen Adelaide"). While, as shown in Section 5.1, synonymy is handled by the lexicon, polysemy is resolved by drawing on a variety of sources, including the ontology.

When the Dialogue Manager receives from the Input Fuser a set of communicative acts, if one of these communicative acts correspond to distinct plausible interpretation results, e.g. "Adelaide:{city, ship}", it can try to resolve the ambiguity by using the context information and by sending a request to the KR agent.

## 6 Conclusion

This paper has described our current work within the FOCAL project to combine ontologies built with the OIL/RACER framework with our spoken dialogue system. It provides some suggestions on how ontologies can help a natural language processing component build semantic representations which are directly used in a complex information management system.

This is work in progress and a formal evaluation has not yet been put in place. However, the reviewers for this paper have rightly asked how this would be conducted. In the agent-based architecture we use, each agent can be tested in isolation and we have already conducted tests to ensure that the answers returned by the KR agent for specific questions in our scenario are correct and consistent. A more interesting evaluation will be possible when the scenario is expanded, to see whether unplanned answers returned when the system is asked new unscripted questions are in fact useful to the users. This will take place in the next phase of the project.

For now, we conclude that an ontology is a knowledge base which can serve as the basis for creating the part of the lexicon for domain content words. This is achieved by producing a list of terms with their meanings, i.e. partial definitions given the two hierarchies in the ontology, and we are exploring methods to automate this process.

## References

- S. Bechhofer and Ian Horrocks. 2003. *The Wonder Web Ontology Language*. Report and Tutorial.
- S. Blackburn. 1996. *The Oxford Dictionary of Philosophy*. Oxford University Press.
- Nathalie Colineau and Cécile Paris. 2003. Framework for the Design of Intelligent Multimedia Presentation Systems: *An architecture proposal for FOCAL*. CMIS Technical Report 03/92, CSIRO, May 2003.
- Daniel C. Dennet. 1987. *The Intentional Stance*. Cambridge: MIT Press.
- Dominique Estival, Michael Broughton, Andrew Zschorn, Elizabeth Pronger. 2003. "Spoken Dialogue for Virtual Advisers in a semi-immersive Command and Control environment". In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan. pp.125-134.
- FOCAL. 2002. *DSTO and Virtual Reality*. <http://www.dsto.defence.gov.au/isl/focal.pdf>.
- Global InfoTek Inc. 2002. *Control of Agent Based Systems*. <http://coabs.globalinfotek.com>.
- T. R. Gruber. 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". *Human and Computer Studies*, vol. 43, no. 5-6.
- Ian Horrocks, Peter F. Patel-Schneider and Frank van Harmelen. 2003. "From SHIQ and RDF to OWL: The Making of a Web Ontology Language". *Journal of Web Semantics*, vol.1, no.1, pp.7-26.
- Ian Horrocks, U. Sattler and S. Tobies. 2000. "Practical reasoning for very expressive description logics". *Logic Journal of the IGPL*, 8(3):239-263.
- Dale A. Lambert. 2001. "An Exegesis of Data Fusion". In *Soft Computing in Measurement and Information Acquisition*, eds. L. Reznik and V. Kreinovich. Physica-Verlag.
- Dale A. Lambert. 2003. "Grand Challenges of Information Fusion". In *Proceedings of the Sixth International Conference on Information Fusion*. Cairns, Australia.
- Chris Nowak. 2003. "On ontologies for high-level information fusion". In *Proceedings of the Sixth International Conference on Information Fusion*. Cairns, Australia.
- Nuance. 2002. <http://www.nuance.com/>.
- Manny Rayner, John Dowding, Beth Ann Hockey. 2001. "A Baseline method for compiling typed unification grammars into context free language models". In *Proceedings of Eurospeech 2001*, pp 729-732. Aalborg, Denmark.
- Manny Rayner, Beth Ann Hockey, John Dowding. 2002. "Grammar Specialisation meets Language Modelling". *ICSLP 2002*. Denver, USA.
- rVoice. 2002. Rhetorical Systems, <http://www.rhetoricalsystems.com/rvoice.html>.
- Steven Wark, Andrew Zschorn, Michael Broughton, Dale Lambert. 2004. "FOCAL: A Collaborative Multimodal Multimedia Display Environment". In *Proceedings of SimTecT*. Canberra, Australia.