

# More on Less: Further Applications of Ontologies in Multi-Modal Dialogue Systems

Robert Porzel<sup>†</sup>, Norbert Pflieger<sup>‡</sup> Stefan Merten\*, Markus Löckelt<sup>‡</sup>  
Iryna Gurevych<sup>†</sup>, Ralf Engel<sup>‡</sup> Jan Alexandersson<sup>‡</sup>

<sup>†</sup>European Media Laboratory

Schloss-Wolfsbrunnenweg 31c

69118 Heidelberg, Germany

{porzel,gurevych@eml.org}

<sup>‡</sup>DFKI GmbH

Stuhlsatzenhausweg 3

66123 Saarbrücken, Germany

{pflieger,loeckelt,engel,janal}@dfki.de

\*DFKI GmbH

Erwin-Schrödinger-Str.57

67608 Kaiserslautern, Germany

merten@dfki.de

## Abstract

We present applications enabled via the employment of a single knowledge representation in the SMARTKOM multi-modal multi-domain dialogue system. We focus on how an rigorously constructed ontology whose ontological and representational choices are shared by multiple components of the system, can be reused in different projects and applied to various tasks.

## 1 Introduction

Historically, the ways in which knowledge has been represented in dialogue systems show that individual representations with different semantics and heterogeneously structured content can be found in various formats within single systems and applications. The diversity and heterogeneity of knowledge representation in earlier systems originates in the fact that each knowledge store is hand-crafted individually for each task. Additionally, we find a multitude of formats and inference engines, which often cause both performance and tractability problems.

We present how ontologically modelled knowledge is employed in the SMARTKOM system, based on the work introduced in Gurevych et al. (2003c). In this paper, we present additional benefits of employing a single knowledge store through a multi-modal dialogue system (MMDS) and extensions of the earlier work. Therefore, we will outline the underlying modeling principles and show the benefits of such a rigorously crafted knowledge store for the actual and future MMDS.

## 2 The Representational Formalism Used

The formalisms pertinent to the following description of the ontology originate in various W3C and Semantic Web projects. These brought about knowledge modeling standards, such as the Resource Description Framework

(RDF), the DARPA Agent Mark-up Language (DAML), the Ontology Interchange Language (OIL) and the Web Ontology Language (OWL).<sup>1</sup> This allows to represent domain and discourse knowledge in ontologies using XML-based semantic mark-up languages, such as OIL, or DAML+OIL. In the work reported here, we used an ontology defined in the OIL-RDFS syntax. OIL-RDFS is a representation format which allows to express any OIL ontology in RDF syntax. This has the advantage that the ontology is partially understandable for non-OIL aware RDFS applications. Additionally it allows for all the formal semantics and reasoning support available for OIL.

Fensel et al. (2001) provide a detailed characterization of the formal properties of the OIL language. The FACT<sup>2</sup> system can be used as a reasoning engine for OIL ontologies, providing some automated reasoning capabilities, such as class consistency or subsumption checking. Graphical ontology engineering front-ends and visualization tools are available for editing, maintaining, and visualizing the ontology.<sup>3</sup>

The OIL semantics is based on a combination of frame- and description logic extended with concrete datatypes. It provides most of the modeling primitives commonly used in the frame-based knowledge representation systems, e.g. frames are used to represent concepts. These frames consist of a collection of classes along with a list of slots and attributes. Under the term *class* or *class expression* a class name, or an enumeration, or a property-restriction, or a boolean combination of class expressions is to be understood. Slots are interpreted as a collection of properties. They are divided into those that relate classes to other classes (so called *object properties*) and those that relate classes to datatype values (so called *datatype*

<sup>1</sup>See [www.w3.org/RDF](http://www.w3.org/RDF), [www.ontoknowledge.org/oil](http://www.ontoknowledge.org/oil), [www.daml.org](http://www.daml.org) and <http://www.w3.org/2001/sw/WebOnt/> for the individual specifications.

<sup>2</sup>See also [www.cs.man.ac.uk/~horroks/FaCT/](http://www.cs.man.ac.uk/~horroks/FaCT/).

<sup>3</sup>See OilEd ([oiled.man.ac.uk](http://oiled.man.ac.uk)) for editing and FrodoRDFSviz ([www.dfki.uni-kl.de/frodo/RDFSviz](http://www.dfki.uni-kl.de/frodo/RDFSviz)) for visualization.

*properties*). Slots can be filled by: class names, names of the atomic elements, collection of the above (conjunctive sets - *and*, disjunctive sets - *or*, or negation - *not*), concrete datatypes (*integers* and *strings*).

Additionally, domain and range restrictions can be defined for slots. Domain restriction asserts that the property only applies to the instances of particular class expressions. Range restriction specifies that a property only assumes values that are instances of the respective class expressions. Slot fillers can have several types of further constraints, also called *facets*. These include *value-type* restrictions (all fillers must be of a particular class), *has-value* restrictions (there must be at least one filler of a particular class). The *value-type* restriction corresponds to the universal quantifier of the predicate logic. The *has-value* restriction is analogous to the existential quantifier. Another constraint on the slot fillers is *cardinality*, which limits the number of possible fillers of the given class. Atomic elements or individuals can also be associated with a class definition via slot constraints.

Schemes based on this combination of frame- and description logic allow to represent enough knowledge for the effective operation of NLP applications described in Section 4. We used the OIL language in particular as a whole range of software is freely available to support ontology construction as mentioned above. Additionally, it simplifies the usage of the ontology in Semantic Web applications.

### 3 The SMARTKOM Ontology

As one of the most advanced current systems, the SMARTKOM system (Wahlster et al., 2001) comprises a large set of input and output modalities together with an efficient fusion and fission pipeline. SMARTKOM features speech input with prosodic analysis, gesture input via infrared camera, recognition of facial expressions and their emotional states. On the output side, the system features a gesturing and speaking life-like character together with displayed generated text and multimedia graphical output. It currently comprises nearly 50 modules running on a parallel virtual machine-based integration software called *Multiplatform* described in Herzog et al. (2003).<sup>4</sup>

Complex MMDS such as SMARTKOM require a homogeneous world model, that serves as a common knowledge representation for various modules throughout the system. It represents and brings together a general conceptualization of the world (top-level or generic ontology) as well as of particular domains (domain-specific ontologies). This way, the ontology represents language-independent knowledge. Language-specific knowledge is stored elsewhere, e.g. in the lexicon containing lexical

<sup>4</sup>The abbreviation stands for "Multiple Language / Target Integration PLATform FOR Modules".

items together with their meanings defined in terms of ontology concepts.

Designed as a general purpose component for knowledge-based NLP, the ontology described herein, includes a top-level - developed following the procedure outlined by Russell and Norvig (1995). It originally covered the tourism domain, encoding knowledge about sights, historical persons and buildings. This existing ontology was adopted in the SMARTKOM project and modified to cover a number of new domains, e. g., new media and program guides, personal assistance system and standard applications. The top-level ontology was re-used with some slight extensions. Further developments were motivated by the need of a *process hierarchy*. This hierarchy models processes which are domain-independent in the sense that they can be relevant for many domains.

Currently, the ontology employed by the system has about 730 concepts and 200 relations. The purpose of the top-level ontology is to provide a basic structure of the world, i. e., abstract classes to divide the universe in distinct parts as resulting from the ontological analysis (Guarino and Poli, 1995). The domain concepts emerged through a comprehensive corpus analysis. The acquisition of domain concepts was done on the basis of dialogues from the SMARTKOM domains. There is no strict methodology for the concept acquisition. Ontology construction on this level is rather a matter of constant negotiation, which distinctions to make. The most important modeling decisions are described in Gurevych et al. (2003c). Once available, the ontology was augmented with comments containing definitions, assumptions and examples that facilitate its appropriate use in a multi-component system such as SMARTKOM and its possible re-use in other systems. Such descriptions of ontology classes are particularly important as the meanings associated with them may vary considerably from one ontology to another.

## 4 Applications in SMARTKOM

### 4.1 Natural language understanding

A template based semantic parser (Engel, 2002) is used for the task of natural language understanding. Similar to production systems templates modify a working memory (WM) which is initially filled with the input words delivered by the speech recognizer. Then the templates transform the initial words step-by-step first to simple instances of the ontology and afterwards combine these instances.

Each template consists of a condition and an action part. The condition part checks the presence of certain words or instances of certain classes in the WM. The action part creates one or more new instances that may contain slots filled with instances matched by a condition.

#### 4.1.1 Challenge

We address two issues related to the ontology.

Since the templates are manually created it is helpful if the templates can be written as general as possible. Writing templates is also an error-prone task. To avoid the production of output which is inconsistent with respect to the ontology, automatic syntactic and semantic checking should be available.

Referential expressions uttered by the user often bear no or little information about the type of the discourse object (e.g., “this one” or “the map” instead of “the street map”). Nevertheless, such utterances have to be analyzed correctly. In many cases the linguistic context within the utterance restricts the type of the discourse object referred to. If this information is available, reference resolution performed by succeeding modules in the analysis chain is improved.

#### 4.1.2 Solution

Exploiting the class hierarchy in the ontology during template matching supports the writing of more general templates as well as the processing of referential expressions. In the first case more general classes can be specified in the template and more specialized expressions uttered by the user can be processed. In the second case the expressions uttered by the user are less specific than the conditions in the template but the matching algorithm is designed in a way that matching still works. To produce valid output and to pass the restricted type to the next module, the class of the instance representing the referential expression remains refined during the end of the template application.

To prevent the generation of invalid output the templates are checked against the ontology while they are loaded. In this way, instances of classes not defined in the ontology (e.g., caused by typos in the templates) cannot even be constructed. Additionally this methodology prevents undefined slots or slots with invalid content to be produced.

### 4.2 Multimodal Fusion

The task of a multimodal fusion component within the SMARTKOM system is to combine or integrate the hypotheses produced by the analyzers of the different modalities. For example, a speech recognition hypothesis containing a deictic expression and a simultaneously performed pointing gesture (dereferencing an object displayed on the screen) are fused into a single hypothesis by replacing the deictic expression with the indicated object.

#### 4.2.1 Challenge

However, it is not always the case that a pointing gesture is accompanied by a spoken deictic expression. In

fact it happens quite often that deictic expressions are omitted due to either recognition errors or vague or reduced utterances. There the ontology can be helpful to combine them. Consider for example a user utterance like *tape [this movie]*<sup>5</sup> or an utterance like *tape Matrix*<sup>6</sup> where both utterances are accompanied by a pointing gesture towards an object representing the movie *Matrix* on the screen. Here, it is not clear at which position within the speech recognition hypothesis the analyzed gesture has to be inserted.

#### 4.2.2 Solution

We apply the ontology in order to be able to search for the appropriate position for an analyzed gesture within a speech recognition hypothesis containing no referential expressions. We use the type hierarchy which is an integral part of the ontology to guide the search for the insertion place. The actual search process decomposes into two concurrent levels. At both levels we try to find the insert place by traversing the entire structure of the speech recognition hypothesis and comparing each slot filler to the gesture object. However, both levels of the search process differ in the manner of integration.

At the first level we try to replace only those parts of the speech recognition hypothesis which are compatible with the object referenced by the gesture. This condition is stronger bound to the result of the language understanding process (see Section 4.1) since it restricts the search process to replace only such material which has been explicitly mentioned.

In contrast to the first level of the search process, the second level permits the insertion of analyzed gestures even if the speech recognition hypothesis provides no clue for an appropriate inserting place. Here, we try to extend the speech recognition hypothesis by generating potential slot fillers up to the slot where the analyzed gesture can be inserted.

### 4.3 Semantic Coherence Scoring

Gurevych et al. (2003a) introduce the notion of *semantic coherence* as a measurement for scoring sets of concepts with respect to the existing knowledge representation. They show how that it can be applied to the task of classifying automatic speech recognition hypotheses (SRH) as *coherent* and *incoherent*. In Porzel et al. (2003) the underlying algorithm was contextually enhanced by means of considering the conceptual context, i.e. conceptual representations of preceding utterances, yielding further improvements in the original task. The applica-

<sup>5</sup>The square brackets mark spoken material that was omitted by the speech recognizer.

<sup>6</sup>The language understanding component does not classify *Matrix* as referential expression

tions thereof provide a mechanism that increases the robustness and reliability of multi-modal dialogue systems.

#### 4.3.1 Challenge

One of the major challenges in making an MMDS reliable enough to be deployed in more complex real world applications is an accurate recognition and interpretation of the users' input. In many cases both correct and incorrect representations of the users' utterances are contained in the automatic speech recognizer's n-best lists. Facing multiple representations of a single utterance poses the question, which of the different hypotheses corresponds most likely to the user's utterance. Additionally if a hypothesis has been chosen, a choice has to be made which of the possible interpretations is the best one.

Different methods have been proposed to solve these problems. Frequently, the scores provided by the recognition system itself are used. More recently, also scores provided by the parsing system have been employed, e.g. Engel (2002). In this application, we propose a new ontology-based method and show that knowledge-based scores can be successfully employed to re-score the speech recognition output and its possible interpretations.

#### 4.3.2 Solution

The software for scoring the sets of concepts and classifying them in terms of their semantic coherence employs the ontology described herein. This means, that the ontology crafted as a general knowledge representation for various processing modules of the system is additionally used as the basis for evaluating the semantic coherence of sets of concepts.

The scoring software performs a number of processing steps:

- converting each input, e.g. a SRH, into a concept representation. For this purpose, each entry of the system's lexicon was augmented with zero, one or multiple ontology concepts, this results in sets of potential interpretations of each SRH in the recognizer's output;
- converting the domain model, i.e. an ontology, into a directed graph with concepts as nodes and relations as edges;
- scoring concept representations using a scoring metric that measures the average path-length between all concepts contained in the sets.

For example, in our data (Gurevych et al., 2002) a user expressed the wish to get more information about a specific church, as:

- (1) *Kann ich bitte Informationen zur*  
 May I please Information about the  
*Heiliggeistkirche bekommen*  
 Church of Holy Spirit get

Looking at two SRHs from the ensuing n-best list we found that Example (3) constituted a suitable representation of the utterance, whereas Example (2) constituted a less adequate representation thereof, labeled accordingly by the human annotators:

- (2) *Kann ich Information zur*  
 May I Information about the  
*Heiliggeistkirche kommen*  
 Church of Holy Spirit come
- (3) *Kann ich Information zur*  
 May I Information about the  
*Heiliggeistkirche bekommen*  
 Church of Holy Spirit get

According to the lexicon entries, the SRHs are transformed into two alternative concept representations:

- $CR_1$ :{Person; Information Search Process; Church; Motion Directed Transliterated Process};  
 $CR_2$ :{Person; Information Search Process; Church; Transaction Process}.

The corresponding scores, computed by measuring the average path-length connecting all concepts within the  $CR$ , are normalized as numbers on a scale from 0 to 1, with higher scores indicating better semantic coherence. Then, the resulting score assigned to Example (2) is 0.6, and the score of Example (3) is 0.75. The evaluation of the method against the hand-annotated corpus has shown that it successfully classifies 73.2% in a German corpus of 2.284 speech recognition hypotheses as either coherent or incoherent, given a baseline 54.55% derived from the annotation experiments (the majority class).

An implicit application of the semantic coherence scoring method is the ranking of alternative interpretations of SRHs, resulting from lexical ambiguities, i.e. lexicon entries which feature more than one concept mapping.

- (4) *Ich bin auf dem Philosophenweg*  
 I am on the Philosopher's Walk

Looking at two SRHs from the ensuing n-best list the semantic coherence scoring ranks Example (6) constituted a suitable representation of the utterance, whereas Example (5) constituted a less adequate representation thereof.

- (5) *Spielfilme auf dem Philosophenweg*  
 Feature films on the Philosopher's Walk

- (6) *bin auf dem Philosophenweg*  
 am on the Philosopher's Walk

According to the lexicon entries, the entry for *am* can be mapped onto: *Self Identification Process*, *Static Spatial Process* and *None*, which corresponds to its potential occurrences as an aspectual marker. Based on this ambiguity alone we get:

$CR_1$ :{Self Identification Process; Two Point Relation; Location};  
 $CR_2$ :{Static Spatial Process; Two Point Relation; Location};

The semantic coherence score for  $CR_2$  is the highest (0.87) and all scores for representations of Example (5) as well as for other representations of Example (6), e.g.  $CR_2$  (0.4), are lower. This example shows that the semantic coherence measurement can be applied for ranking competing SRHs as well as for ranking alternative interpretations of the individual hypothesis. An initial evaluation of the word sense disambiguation against an hand-annotated corpus has shown that the system successfully favours the correct interpretation in 85.76% in a German corpus of speech recognition hypotheses containing 323 ambiguous lexical items, given a baseline 43.96% derived from the annotation experiments (the majority class).

#### 4.4 Computing Dialogue Coherence

By viewing the instances of the ontology as typed feature structures we can use unification and unification-like operations for the enrichment and validation of user hypotheses in the discourse module (Pfleger, 2002; Pfleger et al., 2003). This module performs an important processing step in our system: interpretation in context.

##### 4.4.1 Challenge

We are concerned with using the discourse state as represented in the discourse memory (see Pfleger et al. (2003)) and predictions from the action planner (see Section 4.5) for *scoring* and *enriching* the hypotheses.

##### 4.4.2 Solution

Ontologies allow for *closed world reasoning* based on the types in the inheritance hierarchy. In contrast to non-monotonic operations like default unification, e.g., Bouma (1992), which assume an open world we use the types in the hierarchy in a fashion similar to *priority union*, e.g., Grover et al. (1994). As shown in Grover et al. (1994), their default unification allows for a natural and convenient way for interpreting elliptical phenomena. Important here is that this approach together with a proper domain model – our ontology – makes it possible to inherit discourse information by combining

new information with old one in a straightforward way. Another advantage is that we have a well defined operation with a well defined semantics (Alexandersson and Becker, 2003).

The basis for finding a suitable referent using the overlay operation is laid by the discourse memory (Pfleger, 2002; Pfleger et al., 2003). There, a three-tiered discourse structure based on LuperFoy (1992) consisting of a modality layer comprising entry points for resolving anaphoric expressions, a discourse layer which is used to find discourse objects on a semantic level, and finally, a domain layer consisting of the complete instances of the domain model. There is also a double threaded focus structure – a local and global focus structure – which is used to guide the search for possible entry points for the overlay operation. Finally, the objects in the domain layer is partitioned using the ideas of Salmon-Alt (2000).

An indispensable part of the overlay operation is the scoring mechanism described in Pfleger et al. (2002). This extension is essential since the overlay operation *always* succeeds. Therefore we count type clashes (tc), number of information stemming from covering (co) and background (bg) and conflicting values (cv) indicating that we something was overwritten. These parameters are combined using the following formula

$$score(co, bg, tc, cv) = \frac{co + bg - (tc + cv)}{co + bg + (tc + cv)}$$

which together with additional information, e.g., recency, produces the total score for a hypothesis.

We highlight the effect of our ontology with the following excerpt between System and User.

(7) **U:** *What's on TV tonight*

(8) **S:** [Displays a list of films] *Here you see a list of films.*

(9) **U:** *show me the cinema program.*

*Cinema program* and *TV program* have a non-trivial superclass *AvEntertainment* which defines, e.g., the *begin-time*. Overlay initially *assimilates* the background to the type of the covering<sup>7</sup> thereby removing all slots between, in this case, *TV program* and *AvEntertainment*. However, the common slots remain and we inherit the begin-time automatically.

#### 4.5 Dialogue Management

The dialogue manager constructs and executes plans of communicative actions to accomplish the user's goals in the dialogue system.

<sup>7</sup>There is one exception to this rule: in case the background is more special than the covering, then the covering is coerced to the type of the background.

#### 4.5.1 Challenge

In the SMARTKOM system, eleven different applications are integrated, each providing a set of services. These services require cross-application cooperation as well as mixed-initiative subdialogues with the user. The representation of the system domain provided by the ontology was to be used to model these interactions uniformly and consistently. An important aspect we will briefly elaborate on is the benefit of this uniformity in intra-system cooperation with the discourse memory to enhance the reliability of e. g. reference resolution.<sup>8</sup>

#### 4.5.2 Solution

The applications were defined in a plan language that models the actions necessary to carry out processes provided by the ontology corresponding to the services offered by the system. This is done in terms of dialogue games, where the dialogue engine communicates with the user, application modules, or other parts of the dialogue system.

Some applications provide functionality for the user, some implement services for other applications. In terms of the ontology, the applications together with the dialogue system are seen as being able to realise a set of *processes*, requiring or making available the corresponding *roles*. For example, the telephone application can be used to make phone calls, and may trigger a process offered by an addressbook application to look up numbers. The functionality in that case is modeled by a process *PhoneTelecommunication* that comprises sub-processes *ControllingPhoneDevice* (to operate the physical device) and an *InformationSearch* involving an *Addressbook* role (to query a database for phone and fax numbers). A *Number* role is required by the *ControllingPhoneDevice* process and can be obtained by performing the *InformationSearch* process and transferring its result (but also – alternatively – by a process involving typing on a numeric pad).

The operations on processes and roles exchanged between the dialogue manager and its dialogue game partners are encoded in the ontology-derived XML schemata as described in Section 4.6.

Processes and roles that are uniform across communication channels facilitate operations that involve integrating system output, user input and intra-system communication. For example, the user may refer in its input to some previous multimodal system presentation; this reference can be resolved by integrating objects given in the same representation (as was described in the previous Section). Furthermore, the dialogue manager often provides hints about which roles of the processes modelled

---

<sup>8</sup>Note that information specific to user interaction history is maintained outside the dialogue manager in the discourse memory module, which also does contextual enrichment.

by the applications it expects to be referred to by the user (e. g. in response to a question). Again, the fact that user input is already interpreted just in terms of possible roles and processes greatly facilitates taking advantage of those hints.

#### 4.6 Generating Interface Specifications

In this additional application, we proposed to use the knowledge modeled in the ontology as the basis for defining the semantics and the content of information exchanged between various modules of the system (Gurevych et al., 2003b).

##### 4.6.1 Challenge

In NLP systems, modules typically exchange messages, e.g., a parser might get word lattices as input and produce corresponding semantic representations for later processing modules, such as a discourse manager. The increasing employment of XML-based interfaces for agent-based or other multi-blackboard communication systems sets a *de facto* standard for syntax and expressive capabilities of the information that is exchanged amongst modules. The content and structure of the information to be represented are typically defined in corresponding XML schemata (XMLS) or Document Type Definitions (DTD).

As discussed above, ontologies are a suitable means for knowledge representation, e.g. for the definition of an explicit and detailed model of a system's domains. That way, they provide a shared domain theory, which can be used for communication. Additionally, they can be employed for deductive reasoning and manipulations of models. The meaning of ontology constructs relies on a translation to some logic. This way, the inference implications of statements, e.g. whether a class can be related to another class via a subclass or some other relation, can be determined from the formal specification of the semantics of the ontology language. However, this does not make any claims about the syntactic appearance of the representations exchanged, e.g. an ordering of the properties of a class.

An interface specification framework, such as XMLS or DTD, constitutes a suitable means for defining constraints on the syntax and structure of XML documents. Ideally, the definition of the content communicated between the components of a complex dialogue system should relate both the syntax and the semantics of the XML documents exchanged. Those can then be seen as instances of the ontology represented as XMLS-based XML documents. However, this requires that the knowledge, originally encoded in the ontology, is represented in the XMLS syntax.

##### 4.6.2 Solution

The solution proposed states that the knowledge representations to be expressed in XMLS are first modeled in

OIL-RDFS or DAML+OIL as *ontology proper*, using the advantages of ontology engineering systems available, and then transformed into a communication interface automatically with the help of the software developed for that purpose.<sup>9</sup>

Employing this approach, XMLS and DTDs are created such that they:

- stay logically consistent,
- are easy to manage,
- enable a straightforward mapping back to the respective knowledge representation for inference,
- allow the handling of a range of NLP tasks immediately on the basis of XMLS.<sup>10</sup>

The resulting schemata capture the hierarchical structure and a significant part of the semantics of the ontology. We, therefore, provide a standard mechanism for defining XMLS-based interface specifications, which are *knowledge rich*, and thus can be used as a suitable representation of domain and discourse knowledge by NLP components. Since the software that has been developed completely automates the transformation process, the resulting XMLS are congruent with the XML schema specifications. Furthermore, the ontology can be re-used in multiple systems as a single ontology can be used to generate application-specific communication interfaces.

However, the main advantage of our approach is that it combines the power of ontological knowledge representation with the strengths of XMLS as an interface specification framework in a single and consistent representation. Our experience shows, this would not have been possible for a complex dialogue system, if XML schemata were defined from scratch or hand-crafted, and constitutes a step towards building robust and reusable NLP components.

## 5 Concluding Remarks

In this paper, we presented further application of an ontology which is used as a single knowledge representation in a multi-modal and multi-domain dialogue system, namely natural language understanding, mulimodal fusion, semantic and dialogue coherence scoring, dialogue management and interface specification. Additional applications, excluded in this descriptions, can be found in the system's output pipeline, e.g. the dynamic help system and natural language generation.

<sup>9</sup>This is a free software project. The package and respective documentation can be obtained from <http://savannah.nongnu.org/projects/oil2xsd>.

<sup>10</sup>E.g., the discourse and dialogue modules described in the previous Sections operates on the XML schema obtained via ontology transformation.

In our minds, these examples suffice to strengthen the claims made in Gurevych et al. (2003c) substantially. Firstly, this concerns the benefits of using a single knowledge representation throughout a dialogue system as opposed to using multiple knowledge representations and formats. Secondly it concerns the additional advantages of such a homogeneous world model that defines the processing interfaces as well as the system's world knowledge, as costly mappings between them are no more necessary. This means that modules receive only messages whose content is congruent to the terminological and structural distinctions defined in the ontology.

Our additional concern while designing the ontology was the re-usability of this component within our MMDS as well as other NLP systems. So far, the top-level ontology proved stable. We found the extensions on the lower levels of the ontology to be comparatively cheap. This single knowledge base was successfully tested and applied to multiple NLP problems, e.g., resolving bridging expressions in texts as well as for the resolution of metonymical and polysemous utterances next to the SMARTKOM applications described herein.

## Acknowledgments

This work was partially funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the SmartKom project under Grant 01 IL 905 K7 and by the Klaus Tschira Foundation. The responsibility for the contents lies with the authors.

## References

- Jan Alexandersson and Tilman Becker. 2003. The Formal Foundations Underlying Overlay. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg, The Netherlands, February.
- Gosse Bouma. 1992. Feature structures and nonmonotonicity. *Computational Linguistics*, 18(2):183–203.
- Ralf Engel. 2002. SPIN: Language understanding for spoken dialogue systems using a production system approach. In *Proceedings of the International Conference on Speech and Language Processing 2002*.
- Dieter Fensel, Frank van Harmelen, Ian Horrocks, Donald McGuinness, and P. Patel-Schneider. 2001. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2).
- Claire Grover, Chris Brew, Suresh Manandhar, and Marc Moens. 1994. Priority union and generalization in discourse grammars. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 17–24,

- Las Cruces, NM. Association for Computational Linguistics.
- Nicola Guarino and Roberto Poli. 1995. Formal ontology in conceptual analysis and knowledge representation. *Special issue of the International Journal of Human and Computer Studies*, 43.
- Iryna Gurevych, Robert Porzel, and Michael Strube. 2002. Annotating the semantic consistency of speech recognition hypotheses. In *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*, pages 46–49, Philadelphia, USA, July.
- Iryna Gurevych, Rainer Malaka, Robert Porzel, and Hans-Peter Zorn. 2003a. Semantic coherence scoring using an ontology. In *Proceedings of the Joint Human Language Technology and North American Association of Computational Linguistics Conference*. to appear.
- Iryna Gurevych, Stefan Merten, and Robert Porzel. 2003b. Automatic creation of interface specifications from ontologies. In *Proceedings of the HLT-NAACL'03 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*. to appear.
- Iryna Gurevych, Robert Porzel, Elena Slinko, Norbert Pflieger, Jan Alexandersson, and Stefan Merten. 2003c. Less is more: Using a single knowledge representation in dialogue systems. In *Proceedings of the HLT-NAACL'03 Workshop on Text Meaning*, Edmonton, Canada.
- Gerd Herzog, Heinz Kirchmann, Stefan Merten, Alasane Ndiaye, Peter Poller, and Tilman Becker. 2003. Multiplatform testbed: an integration platform for multimodal dialog systems. In *Proceedings of the HLT-NAACL'03 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*. to appear.
- Susann LuperFoy. 1992. The representation of multimodal user interface dialogues using discourse pegs. In *Proceedings of the ACL Conference*, pages 22–31.
- Norbert Pflieger, Jan Alexandersson, and Tilman Becker. 2002. Scoring functions for overlay and their application in discourse processing. In *KONVENS-02*, Saarbrücken, September – October.
- Norbert Pflieger, Jan Alexandersson, and Tilman Becker. 2003. A robust and generic discourse model for multimodal dialogue. In *This volume*.
- Norbert Pflieger. 2002. Discourse processing for multimodal dialogues and its application in SmartKom. Master's thesis, Universität des Saarlandes.
- Robert Porzel, Iryna Gurevych, and Christof E. Müller. 2003. Ontology-based contextual coherence scoring. In *Proceedings of the Fourth SIGdial Workshop on Discourse and Dialogue*, page in press, Sapporo, Japan, July.
- Stuart J. Russell and Peter Norvig. 1995. *Artificial Intelligence. A Modern Approach*. Prentice Hall, Englewood Cliffs, N.J.
- Susanne Salmon-Alt. 2000. Interpreting referring expressions by restructuring context. In *Proceedings of ESSLLI 2000*, Birmingham, UK. Student Session.
- Wolfgang Wahlster, Norbert Reithinger, and Anselm Blocher. 2001. SmartKom: Multimodal communication with a life-like character. In *Proceedings of the 7th European Conference on Speech Communication and Technology*., pages 1547–1550.