SmartWeb Handheld — Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services

Daniel Sonntag, Ralf Engel, Gerd Herzog, Alexander Pfalzgraf, Norbert Pfleger, Massimo Romanelli, and Norbert Reithinger

German Research Center for Artificial Intelligence
66123 Saarbrücken, Germany
firstname.lastname@dfki.de

Abstract. SmartWeb aims to provide intuitive multimodal access to a rich selection of Web-based information services. We report on the current prototype with a smartphone client interface to the Semantic Web. An advanced ontology-based representation of facts and media structures serves as the central description for rich media content. Underlying content is accessed through conventional web service middleware to connect the ontological knowledge base and an intelligent web service composition module for external web services, which is able to translate between ordinary XML-based data structures and explicit semantic representations for user queries and system responses. The presentation module renders the media content and the results generated from the services and provides a detailed description of the content and its layout to the fusion module. The user is then able to employ multiple modalities, like speech and gestures, to interact with the presented multimedia material in a multimodal way.

1 Introduction

The development of a context-aware, multimodal mobile interface to the Semantic Web [1], i.e., ontologies and web services, is a very interesting task since it combines many state-of-the-art technologies such as ontology development, distributed dialog systems, standardized interface descriptions (EMMA[1], SSML[2], RDF[3], OWL-S[4], WSDL[5], SOAP[6], MPEG7[7]), and composition of web services. In this contribution we describe the intermediate steps in the dialog system development process for the project SmartWeb [2], which was started in 2004 by partners in industry and academia.

[1] http://www.w3.org/TR/emma
[2] http://www.w3.org/TR/speech-synthesis
[3] http://www.w3.org/TR/rdf-primer
[4] http://www.w3.org/Submission/OWL-S
[5] http://www.w3.org/TR/wsdl
[6] http://www.w3.org/TR/soap
[7] http://www.chiariglione.org/mpeg

T.S. Huang et al. (Eds.): AI for Human Computing, LNAI 4451, pp. 272–295, 2007.

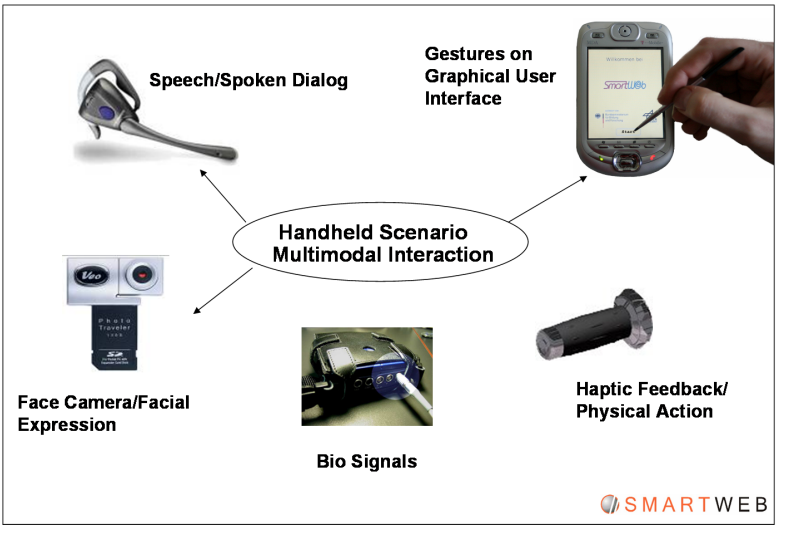


Fig. 1. The multimodal dialog handheld scenario comprises spoken dialog recorded by a bluetooth micro, gestures on the graphical PDA touchscreen, and camera signals. In addition, the SmartWeb project uses recognition of user state in biosignals to adapt system output in stressed car driving situations and haptic input from a force-feedback device installed on a motorbike.

In our main scenario, the user carries a smartphone PDA, as shown in figure 1, and poses closed and open domain multimodal questions in the context of football games and a visit to a Football Worldcup stadium. The PDA serves as an easy-to-use user interaction device which can be queried by natural language speech or handwriting, and which can understand social signaling - hand gestures on the PDA touchscreen and head movement perceived by the PDA camera. By our multimodal dialog interface we aim at providing natural interaction for human users in the Human Computing paradigm [3].

Many challenging tasks, such as interaction design for mobile devices with restricted computing power, have to be addressed: the user should be able to use the PDA as a question answering (QA) system, using speech and gestures to ask for information about players or games stored in ontologies, or other up-to-date information like weather forecast information accessible through web services, Semantic Web pages (Web pages wrapped by semantic agents), or the Internet.

The partners of the SmartWeb project share experience from earlier dialog system projects [4,5,6,7]. We followed guidelines for multimodal interaction, as explained in [8] for example, in the development process of our first demonstrator system [9] which contains the following assets: *multimodality*, more modalities allow for more natural communication, *encapsulation*, we encapsulate the multimodal dialog interface proper from the application, *standards*, adopting to standards opens the door to scalability, since we can re-use ours as well as other's resources, and *representation*. A shared representation and a common

ontological knowledge base eases the data flow among components and avoids costly transformation processes. In addition, semantic structures are our basis for representing dialog phenomena such as multimodal references and user queries. The same ontological query structures are input into the knowledge retrieval and web service composition process.

In the following chapters we demonstrate the strength of Semantic Web technology for information gathering dialog systems, especially the integation of multiple dialog components, and show how knowledge retrieval from ontologies and web services can be combined with advanced dialogical interaction, i.e., system-initiative callbacks, which present a strong advancement to traditional QA systems. Traditional QA realizes like a traditional NLP dialog system a (recognize) - analyze - react - generate - (synthesize) pipeline [10]. Once a query is started, the information is pipelined until the end, which means that the user-system interaction is reduced to user and result messages. The types of dialogical phenomena we address and support include reference resolution, system-initiated clarification requests and pointing gesture interpretation, among others. Support for underspecified questions and enumeration question types additionally shows advanced QA functionality in a multimodal setting. One of the main contributions is the ontology-based integration of verbal and non-verbal system input (fusion) and output (system reaction). System-initiative clarification requests and other pro-active or mixed-initiative system behaviour are representative for emerging multimodal and embedded HCI systems. Challenges for the evaluation of emerging Human Computing applications [11] traces back to challenges in multimodal dialog processing, such as error-prone perception and intergration of multimodal input channels [12,13,14]. Ontology-based integration of verbal and non-verbal system input and output can be seen as groundwork for robust processing of multimodal user input.

The paper is organized as follows: we begin with an example interaction sequence, in section 3, we explain the dialog system architecture. Section 4 describes the ontological knowledge representation, and section 5 the Web Service access. Section 6 then gives a description of the underlying ontology-based language parsing and discourse processing steps as well as their integration into a robust demonstrator system suitable for exhibitions such as CeBIT. Conclusions about the success of the system so far and future plans are outlined in section 7.

2 Multimodal Interaction Sequence Example

The following interaction sequence is typical for the SmartWeb dialog system.

---

(1) **U:** “When was Germany world champion?”
(2) **S:** “In the following 4 years: 1954 (in Switzerland), 1974 (in Germany), 1990 (in Italy), 2003 (in USA)”
(3) **U:** “And Brazil?”

(4) **S:** "In the following 5 years: 1958 (in Sweden), 1962 (in Chile), 1970 (in Mexico), 1994 (in USA), 2002 (in Japan)" + [*team picture, MPEG-7 annotated*]
(5) **U:** Pointing gesture on player *Aldair* + "How many goals did this player score?"
(6) **S:** "Aldair scored none in the championship 2002."

(7) **U:** "What can I do in my spare time on Saturday?"
(8) **S:** "Where?"
(9) **U:** "In Berlin."
(10) **S:** *The cinema program, festivals, and concerts in Berlin are listed.*

The first and second enumeration questions are answered by deductive reasoning within the ontological knowledge base modeled in OWL [15] representing the static but very rich implicit knowledge that can be retrieved. The second example beginning with (7) evokes a dynamically composed web service lookup. It is important to note that the query representation is the same for all the access methods to the Semantic Web (cf. section 6.1) and is defined by foundational and domain-specific ontologies. In a case where the GPS co-ordinates were accessible from the mobile device, the clarification question would have been omitted.

3 Architecture Approach

A flexible dialog system platform is required in order to allow for true multi-session operations with multiple concurrent users of the server-side system as well as to support audio transfer and other data connections between the mobile device and a remote dialog server. These types of systems have been developed, like the Galaxy Communicator [16] (cf. also [17,18,19,20]), and commercial platforms from major vendors like VoiceGenie, Kirusa, IBM, and Microsoft use X+V1, HTML+SALT2, or derivatives for speech-based interaction on mobile devices. For our purposes these platforms are too limited. To implement new interaction metaphors and to use Semantic Web based data structures for both dialog system internal and external communication, we developed a platform designed for Semantic Web data structures for NLP components and backend knowledge server communication. The basic architecture is shown in figure 2.

It consists of three basic processing blocks: the PDA client, the dialog server, which comprises the dialog manager, and the Semantic Web access system.

On the PDA client, a local Java-based control unit takes care of all I/O, and is connected to the GUI-controller. The local VoiceXML-based dialog system resists on the PDA for interaction during link downtimes.

The dialog server system platform instantiates one dialog server for each call and connects the multimodal recognizer for speech and gesture recognition. The

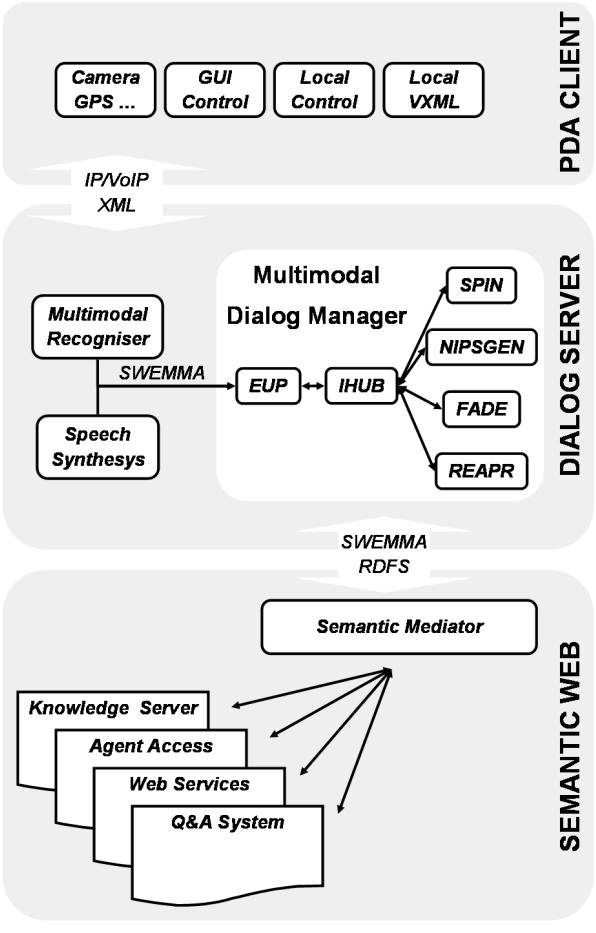


Fig. 2. SmartWeb handheld architecture

dialog system instantiates and sends the requests to the *Semantic Mediator*, which provides the umbrella for all different access methods to the Semantic Web we use. It consists of an open domain QA system, a Semantic Web service composer, Semantic Web pages (wrapped by semantic agents), and a knowledge server.

The dialog system consist of different, self-contained processing components. To integrate them we developed a Java-based hub-and-spoke architecture [21]. The most important processing modules in the dialog system connected to the IHUB are: a speech interpretation component (SPIN), a modality fusion and discourse component (FADE), a system reaction and presentation component (REAPR), and a natural language generation module (NIPSGEN), all discussed in section 6. An EMMA Unpacker/Packer (EUP) component provides the communication with the dialog server and Semantic Web subsystem external to the

multimodal dialog manager and communicates with the other modules of the dialog server, the multimodal recognizer, and the speech synthesis system.

Processing a user turn, normal data flows through $SPIN \rightarrow FADE \rightarrow REAPR \rightarrow SemanticMediator \rightarrow REAPR \rightarrow NIPSGEN$. However, the data flow is often more complicated when, for example, misinterpretations and clarifications are involved.

4 Ontology Representation

The ontological infrastructure of the SmartWeb dialog system project, the SWIntO (**S**mart**W**eb **Int**egrated **O**ntology) [22], is based on an upper model ontology realized by merging well chosen concepts from two established foundational ontologies, DOLCE [23] and SUMO [24], into a unique one: the SmartWeb foundational ontology SmartSUMO [25]. Domain specific knowledge (sportevent, navigation) is defined in dedicated ontologies modeled as subontologies of the SmartSUMO. The SWIntO integrates question answering specific knowledge of a discourse ontology (DiscOnto) and representation of

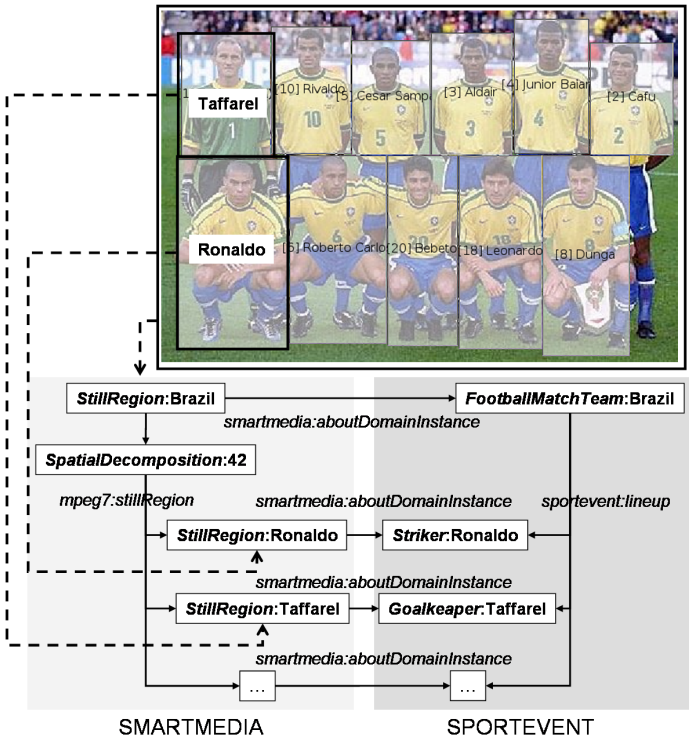


Fig. 3. A SmartMedia instance representing the decomposition of the Brazil 1998 world cup football team image

multimodal information of a media ontology (SMARTMEDIA). The data exchange is RDF-based.

4.1 The Upper Model

In order to integrate knowledge from different domains we created an abstract foundational ontology that models basic properties and a common relational background for interoperability. Each domain specific ontology has been then aligned to this foundational ontology. The process of creation of SMARTSUMO has been constrained by two essential principles: the ontology must offer a *rich axiomatization* with a high abstraction level and cover a large number of general concepts. The ontology should also be *descriptive* for modeling artifacts of human common sense and give the possibility of modeling entities extended in time and space. Therefore the SMARTSUMO requires a *perdurantism* approach. From about a dozen freely available foundational ontologies (see [26] for an overview) the DOLCE and the SUMO ontology were selected as being the best fit for the given task. Both ontologies have been modified and combined. The upper level of SMARTSUMO is basically derived from DOLCE with the distinction between *endurant*, *perdurant*, *abstract* and *qualities*, and the rich axiomatisation that allows the modelling of location in space and time, and the use of relations such as parthood and dependence. We also borrowed the ontology module *Descriptions & Situations* [27] from DOLCE. From this minimal core of generic concepts we aligned the rich SUMO taxonomy.

4.2 The DiscOnto Ontology

We created a discourse ontology (DISCONTO) with particular attention to the modeling of discourse interactions in QA scenarios. The DISCONTO provides concepts for dialogical interaction with the user as well as more technical request-response concepts for data exchange with the Semantic Web subsystem including answer status, which is important in interactive systems. In particular DISCONTO comprises concepts for multimodal dialog management, a dialog act taxonomy, lexical rules for syntactic-semantic mapping, HCI concepts (e.g. pattern language for interaction design [28]), and concepts for questions, question focus, semantic answer types [29], and multimodal results [30].

Information exchange between the components of the server-side dialog system is based on the W3C EMMA standard that is used to realize containers for the ontological instances representing, e.g., multimodal input interpretations. SWEMMA is our extension of the EMMA standard which introduces additional *Result* structures in order to represent components output. On the ontological level we modeled an RDF/S-representation of EMMA/SWEMMA.

4.3 The Smartmedia Ontology

The SMARTMEDIA is an MPEG7-based media ontology and an extension to [31,32] that we use to represent output result, offering functionality for multimedia decomposition in space, time and frequency (mpeg7:SegmentDecomposition),

file format and coding parameters (mpeg7:MediaFormat), and a link to the Upper Model Ontology (smartmedia:aboutDomainInstance). In order to close the semantic gap between the different levels of media representations, the *smartmedia:aboutDomainInstance* property has been located in the top level class *smartmedia:Segment*. The link to the upper model ontology is inherited to all segments of a media instance decomposition to guarantee deep semantic representations for the *smartmedia* instances referencing the specific media object and for making up segment decompositions [33].

Figure 3 shows an example of this procedure applied to an image of the Brazilian football team in the final match of the World Cup 1998, as introduced in the interaction example. In the example an instance of the class *mpeg7:StillRegion*, representing the complete image, is decomposed into different *mpeg7:StillRegion* instances representing the segments of the image which show individual players.

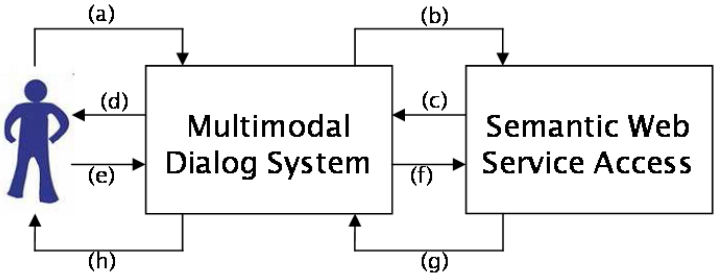
The *mpeg7:StillRegion* instance representing the entire picture is then linked to a *sportevent:MatchTeam* instance, and each segment of the picture is linked to a *sportevent:FieldFootballPlayer* instance or sub-instance. These representations offer a framework for gesture and speech fusion when users interact with Semantic Web results such as MPEG7-annotated images, maps with points-of interest, or other interactive graphical media obtained from the ontological knowledge base or multimedia web services.

5 Multimodal Access to Web Services

To connect to web services we developed a semantic representation formalism based on OWL-S and a service composition component able to interpret an ontological user query. We extended the OWL-S ontologies to flexibly compose and invoke web services on the fly, gaining sophisticated representation of information gathering services fundamental to SmartWeb.

Sophisticated data representation is the key for developing a composition engine that exploits the semantics of web service annotation and query representation. The composition engine follows a plan-based approach as explained, e.g., in [34]. It infers the initial and goal state from the semantic representation of the user query, whereas the set of semantic web services is considered as planning operators. The output gained from automatic web service invocation is represented in terms of instances of the SmartWeb domain ontologies and enriched by additional media instances, if available. Media objects are represented in terms of the Smartmedia ontology (see above) and are annotated automatically during service execution. This enables the dialog manager for multimodal interaction with web service results.

A key feature of the service composition engine is to detect underspecified user queries, i.e., the lack of required web service input parameters. In these cases the composition engine is able to formulate a clarification request as specified within the discourse ontology (DiscOnto). This points out the missing pieces of



(a) User query: What can I do in my spare time on Saturday?
(b) Ontological user query is sent to web services.
(c) Clarification request (asking for a city) is sent back.
(d) Verbalized clarification request: Where?
(e) User clarification response: In Berlin.
(f) Completed ontological query is sent to web services.
(g) Ontological result of service execution is sent to dialog.
(h) Generated results are multimodally presented to the user.

Fig. 4. Data flow for the processing of a clarification request as in the example (7-10) "What can I do in my spare time on Saturday?"

information to be forwarded to the dialog manager. Then the composition engine expects a clarification response enabling it to replan the refined ontological user query.

According to the interaction example (7-10) the composition engine searches for a web service demanding an activity event type and gets its description. Normally, the context module incorporated in the dialog manager would complete the query with the venue obtained from a GPS receiver attached to the handheld device. In the case of no GPS signal, for instance indoors, the composition engine asks for the missing parameter (cf. figure 4), which makes the composition engine more robust and thus more suitable for interactive scenarios.

In the interaction example (7-10) the composition planner considers the *T-Info EventService* appropriate for answering the query. This service requires both date and location for looking up events. While the date is already mentioned in the initial user query, the location is then asked of the user through clarification request. After the location information (dialog step (9) in the example:*In Berlin*) is obtained from the user, the composition engine invokes in turn two T-Info (DTAG) web services[8] offered by Deutsche Telekom AG (see also [35]): first the *T-Info EventService* as already mentioned above, and then the *T-Info MapService* for calculating an interactive map showing the venue as point-of-interest. Text-based event details, additional image material, and the location map are semantically represented (the map in MPEG7) and returned to the dialog engine.

[8] http://services.t-info.de/soap.index.jsp

6 Semantic Parsing and Discourse Processing

Semantic parsing and other discourse processing steps are reflected on the interaction device as advanced user perceptual feedback functionality. The following screenshot illustrates the two most important processing steps for system-user interaction, the feedback on the natural language understanding step and the presentation of multimodal results. The semantic parser produces a semantic query (illustrated on the left in figure 5), which is presented to the user in nested attribute-value form. The web service results (illustrated on the right in figure 5) for the interaction example (7-10) are presented in a multimodal way, combining text, image, and speech: *5 Veranstaltungen* (five events).

Fig. 5. Semantic query (illustrated on the left) and web service results (illustrated on the right)

6.1 Language Understanding with SPIN and Text Generation with NIPSGEN

Language Understanding

The parsing module is based on the semantic parser SPIN [36]. A syntactic analysis of the input utterance is not performed, but the ontology instances are created directly from word level. The typical advantages of a semantic parsing approach are that processing is faster and more robust against speech recognition errors and disfluencies produced by the user and the rules are easier to write and maintain. Also, multilingual dialog systems are easier to realize, as a syntactic analysis is not required for each supported language. A disadvantage is that

the complexity of the possible utterances is somewhat limited, but – in our experience – this is acceptable for dialog systems.

Several semantic parsers were developed for spoken dialog systems. Most of them use as underlying formalisms context free grammars (CFGs), e.g., [37] or finite state transducers (FSTs), e.g., [38] or variants of them, e.g., [39,40].

The SPIN parser uses a more powerful rule language to simplify writing of rules and to reduce the amount of required rules.

Properties of the rule language include:

- Direct handling of nested typed feature structure is available, which is important for processing more complex utterances.
- Order-independent matching is supported, i.e., the order of matched input elements is not important. This feature helps with the processing of utterances in free word order languages, like German, Turkish, Japanese, Russian or Hindi, and simplifies the writing of rules that are robust against speech recognition errors and disfluencies produced by the user. The increased robustness is achieved, as the parts of the utterance that are recognized incorrectly can be skipped. This is a mechanism that is also used in other approaches, e.g., [41].
- Built-in support for referring expressions is available.
- Regular expressions are available. Formulating the rules in a more elegant way is supported by this feature whereby the amount of required rules is reduced. Furthermore, the writing of robust rules is simplified.
- Constraints over variables and action functions are supported providing enough flexibility for real-world dialog system. Especially, if the ontology is developed without the parsing module in mind, flexibility is highly demanded.

SPIN's powerful rule language requires an optimizing parser, otherwise processing times would not be acceptable. Principally, the power of the rule language avoids the development of a parser which delivers sufficient performance for an arbitrary rule set. In particular, order-independent matching makes efficient parsing much harder, parsing of arbitrary grammars is NP-complete, see also [42]. Therefore, the parser is tuned for rule sets that are typical for dialog systems. A key feature achieving fast processing is the pruning of results that can be regarded as irrelevant for further processing within the dialog system. Pruning of results means that the parsing algorithm is not complete. Pruning of irrelevant results is achieved using a fixed application order for the rules in combination with tagging some of the rules as destructive. More details of the parsing algorithm can be found in [36]. The rule set used for the SmartWeb project consists of 1069 rules where 363 rules are created manually, and 706 are generated automatically from the linguistic information stored in SWIntO, e.g., country names. The lexicon contains 2250 entries. Currently, the knowledge base for the SmartWeb system consists of 1069 rules whereby 363 rules were created manually, and 706 were generated automatically from the linguistic information stored in SWIntO, e.g., country names. The lexicon contains 2250 entries. The average processing time is about 50ms per utterance, which ensures direct feedback to user inputs.

To demonstrate processing of rules, four rules are provided as examples of how to process the utterance *When was Brazil world champion?*. The first one transforms the word *Brazil* into the ontology instance `Country`:

```
Brazil
→ Country(name:BRAZIL)
```

The second rule transforms countries to teams, as each country can stand for a team in our domain:

```
$C=Country()
→ Team(origin:$C)
```

The third rule processes *when*, generating an instance of the type `TimePoint` which is marked as questioned:

```
when
→ TimePoint(variable:QEVariable(focus:text))
```

The fourth rule processes the verbal phrase *<TimePoint> was <Team> world champion*

```
$TP=TimePoint() was $TM=Team() world champion
→ QEPattern(patternArg:Tournament(
                winner:$TM, happensAt:$TP))
```

Text Generation

Within the dialog system, the text generation module is used within two processing steps. First, the abstract interpretation of the user utterance is shown as human readable text, called paraphrase. This allows the user to check if the query has been interpreted with the desired meaning and if all of the provided information has been included. Second, the search results represented as instances of SWIntO are verbalized.

The text generation module uses the same SPIN parser that is used in the language understanding module together with a TAG (tree adjoining grammar) grammar module [43]. The TAG grammar in this module is derived from the XTAG grammar for English developed at the University of Pennsylvania.[9]

The inputs of the generation module are instances of SWIntO representing the search results. Then these results are verbalized in different ways, e.g., as a heading, as an image description, as a row of a table, or as a text which is synthesized. A processing option indicates the current purpose. Figure 6 shows an example containing different texts.

The input is transformed into an utterance in four steps:

1. An intermediate representation is built up on a phrase level. The intermediate representation is introduced, as a direct generation of the TAG tree description would lead to overly complicated and unintuitive rules. The required rules are domain dependent.

[9] http://www.cis.upenn.edu/∼xtag/

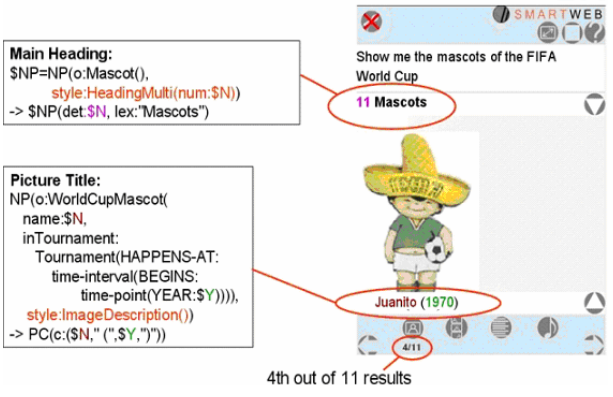


Fig. 6. The verbalized result for the utterance *Show me the mascots of the FIFA World Cup*. The rules generating the main heading and the image description are shown on the left.

2. A set of domain independent rules transforms the intermediate representation to a derivation tree for the TAG-grammar. Each tree in the TAG grammar has a corresponding type in the ontology of the text generation module. The features of a TAG tree type represent the type of operation (adjunction (`a`), substitution (`s`), lexical replacement (`l`)) and the position in the tree, e.g., `211`.
3. The actual syntax tree is constructed using the derivation tree. After the tree has been built up, the features of the tree nodes are unified.
4. The correct inflections for all lexical leafs are looked up in the lexicon. Traversing the lexical leafs from left to right produces the result text.

For text generation, the parser is driven in a slightly different mode: The automatic ordering of rules is switched off, instead the order in which the rules are applied is taken from the file containing the rules. Regions that have to be applied in a loop and rules that have to be applied optionally are marked explicitly. In the current system, two loops exists, one for each phase. A more detailed description of the text generation module can be found in [44].

In the SmartWeb system currently 179 domain dependent generation rules and 38 domain independent rules are used.

6.2 Multimodal Discourse Processing with FADE

An important aspect of SmartWeb is its context-aware processing strategy. All recognized user actions are processed with respect to their situational and discourse context. A user is thus not required to pose separate and unconnected questions. In fact, she might refer directly to the situation, e.g., *"How do I get to Berlin from here?"*, where *here* is resolved from GPS information, or to previous contributions (as in the elliptical expression *"And in 2002?"* in the context of a previously posed question *"Who won the Fifa World Cup in 1990?"*). The

interpretation of user contributions with respect to their discourse context is performed by a component called *Fusion and Discourse Engine*—FADE [45,46][10]. The task of FADE is to integrate the verbal and nonverbal user contributions into a coherent multimodal representation to be enriched by contextual information, e.g., resolution of referring and elliptical expressions.

The basic architecture of FADE consists of two interwoven processing layers: (1) a production rule system—PATE—that is responsible for the reactive interpretation of perceived monomodal events, and (2) a discourse modeler—DiM—that is responsible for maintaining a coherent representation of the ongoing discourse and for the resolution of referring and elliptical expressions.

In the following two subsections we will briefly discuss some context-related phenomena that can be resolved by FADE.

Resolution of referring expressions. A key feature of the SmartWeb system is that the system is capable of dealing with a broad range of referring expressions as they occur in natural dialogs. This means the user can employ deictic references that are accompanied by a pointing gesture (such as in *"How often did this team [pointing gesture] win the World Cup?"*) but also—if the context provides enough disambiguating information—without any accompanying gestures (e.g., if the previous question is uttered in the context of a previous request like *"When was Germany World Cup champion for the last time?"*).

Moreover, the user is also able to utter time deictic references as in *"What's the weather going to be like tomorrow?"* or *"What's the weather going to be like next Saturday?"*.

Another feature supported by FADE is the resolution of *cross modal* spatial references, i.e., a spoken reference to visually displayed information. The user can refer, for example, to an object that is currently displayed on the screen. If a picture of the German football team is displayed, the system is able to resolve references like *"this team"* even when the team has not yet been mentioned verbally. MPEG7-annotated images (see section 4) even permit spatial references to objects displayed within pictures, e.g., as in *"What's the name of the guy to the right of Ronaldo?"* or *"What's the name of the third player in the top row?"*.

Resolution of elliptical expression. Humans tend to keep their contributions as short and efficient as possible. This is particularly the case for follow-up questions or answers to questions. Here, people often make use of elliptical expressions, e.g., when they ask a follow-up question *"And the day after tomorrow?"* in the context of a previous question *"What's the weather going to be like tomorrow?"*. But even for normal question-answer pairs people tend to omit everything that has already been conveyed by the question (User: *"Berlin"* in the context of a clarification question of the system like *"Where do you want to start?"*; see section 5).

Elliptical expressions are processed in SmartWeb as follows: First, SPIN generates an ontological query that contains a semantic representation of the

[10] The situational context is maintained by another component called *SitCom* that is not discussed in this paper (see [47]).

elliptical expression, e.g., in case of the aforementioned example “Berlin”. This analysis would only comprise an ontological instance representing the city Berlin. FADE in turn, then tries to integrate the elliptical expression with the previous system utterance, if this was a question. Otherwise it tries to integrate the elliptical expression with the previous user request. If the resolution succeeded, the resulting interpretation either describes the answer to the previous clarification question, or it describes a new question.

OnFocus/OffFocus identification. An important task for mobile, speech-driven interfaces that support an open-microphone[11] is the continuous monitoring of all input modalities in order to detect when the user is addressing the system. In the mobile scenario of SmartWeb, the built-in camera of the MDA Pro handheld can be used to track whether a user is present. This camera constantly captures pictures of the space immediately infront of the system. These pictures are processed by a server-side component that detects whether the user is looking at the device or not.

In SmartWeb, there are two components that determine the attentional state of the user: (i) the OnView recognizer, and the (ii) the OnTalk recognizer. The task of the OnView recognizer is to determine whether the user is looking at the system or not. The OnView-Recognizer analyzes a video signal captured by a video camera linked to the mobile device and determines for each frame whether the user is in OnView or OffView mode (figure 7 shows two still images of these different modes).

Fig. 7. Two still images illustrating the function of the OnView/OffView recognizer: The image on the left shows the OnView case and the one the right shows the Offview case

The task of the OnTalk recognizer is to determine whether a user’s utterance is directed to the system. To this end, the OnTalk recognizer analyzes the speech signal and computes about 99 prosodic features based on F0, energy, duration,

[11] Open-microphone means the microphone is always active so that the user can interact with the system without further activation. In contrast to an open-microphone interface, systems often require the user to push some hard- or software button in order to activate the system (i. e., a *push-to-activate* button).

jitter, and shimmer (see [48]). This is done for each word but the final result is averaged over the complete turn. Both recognizers provide a score reflecting the individual confidence of a classification.

FADE receives and stores the results of the OnView recognizers as a continuous stream of messages (i. e., every time the OnView state changes, FADE receives an update). OnTalk/OffTalk classifications are only sent to FADE if the speech recognition components detected some input event. The actual algorithm goes as follows: The overall idea is to combine the two distinct classifications for OnView/OffView and OnTalk/OffTalk in order to compensate for potential classification errors. If the OnView value is above 0.3 (where 0 means OffView and 1 means OnView), the OffTalk value must be very low (below 0.2) in order to classify a contribution as OffFocus. Otherwise, a OnTalk value below 0.5 is already sufficient to classify an utterance as OffFocus.

6.3 Reaction and Presentation Planning for the Semantic Web

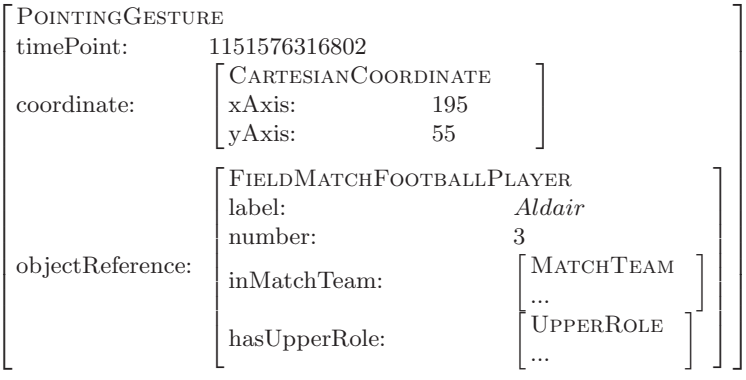
An integral part of dialog management is the reaction and presentation module (REAPR). It manages the dialogical interaction for the supported dialog phenomena such as flexible turn-taking, incremental processing, and multimodal fusion of system output. REAPR is based on a finite-state-automaton and information space (IS). The FSA makes up the integral part of the dialog managment decisions in the specific QA domain we model. The dialog structure that is embedded and committed by the transitions of the FSA allows for a declarative control mechanism for reaction and presentation behaviour.

Our new approach differs from other IS approaches (e.g. [49]) by generating IS features from the ontological instances generated during dialog processing [50].[12]

Since the dialog ontology is a model for multimodal interaction, multimodal MPEG7 result representations, multimodal result presentations, dialog state, and (agent) communication with the backend knowlege servers, large information spaces can be extracted from the ontological instances describing the system and user turns in terms of special dialog acts - to ensure accurate dialog management capabilities. REAPR decides, for example, if a semantic query is acceptable for transfer to the Semantic Mediator. The IS approach to dialog modeling comprises, apart from dialog moves and update strategies, a description of informational components (e.g. common ground) and their formal representations. Since in REAPR the formal dialog specification consists of ontological Semantic Web data structures, a formal well-defined complement to previous formal logic-based operators and Discourse Representation Structures (DRS) is provided. However, the ontological structures resemble the typed feature structures (TFS) [51] that we use for illustration further down. During interaction, many message transfer processes take place, mainly for query recognition and query processing, all of which are based on Semantic Web ontological structures, and REAPR is involved

[12] The IS state is traditionally divided into global and local variables which make up the knowledge state at a given time point. Ontological structures that change over time vastly enhance the representation capabilities of dialog management structures, or other structures like queries from which relevant features can also be extracted.

in many of them. Here we give an example of ontological representations of user pointing gestures (dialog step (5) in the interaction example) which are obtained from the PDA and transformed into ontology-structures to be used by the input fusion module. The following figure shows the ontological representation of a pointing gesture as TFS.



It is important to mention that dialog reaction behaviour within SmartWeb is governed by the general QA scenario, which means that almost all dialog and system moves relate to questions, follow-up questions, clarifications, or answers. As these dialog moves can be regarded as adjacency pairs, the dialog behaves according to some finite state grammar for QA, which makes up the automaton part (FSA) in REAPR. The finite state approach enhances robustness and portability and allows to demonstrate dialog management capabilities even before the more complex IS states are available to be integrated into the reaction and presentation decision process.

6.4 Information States for QA

Information state theory of dialog modelling consists basically of a description of informal components (e.g., obligations, beliefs, desires, intentions) and their formal representation [52]. IS states as envisioned here do not declare update rules and an update strategy (for e.g. discourse obligations [53]) because the data-driven approach is pattern-based, using directly observable processing features, which complements an explicit manual formulation of update rules. Since the dialog ontology is a formal representation model for multimodal interaction, multimodal MPEG-7 result representations [30], result presentations [28], dialog state, and (agent) communication with the backend knowledge servers, large information spaces can be extracted from the ontological instances describing the system and user turns in terms of realized dialog acts.

The turn number represents our first FSA extension to IS with the result of increased flexibility to user replies. Replies which are not specified in a pathway,

Table 1. IS Feature Classes and Features

Feature Class	IS State Features
MMR	*Listening, Recording, Barge-in, Last-ok,*
	Input dominance (text or voice)
NLU	*Confidence, Domain relevance*
Query	*Dialog act, Focus medium,*
	Complexity, Context object, Query text
Fusion	*Fusion act, Co-reference resolution*
Answer	*Success, Speed, Answer streams, Status,*
	Anser type, Content, Answer text
Manager	*Turn/Task numbers, Idle states,*
	Waiting for Results, User/system turn,
	Elapsed times: input/output,
	Dialog act history (system and user)
	e.g. reject, accept, clarify

are not considered erroneous by default, since the IS now contains a new turn value. Ontologicial features for IS extraction under investigation are summarised in table 1.

In previous work on dialog management adaptations [54,55,56], reinforcement learning was used, but large state spaces with more than about five non-binary features are still hard to deal with. As seen in table 1, more than five relevant features can easily be declared. Since our optimisation problem can be formulated at very specific decisions in dialog management due to the FSA ground control, less training material for larger feature extractions is to be expected.

Relevance selection of ontology-based features is the next step for ontology-based dialog management adaptations. In the context of Human Computing one question is how prior user knowledge can be incorporated in order to select relevant features so as to converge faster toward more effective and natural dialog managers. We already incorporated human dialog knowledge by the dialog FSA structure. In a more user-centered and dynamic scenario, the user in the loop should accelerate Learning [57] by e.g. selecting the IS features that are most relevant in the specific dialog application. The human-user interaction for this selection process is of particular interest in dialog applications. Is it possible to integrate the feature selection process into a normal dialog session that the user and the dialog system engage in? In the context of the SmartWeb project we will develop a tool to run the dialog system with the additional possibility to interfere in the dialog management in case the user is not satisfied with the processing. Our future plans include measuring when the direct user feedback is likely to be useful for adapting dialog management strategies automatically. One example is to generate useful reactions in cases where the natural language understanding component fails. Whenever there is the freedom to formulate statements, which is a precondition for natural language communication, understanding may be difficult. What can be done in such cases is to produce useful

reactions and to give hints to the user or examples that the use of supported terminology is not insisted, but at least directed.

6.5 Dialog Components Integration

In this section we will focus on issues of interest pertaining to the system integration. In the first instance, dialog component integration is an integration on a conceptual level. All dialog manager components communicate via ontology instances. This assumes the representation of all relevant concepts in the foundational and domain ontologies – which is hard to provide at the beginning of the integration. In our experience, using ontologies in information gathering dialog systems for knowledge retrieval from ontologies and web services in combination with advanced dialogical interaction is an iterative ontology engineering process. This process requires very disciplined ontology updates, since changes and extensions must be incorporated into all relevant components. The additional modeling effort pays off when regarding the strength of this Semantic Web technology for larger scale projects.

We first built up an initial discourse ontology of request-response concepts for data exchange with the Semantic Web sub-system. In addition, an ontological dialog act taxonomy has been specified, to be used by the semantic parsing and discourse processing modules. A great challenge is the mapping between semantic queries and the ontology instances in the knowledge base. In our system, the discourse (understanding) specific concepts have been linked to the foundational ontology and, e.g., the sportevent ontology, and the semantic parser only builds up interpretations with SWIntO concepts. Although this limits the space of possible interpretations according to the expressivity of the foundational and domain ontologies, the robustness of the system is increased. We completely circumvent the problem of concept and relation similarity matching between conventional syntactic/semantic parsers and backend retrieval systems.

Regarding web services we transform the output from the web services, in particular maps with points of interest, into instances of the SmartWeb domain ontologies for the same reasons of semantic integration. As already noted, ontological representations offer a framework for gesture and speech fusion when users interact with Semantic Web results such as MPEG7-annotated images and maps. Challenges in multimodal fusion and reaction planning can be addressed by using more structured representations of the displayed content, especially for pointing gestures, which contain references to player instances after integration. We extended this to pointing gesture representations on multiple levels in the course of development, to include representations of the interaction context, the modalities and display patterns used, and so on.

The primary aim is to generate structured input spaces for more context-relevant reaction planning to ensure naturalness in system-user interactions to a large degree. Currently, as shown in chapter 6.2, we are experimenting with the MDA's camera input indicating whether the user is looking at the device, to combine it with other indicators to a measure of user focus. The challenge of integrating and fusing multiple input modalities can be reduced by ontological

representations, which exist at well-defined time-points, and are also accessible to other components such as the semantic parser, or the reaction and presentation module.

7 Conclusions

We presented a mobile system for multimodal interaction with an ontological knowledge base and web services in a dialog-based QA scenario. The interface and content representations are based on W3C standards such as EMMA and RDF. The world knowledge shared in all knowledge-intensive components is based on the existing ontologies SUMO and DOLCE, for which we added additional concepts for QA and multimodal interaction in a discourse ontology branch.

We presented the development of the second demonstrator of the SmartWeb system which was successfully demonstrated in the context of the Football World Cup 2006 in Germany. The SWIntO ontology now comprises 2308 concept classes, 1036 slots and 90522 instances.[13] For inference and retrieval the ontology constitutes 78385 data instances after deductions.[14] The answer times are in a 1 to 15 seconds time frame for about 90% of all questions. In general, questions without images and videos as answers can be processed much faster. The web service composer addresses 25 external services from traveling (navigation, train connections, maps, hotels), event information, points of interest (POIs), product information (books, movies), webcam images, and weather information.

The SmartWeb architecture supports advanced QA functionalities such as flexible control flow to allow for clarification questions of web services when needed, long- and short-term memory provided by distributed dialog management in the fusion and discourse module and in the reaction and presentation module, as well as semantic interpretations provided by the speech interpretation module. This can be naturally combined with dialog system strategies for error recoveries, clarifications with the user, and multimodal interactions. Support for inferential, i.e., deductive reasoning, which we provide, complements the requirements for advanced QA in terms of information- and knowledge retrieval. Integrated approaches as presented here rely on ontological structures and a deeper understanding of questions, not at least to provide a foundation for result provenance explanation and justification. Our future plans on the final six month agenda include dialog management adaptations via machine learning and collaborative filtering of redundant results in our multi-user enviroment, and incremental presentation of results.

Acknowledgments

The research presented here is sponsored by the German Ministry of Research and Technology (BMBF) under grant 01IMD01A (SmartWeb). We thank our

[13] The SWIntO can be downloaded at the SmartWeb homepage for research purposes.
[14] The original data instance set was 175293 instances, but evoked processing times up to two minutes for single questions by what interactivity was no longer guaranteed.

student assistants and the project partners. The responsibility for this papers lies with the authors.

References

1. Fensel, D., Hendler, J.A., Lieberman, H., Wahlster, W., eds.: Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. In Fensel, D., Hendler, J.A., Lieberman, H., Wahlster, W., eds.: Spinning the Semantic Web, MIT Press (2003)
2. Wahlster, W.: SmartWeb: Mobile Applications of the Semantic Web. In Dadam, P., Reichert, M., eds.: GI Jahrestagung 2004, Springer (2004) 26–27
3. Pantic, M., Pentland, A., Nijholt, A., Huang, T.: Human computing and machine understanding of human behavior: a survey. In: ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces, New York, NY, USA, ACM Press (2006) 239–248
4. Wahlster, W., ed.: VERBMOBIL: Foundations of Speech-to-Speech Translation. Springer (2000)
5. Wahlster, W.: SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell. In Krahl, R., Günther, D., eds.: Proc. of the Human Computer Interaction Status Conference 2003, Berlin, Germany, DLR (2003) 47–62
6. Reithinger, N., Fedeler, D., Kumar, A., Lauer, C., Pecourt, E., Romary, L.: MIAMM - A Multimodal Dialogue System Using Haptics. In van Kuppevelt, J., Dybkjaer, L., Bernsen, N.O., eds.: Advances in Natural Multimodal Dialogue Systems. Springer (2005)
7. Wahlster, W.: SmartKom: Foundations of Multimodal Dialogue Systems (Cognitive Technologies). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
8. Oviatt, S.: Ten myths of multimodal interaction. Communications of the ACM **42**(11) (1999) 74–81
9. Reithinger, N., Bergweiler, S., Engel, R., Herzog, G., Pfleger, N., Romanelli, M., Sonntag, D.: A Look Under the Hood Design and Development of the First SmartWeb System Demonstrator. In: Proceedings of 7th International Conference on Multimodal Interfaces (ICMI 2005), Trento, Italy (October 04-06 2005)
10. Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., Stent, A.: An Architecture for a Generic Dialogue Shell. Natural Language Engineering **6**(3) (2000) 1–16
11. Poppe, R., Rienks, R.: Evaluating the future of hci: Challenges for the evaluation of upcoming applications. In: Proceedings of the International Workshop on Artificial Intelligence for Human Computing at the International Joint Conference on Artificial Intelligence IJCAI'07, Hyderabad, India (2007) 89–96
12. Oviatt, S.: Multimodal Interfaces. In: The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. Lawrence Erlbaum Assoc. (2003) 286–304
13. Wasinger, R., Wahlster, W.: The Anthropomorphized Product Shelf: Symmetric Multimodal Interaction with Instrumented Environments. In Aarts, E., Encarnao, J.L., eds.: Chapter in: True Visions: The Emergence of Ambient Intelligence. Springer-Verlag, Berlin, Heidelberg, Germany (2006)

14. Wahlster, W.: Towards symmetric multimodality: Fusion and fission of speech, gesture, and facial expression. In: KI. (2003) 1–18
15. Krotzsch, M., Hitzler, P., Vrandecic, D., Sintek, M.: How to reason with OWL in a logic programming system. In: Proceedings of RuleML'06. (2006)
16. Cheyer, A.J., Martin, D.L.: The Open Agent Architecture. Autonomous Agents and Multi-Agent Systems **4**(1–2) (2001) 143–148
17. Seneff, S., Lau, R., Polifroni, J.: Organization, Communication, and Control in the Galaxy-II Conversational System. In: Proc. of Eurospeech'99, Budapest, Hungary (1999) 1271–1274
18. Thorisson, K.R., Pennock, C., List, T., DiPirro, J.: Artificial intelligence in computer graphics: A constructionist approach. Computer Graphics (February 2004) 26–30
19. Herzog, G., Ndiaye, A., Merten, S., Kirchmann, H., Becker, T., Poller, P.: Large-scale Software Integration for Spoken Language and Multimodal Dialog Systems. Natural Language Engineering **10** (2004) Special issue on Software Architecture for Language Engineering.
20. Bontcheva, K., Tablan, V., Maynard, D., Cunningham, H.: Evolving GATE to Meet New Challenges in Language Engineering. Natural Language Engineering **10** (2004) Special issue on Software Architecture for Language Engineering.
21. Reithinger, N., Sonntag, D.: An integration framework for a mobile multimodal dialogue system accessing the semantic web. In: Proc. of Interspeech'05, Lisbon, Portugal (2005)
22. Oberle, D., Ankolekar, A., Hitzler, P., Cimiano, P., Sintek, M., Kiesel, M., Mougouie, B., Vembu, S., Baumann, S., Romanelli, M., Buitelaar, P., Engel, R., Sonntag, D., Reithinger, N., Loos, B., Porzel, R., Zorn, H.P., Micelli, V., Schmidt, C., Weiten, M., Burkhardt, F., Zhou, J.: Dolce ergo sumo: On foundational and domain models in swinto (smartweb integrated ontology). Technical report, AIFB, Karlsruhe (July 2006)
23. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schcneider, L.: Sweetening Ontologies with DOLCE. In: In 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02). Volume 2473 of Lecture Notes in Computer Science., Sigünza, Spain (Oct. 1–4 2002) 166 ff
24. Niles, I., Pease, A.: Towards a Standard Upper Ontology. In Welty, C., Smith, B., eds.: Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine (October 17–19 2001)
25. Cimiano, P., Eberhart, A., Hitzler, P., Oberle, D., Staab, S., Studer, R.: The smartweb foundational ontology. Technical report, (AIFB), University of Karlsruhe, Karlsruhe, Germany (2004) SmartWeb Project.
26. Oberle, D.: Semantic Management of Middleware. Volume I of The Semantic Web and Beyond. Springer (2006)
27. Gangemi, A., Mika, P.: Understanding the semantic web through descriptions and situations. In: Databases and Applications of Semantics (ODBASE 2003), Catania, Italy (November 3–7 2003)
28. Sonntag, D.: Towards interaction ontologies for mobile devices accessing the semantic web - pattern languages for open domain information providing multimodal dialogue systems. In: Proceedings of the workshop on Artificial Intelligence in Mobile Systems (AIMS). 2005 at MobileHCI, Salzburg (2005)
29. Hovy, E., Gerber, L., Hermjakob, U., Lin, C.Y., Ravichandran, D.: Towards semantic-based answer pinpointing. In: Proceedings of Human Language Technologies Conference, San Diego CA. (March 2001) 339–345

30. Sonntag, D., Romanelli, M.: A multimodal result ontology for integrated semantic web dialogue applications. In: Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006), Genova, Italy (May 24–26 2006)
31. Hunter, J.: Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology. In: Proceedings of the International Semantic Web Working Symposium (SWWS). (2001)
32. Benitez, A.B., Rising, H., Jorgensen, C., Leonardi, R., Bugatti, A., Hasida, K., Mehrotra, R., Tekalp, A.M., Ekin, A., Walker, T.: Semantics of Multimedia in MPEG-7. In: IEEE International Conference on Image Processing (ICIP). (2002)
33. Romanelli, M., Sonntag, D., Reithinger, N.: Connecting foundational ontologies with mpeg-7 ontologies for multimodal qa. In: Proceedings of the 1st International Conference on Semantics and digital Media Technology (SAMT), Athens, Greece (December 6-8 2006)
34. Ghallab, M., Nau, D., Traverso, P.: Automated planning. Elsevier Kaufmann, Amsterdam (2004)
35. Ankolekar, A., Hitzler, P., Lewen, H., Oberle, D., Studer, R.: Integrating semantic web services for mobile access. In: Proceedings of 3rd European Semantic Web Conference (ESWC 2006). (2006)
36. Engel, R.: Robust and efficient semantic parsing of free word order languages in spoken dialogue systems. In: Proceedings of 9th Conference on Speech Communication and technology, Lisboa (2005)
37. Gavaldà, M.: SOUP: A parser for real-world spontaneous speech. In: Proc. of 6th IWPT, Trento, Italy (February 2000)
38. Potamianos, A., Ammicht, E., Kuo, H.K.J.: Dialogue managment in the bell labs communicator system. In: Proc. of 6th ICSLP, Beijing, China (2000)
39. Ward, W.: Understanding spontaneous speech: the Phoenix system. In: Proc. of ICASSP-91. (1991)
40. Kaiser, E.C., Johnston, M., Heeman, P.A.: PROFER: Predictive, robust finite-state parsing for spoken language. In: Proc. of ICASSP-99. Volume 2., Phoenix, Arizona (1999) 629–632
41. Lavie, A.: GLR*: A robust parser for spontaneously spoken language. In: Proc. of ESSLLI-96 Workshop on Robust Parsing. (1996)
42. Huynh, D.T.: Communicative grammars: The complexity of uniform word problems. Information and Control **57**(1) (1983) 21–39
43. Becker, T.: Natural language generation with fully specified templates. In Wahlster, W., ed.: SmartKom: Foundations of Multi-modal Dialogue Systems. Springer, Heidelberg (2006) 401–410
44. Engel, R.: Spin: A semantic parser for spoken dialog systems. In: Proceedings of the 5th Slovenian First International Language Technology Conference (IS-LTC 2006). (2006)
45. Pfleger, N.: Fade - an integrated approach to multimodal fusion and discourse processing. In: Proceedings of the Dotoral Spotlight at ICMI 2005, Trento, Italy (2005)
46. Pfleger, N., Alexandersson, J.: Towards Resolving Referring Expressions by Implicitly Activated Referents in Practical Dialogue Systems. In: Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue (Brandial), Postdam, Germany (September 11-13 2006) 2–9
47. Porzel, R., Zorn, H.P., Loos, B., Malaka, R.: Towards a Separation of Pragmatic Knowledge and Contextual Information. In: Proceedings of ECAI 06 Workshop on Contexts and Ontologies, Riva del Garda, Italy (2006)

48. Hacker, C., Batliner, A., Nöth, E.: Are You Looking at Me, are You Talking with Me – Multimodal Classification of the Focus of Attention. In Sojka, P., Kopecek, I., Pala, K., eds.: Text, Speech and Dialogue. 9th International Conference (TSD 2006). Number 4188 in Lecture Notes in Artificial Intelligence (LNAI), Heidelberg, Germany, Springer (2006) 581–588
49. Matheson, C., Poesio, M., Traum, D.: Modelling grounding and discourse obligations using update rules. In: Proceedings of NAACL 2000. (May 2000)
50. Sonntag, D.: Towards combining finite-state, ontologies, and data driven approaches to dialogue management for multimodal question answering. In: Proceedings of the 5th Slovenian First International Language Technology Conference (IS-LTC 2006). (2006)
51. Carpenter, B.: The logic of typed feature structures (1992)
52. Larsson, S., Traum, D.: Information state and dialogue management in the TRINDI dialogue move engine toolkit. Natural Language Engineering, Cambridge University Press (2000)
53. Matheson, C., Poesio, M., Traum, D.: Modelling grounding and discourse obligations using update rules. In: Proceedings of NAACL 2000. (May 2000)
54. Walker, M., Fromer, J., Narayanan, S.: Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email (1998)
55. Singh, S., Litman, D., Kearns, M., Walker, M.: Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. Journal of Artificial Intelligence Research (JAIR), Volume 16, pages 105-133 (2002)
56. Rieser, V., Kruijff-Korbayova, K., Lemon, O.: A framework for learning multimodal clarification strategies. In: Proceedings of the International Conference on Multimodal Interfaces (ICMI). (2005)
57. Raghavan, H., Madani, O., Jones, R.: When will a human in the loop accelerate learning. In: Proceedings of the International Workshop on Artificial Intelligence for Human Computing at the International Joint Conference on Artificial Intelligence IJCAI'07, Hyderabad, India (2007) 97–105