Ends-based Dialogue Processing*

Jan Alexandersson, Tilman Becker, Ralf Engel, Markus Löckelt,
Elsa Pecourt, **Peter Poller**, **Norbert Pfleger** and **Norbert Reithinger**
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken
{janal,becker,engel,loeckelt,pecourt,poller,pfleger,bert}@dfki.de

Abstract

We describe a reusable and scalable dialogue toolbox and its application in multiple systems. Our main claim is that ends-based representation and processing throughout the complete dialogue backbone it essential to our approach.

1 Introduction

In the last couple of years our group at DFKI in Saarbrücken has been involved in a number of projects aiming at interfacing different devices in an intelligent way. The main goal of these projects has been to build functioning robust systems with which it is natural to communicate (not only for some few examples phrases). During the projects we have developed a dialogue toolbox consisting of a number of modules. By combining these modules in different ways we are able to realize a number of different types of dialogues, e. g., information seeking/browsing, device control, multi/cross-application and agent-mediated interactions for a number of (diverse) applications and systems. The full-blown combination of all modules form our *dialogue backbone* capable of engaging in multimodal man–machine communication.

In this paper, we discuss some of the design decisions taken along the road as well as lessons learned during the projects. Based on our experiences, we argue that *ends-based processing* is vital to the success of our approach. We strive for a balance between complex theories and pragmatic decisions. Of secondary interest is the implementation of theories capable of processing linguistically exotic phenomena in favor of ends-based processing in all modules of the toolbox. Hence it is more important to reach the representation rather than how we get there.

An ontology is often – as we understand it – a good ends-based representation but we can do without it. In the MIAMM project (see section 2) we use no ontology but instead an event based representation. Whatever representation we do choose, we would like to stress the importance of a consequent principle-based design of the representation and the fact that the complete backbone uses it. Exactly this guarantees, e. g., the scalability of our approach.

The paper is organized as follows: the next section provides an overview of projects and systems central to the development of our toolbox. Section 3 describes most of its modules. Before we conclude the paper, we provide a list of claims and lessons learned in section 4.

2 A Number of Projects

In this paper, we describe a toolbox which we can customize according to the projects needs. Using this toolbox we have implemented a number of systems, all having different requirements, needs and ends. They range from (monomodal) typed input/output as in the NaRATo project to multimodal agent-mediated communication as in SmartKom. Below we describe the different projects and systems showing that we are able to cover several kinds of communication paradigm.

SmartKom

SMARTKOM is a mixed-initiative dialogue system that provides full symmetric multimodality by combining speech, gesture, and facial expressions for both user input and system output (Wahlster, 2003). It provides an anthropomorphic and affective user interface through its personification of an embodied conversational agent, called Smartakus. The interaction metaphor is based on the so-called *situated, delegation-oriented dialogue paradigm*: the user delegates a task to a virtual communication assistant which is visualized as a life-like character. The interface agent recognizes the user's intentions and goals, asks the user for feedback if necessary, accesses the various services on behalf of the user, and presents the results in an adequate manner. Non-verbal reactions of the users are extracted from their facial expression or the prosodic features and affect subsequent

*The research presented here is funded by the German Ministry of Research and Technology under grant 01 IL 905, the European Union under the grants IST-2000-29487 and IST-2001-32311 and IDS-Scheer AG.

system presentations.

As it is depicted in Figure 1, SMARTKOM realizes a flexible and adaptive shell for multimodal dialogues and addresses three different application scenarios:

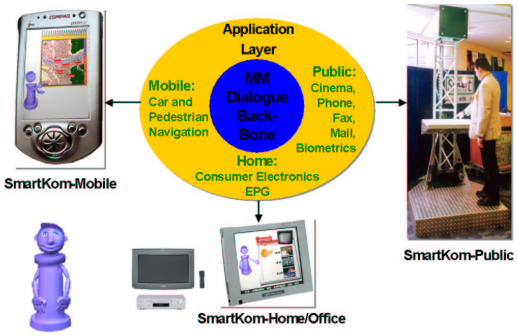

Figure 1: SMARTKOM's dialogue backbone and application scenarios

SMARTKOM PUBLIC realizes an advanced multimodal information and communication kiosk for, e. g., shopping malls. The user can get information about movies, reserve seats in a theater, and communicate using telephone, fax, or electronic mail. Before the system grants access to personal data, e. g., an address book, the user has to authenticate himself using either hand contour recognition, signature or voice verification. SMARTKOM HOME serves as a multimodal infotainment companion for the home theater. A portable web-pad acts as an advanced remote control where the user gets programming information from an electronic program guide service and easily controls consumer electronics devices like a TV set or a VCR. Similar to the kiosk application, the user may also use communication services at home. SMARTKOM MOBILE realizes a mobile travel companion for navigation and location-based services. It uses a PDA as a front end which can be added to a car navigation system. This system offers services like integrated trip planning and incremental route guidance. In the mobile scenario speech input can be combined with pen-based pointing.

All functionalities, modality combinations and technical realizations including a wide variety of hardware options for the periphery are addressed by the same core dialogue system with common shared knowledge sources. The processing relies on a knowledge based, configurable approach: we provide general solutions based on declarative knowledge sources in favour for special solutions and/or shortcuts or application specific procedural processing steps within the dialogue core of the system. The interaction processing is based on M3L (**M**ulti**m**odal **M**arkup **L**anguage), a complete XML language designed in the context of SMARTKOM that covers all data interfaces within the complex multimodal dialogue system (Gurevych et al., 2003a). The technical realization is based on the MULTIPLATFORM testbed (Herzog et al., 2004), an integration platform that provides a distributed component architecture. MULTIPLATFORM is implemented on the basis of the scalable and efficient publish/subscribe approach that decouples data producers and data consumers. Software modules communicate via so-called data pools that correspond to named message queues. Every data pool can be linked to an individual data type specification in order to define admissible message contents.

MIAMM

The main objective of the MIAMM project is to develop new concepts and techniques in the field of multimodal interaction to allow fast and natural access to large multimedia databases (Reithinger et al., 2003b). This implies both the integration of available technologies in the domain of speech interaction (Natural Language Understanding — SPIN — see section 3.1) and interaction management (Action Planner — AP — see section 3.3) and the design of novel technology for haptic designation and manipulation coupled with an adequate visualization. The envisioned end-user device is a hand-held PDA that provides an interface to a music database. The device includes three force-feedback buttons on the left side and one wheel on the upper right side (see figure 2). The buttons allow navigation through the visualized data, and performing of various actions on the presented objects (e. g., select, play).

The MIAMM architecture follows the "standard" architecture of interactive systems, with the consecutive steps mode analysis, mode coordination, interaction management, presentation planning, and mode design. To cope with artefacts arising from processing time requirements and coordination of different processes, this architecture was modified, so that only events that are relevant to other modules are sent, whereas the others remain internal. Thus, haptic interaction is decoupled from the more time-consuming speech processes, and only sends feedback when it is needed for the resolution of underspecified structures or when the interaction involves external actions, e. g., playing a selected track. The system consists of two modules for natural language input processing, namely recognition and interpretation. On the output side an MP3 player is used to play the songs and the pre-recorded speech prompts to provide acoustic feedback. The visual-haptic-tactile module is responsible for the selection of the visualization, and for the assignment of haptic features to the force-feedback buttons. The visualization module renders the graphic output and interprets the force imposed by the user to the haptic buttons. The dialogue manager consists of two main blocks,

Figure 2: The force-feedback device developed in the MIAMM project. The display shows a view of a database using a timeline.

the multimodal FUSION (see section 3.2) which is responsible for the resolution of multimodal references using the contextual information hold in the dialogue history, and the AP, that interprets the user intention and triggers a suitable system response. The AP is connected via a domain model to the multimedia database. The domain model uses an inference engine that facilitates access to the database.

The integration environment is based on the Simple Object Access Protocol (SOAP) (see `www.w3.org/TR/SOAP`). The communication between the modules is based on the multimodal interface language (MMIL). This specification accounts for the incremental integration of multimodal data to achieve a full understanding of the multimodal acts within the system. It is flexible enough to handle the various types of information processed and generated by the different modules.

COMIC

COMIC is an European IST 5th framework project focusing on new methods of work and e-commerce (den Os and Boves, 2003). Goal of this project is to develop a user centric, multimodal interface for a bathroom design tool which was developed by the COMIC partner ViSoft (see `www.visoft.de`). The implementation work is accompanied by research in the cognitive aspects of human-human and human-computer interaction.

Figure 3 shows a user interacting with the initial prototype of the system. The system enables the users to enter by speech and pen the blueprint of their bathroom including handwriting and drawing dimensions of walls, windows, and doors respectively. In a second step the user can browse and choose decoration and sanitary ware for the bathroom. Finally, the underlying application allows real-time, three-dimensional exploring of the modeled bathroom. System output includes the application itself and a realistically animated, speaking head.

Figure 3: Interaction with the COMIC system.

The architecture of the COMIC system again resembles the architecture of our core dialogue backbone. However, only SPIN, FUSION and Generation[1] are used for this project all other modules are provided by other partners. COMIC is also based on MULTIPLATFORM as the integration middleware, allowing a reuse of the module wrappers and engines. The representation of information is similar to that of SMARTKOM although the actual ontology differs in significant parts (e. g., no upper model). Hence the integration of SPIN and Generation was limited to the revision and adaption of the language and ontology dependent knowledge sources. FUSION, however, needed a deeper adaption as outlined in section 3.2.

Yet another (kind of) system

For the system *NaRATo* we have used parts of our toolbox – language understanding, discourse modeling , action planning, and generation – for a dialogue system interfacing the ARIS tool-set, a business process management system (see `www.ids-scheer.com`). The system uses typed input and output to provide access to a given process model stored in a database.

3 A Number of Modules

Our toolbox deploys a number of modules which are connected in a (nowadays) standard fashion (see figure 4). The input channels are fused by the modality fusion. This module is also responsible for resolving not just deictic expressions using gesture and speech but also referential expressions involving the dialogue context. The discourse module is the central repository for modality dependent and modality independent information. Here, the

[1]Generation in COMIC is actually only realization as the Fission module takes care of content selection and (most of) sentence planning.

user contribution is interpreted in context which involves resolving, e. g., a wide range of elliptical contributions. The action planner is the actual engine: using a regression planning approach the next system action is planned possibly preceeded by access of some external device. Finally, the presentation manager renders the system action. Here, the availability of different output modalities and the situation are influencing the realization of the action.

Our architecture differs from that of (Blaylock et al., 2003) in that the responsibility of the next system action is in our case purely decided by the action planner; the approach has some similarities with the one taken in (Larsson, 2002) in that most communicative actions represent request-response interactions along goals (akin to QUDs), and there is a notion of information state, which is however kept separated between the discourse modeler (for information specific to dialogue content, roughly equivalent to the SHARED information in IBiS) and the action planner (for other information, such as the agenda of the dialogue engine).

3.1 Natural Language Understanding

The task of the natural language understanding module is to transform the output of the speech recognizer into a list of possible user intentions which are already represented in the system-wide high-level ontology (see section 4). For this task a new template-based semantic parsing approach called SPIN (Engel, 2002) was developed at DFKI and is used in all aforementioned projects.

As typical for a semantic parser, the approach does not need a syntactic analysis, but the high level output structure is built up directly from word level. This is feasible since the input consists of spoken utterances intended to interact with a computer system and therefore, they are usually syntactically less complicated and limited in length. Furthermore, the lack of a syntactical analysis increases the robustness against speech recognition errors (speaker independent recognizers still have a word error rate of 10%-30%) and syntactically incorrect input by the user.

SPIN differs from other existing semantic parsing approaches by providing a more powerful rule language and a powerful built-in ontology formalism. The main motivation for the powerful rule language is to simplify the creation and maintenance of rules. As the amount of required rules is quite large (e.g., in the SmartKom project 435 templates are used), easy creation and maintenance of the rules is one of the most important issues for parsers in dialogue systems. Additionally, high-level output structures have to be generated and these output structures may be structurally quite different from the implied structure of the input utterance. A powerful rule language simplifies this task significantly.

Several off-line optimizations still provide fast processing despite the increased rule power. The most important off-line optimization is the computation of a fixed rule application order with the objective to avoid wasting time by the generation of sub-optimal results.

The powerful built-in ontology formalism helps to integrate the module in dialogue systems by only creating the knowledge bases and an interface layer but without any changes in the code base. Due to the lack of a standard ontology formalism for dialogue systems, each dialogue system uses a slightly different formalism. The powerful internal ontology formalism simplifies the task of mapping the system-wide ontology formalism to the internal one.

Current research will improve the approach in two areas. First, the time-consuming creation of the knowledge bases which has to be done completely manually up to now will be supported by machine learning techniques. Second, the external linguistic preprocessing of the speech recognizer output, like a syntactic analysis, will be possible without incorporating linguistic information into the knowledge bases. This would allow to process syntactically more complicated user utterances and still provides easy creation of the knowledge bases.

3.2 Modality Fusion

Multimodal dialogue systems like SmartKom or Comic give users the opportunity to express their needs not only by speech but also by different modalities, e. g., by gesturing or by using a pen. Furthermore, users can also combine several modalities to express one *multimodal* utterance (e. g., "I want to start here" accompanied by a pointing gesture towards a location on a map). As the recognizers and analyzers of the different modalities generate modality specific hypotheses, a component is needed to synchronize and integrate those monomodal hypotheses into multimodal ones. This module is called *FUSION*.

Based on human-human communication research, e. g., (Oviatt, 1999), we can identify four basic interaction patterns of how to use different modalities within a single multimodal utterance:

redundant the information provided by two modalities is basically the same,

concurrent two modalities are used one after another to provide information,

complementary the information provided by two modalities can be intertwined,

contradicting the information provided by one modality is contradictory to the information provided by the other modality.

All these interaction patterns can be resolved by obtaining access to information about the internal structure of objects. Especially when having to integrate information from one source into another, we need to know what specific objects look like, e. g., which sub-objects they

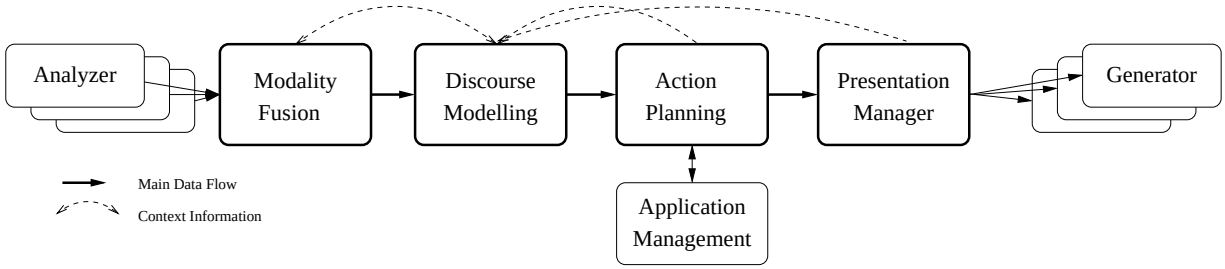

Figure 4: The architecture of the full blown version of our dialogue toolbox. Modality Fusion combines the different results from the analyzers; Discourse Modeling interprets in context; Action Planning determines the next system action; Presentation Management splits and coordinates the output on the different output modalities.

comprise. This information is typically provided by an ontology, e. g., via the type hierarchy and the slot definitions of each object. So, what FUSION must accomplish is to utilize processing strategies based on a type hierarchy and a given set of object definitions.

In SmartKom we applied a so called slot-filling approach for the integration of the two modalities speech and gesture. Multimodal hypotheses are compiled by inserting the hypotheses of the gestural modality into the hypotheses of the speech modality. The advantage of this approach is that apart from an ontology no further knowledge sources are required. This approach proved to be very fast and robust. However, the drawback is that an adaption to a different dialogue system or to new modalities is quite expensive.

With respect to our overall goal of building a scalable and reusable core dialogue system, we uncoupled the core FUSION system from the needs of the dialogue system, the available modalities, and processing strategies. Thus, we implemented a special purpose production rule system. Key to this approach is that all processing strategies are defined by production rules which can be easily created and adapted to the new surroundings and there are two powerful operations for accumulating information – unification and overlay (Alexandersson and Becker, 2003).

3.3 Action Planning

Task oriented cooperative dialogues, where participants collaborate to achieve a common goal, can be viewed as coherent sequences of utterances asking for actions to be performed or introducing new information to the dialogue context. The task of the action planner is to recognize the user's goal, to trigger required actions for its achievement, and to devise appropriate sub-dialogues and feedback. The actions can be internal, such as updating the internal state of the system, or external, like database queries, device operation or communication with the user. Thus, the action planner controls both the task and the interaction structure. Task and dialogue interactions are viewed as joint communicative games played with different agents, including the user and all modules that directly communicate with the action planner.[2] Participants are abstractly represented by communication channels transforming between the uniform internal representation of communicative moves to the data structures used by external participants. Each game is composed of a number of moves, defined by dialogue knowledge sources. The game definitions are similar to STRIPS plan operators. They specify sets of preconditions and effects, and additionally, for each move the channel through which the data flows, and data structures containing the semantic content of the move intention. The adoption of a dialogue goal triggers a planning process (non-linear regression planning, with hierarchical decomposition of subgoals) resulting in a series of communicative games to be played to achieve the goal. Move execution is then interleaved with checking their outcome, and possibly replanning if preconditions are violated. This strategy allows the system to deal with unexpected user inputs like misunderstandings or changing of goals.

The approach of planning with communicative games has two benefits with respect to the scalability of the system, one regarding communication channels, the other stemming from the use of small dialogue game units.

It is possible to integrate support for any number of additional devices to an already existing system by adding new communication channels (one Java class each); dialogue moves that do not use these channels will not be affected. Still, dialogue specifications for newly added devices can make use of the already defined ones.

As described above, the dialogue behavior is coded in terms of communicative games consisting of dialogue moves. For predetermined sequences of moves (e. g., a

[2] We use the term "communicative games" in addition to "dialogue games," since our dialogue model also includes communication interaction with applications and devices, such as database requests and answers, in terms of game moves.

fixed protocol for sending fax messages: (1) scan document, (2) convert to fax format, (3) send it via fax application), the dialogue game can resemble a fixed script, like the pre-made plans used, e. g., by (Larsson, 2002)), but in general, games specify atomic steps like single request-response subdialogues. To devise the course of action, a plan is then constructed dynamically as a game sequence. This has the advantage that (1) the plan can be flexibly adapted to changed circumstances, e. g., if a step becomes obsolete or is addressed early, and (2) games can be shared and reused as building blocks for other applications. So, when new functionality is integrated, the plan knowledge source will stay reasonably small– growing linearly in the number of games, not exponentially with the possible recipes.[3]

3.4 Discourse Modeling

The main objective of the discourse modeler (henceforth DIM) is to incorporate information stemming from the previous discourse context into the current intention hypotheses produced by the analysis modules. This objective decomposes into two main tasks which are on the one hand enhancing a hypothesis with compatible background information and estimating how well it fits the previous discourse context – what we call *enrichment* and *validation* – and on the other hand the resolution of referring expressions.

Discourse processing in the framework of a multimodal dialogue system has to deal with an extended set of input and output devices. Gestures, for example, accompanying speech not only support the resolution of referring expressions, in addition they change the discourse context. In general, the resolution of referring expressions within a multimodal approach requires access to a visual context representation. One key aspect of DIM is a unified context representation taking both the discourse and the visual context into account.

Our approach consists of a three-tiered discourse representation combined with a two layered focus handling, see (Pfleger et al., 2003). The actual processing is done by utilizing two operations: *unification* and *overlay* (Alexandersson and Becker, 2003). In combination with a scoring function (Pfleger et al., 2002), the latter is our main tool for enrichment and validation. Key to this approach is that DIM can be easily adapted to other dialogue systems with different tasks and demands. In that sense, the actual context representation is independent from the type of objects to be stored. Additionally, DIM can be used not only within a multimodal dialogue system but also within monomodal ones, as we showed in the NaRATo project.

[3]The usual downside is, the *planning* space is of course exponential. But as we use goal-directed search, only a small fraction of the possible plans is ever examined in practice.

3.5 Modality Fission

The modalities used in the SmartKom system are gesture, mimics, speech and also graphical presentations on devices of different sizes. The main task of multimodal fission is partitioning, i. e., dividing the presentation tasks into subtasks and generating an execution plan. A follow-up task is then the coordination and synchronization of related tasks, e. g., presentation of a graphical element with a pointing gesture and synchronization with speech.

The fission module is embedded in a presentation planner that also subsumes the graphical realization task. The module generates a full plan for graphics, gesture and mimics while the plan for speech is generated only on an abstract subtask level that is handed as input to the Text Generator (see next section).

The planning of a multimodal presentation consists of two parts: static gesture-sensitive graphical elements and a corresponding multimodal animation of the agent including gestures referring to objects with aligned audio-visual speech output. The first step performed on the input is a transformation into the internal input format of the core planning component PrePlan by applying an appropriate XSLT-stylesheet.

Then, the presentation planner starts the planning process by applying a set of presentation strategies which define how the facts are presented in the given scenario. Based on constraints, the strategies decompose the complex presentation goal into primitive tasks and at the same time they execute the media fission step depending on available modalities, which means they decide which part of the presentation should be instantiated as spoken output, graphics, or gestures of our presentation agent.

After planning the graphical presentation, appropriate speech and gesture presentations are generated. The gesture and speech form is chosen depending on the graphically shown information. I.e., if the graphically presented information is in the focus of a presentation, only a comment is generated in speech output. The goal of the gesture presentation is then to focus on the appropriate graphical element. If there is no graphically presentable information or it is insufficient, more speech is generated.

3.6 Natural Language Generator

The design of the Natural Language Generation (NLG) module is guided by the need to (i) adapt only knowledge sources when adding a new application and (ii) generalizing the knowledge sources from the applications.

Thus the NLG module is divided into an engine and declarative knowledge sources which are designed with the goal of capturing generalizations. The input to the NLG module are abstract presentation goals that are based on the ends-based presentation; the output is (annotated) text that typically is sent to a speech synthesizer. E.g., the NLG module in SmartKom uses syntactic struc-

ture and discourse information to supply richly annotated text for the Concept-To-Speech (CTS) approach.

On the one hand, the NLG module is templated–based (see also SPIN), skipping multiple layers of representation when mapping from the presentation goals. On the other hand, the templates are "fully specified" in the sense that they include intermediate layers of representation where possible to permit a later separation of rules into a multi-stage generation module. E.g., including syntax was also necessary for CTS, including semantics allows for the extraction of a realization module for COMIC. The template rules are based on the same PrePlan planning component used in fission. At least since (Reiter, 1995) the use of templates and "deep representations" is not seen as a contradiction. Picking up on this idea, the generation component in SmartKom is based on fully lexicalized generation (Becker, 1998), using partial derivation trees of a Tree-Adjoining Grammar (TAG). Right from the beginning of development, derivation trees which are seen as reflecting syntactic dependencies have been an explicitly represented layer in the template rules. Thus the higher level planning rules decide content selection, sentence plans and lexicalization, leaving syntactic realization to a TAG-based second step.

During development, we have enriched the syntactic trees with nested feature structures and have just finished a transformation of the phrasal templates to a fully lexicalized TAG, where every lexical item has its unique tree.

4 Ends-Based Processing

One of the most important constraints when building a functioning system has been the domain of the application. Based on the domain we developed ends-based representations which have so far mostly been ontologies or ontology-like structures, e. g., (Gurevych et al., 2003b) but which in fact could be event-based representations as well. How interpretation and presentation are connected to the abstract representation is of secondary interest; Our backbone uses this task-oriented representation for communication and processing and the way there and back may exclude, for instance, traditional semantics. We make two important observations: on the one hand, that the complete backbone should use a single representation, so that translations between different representations are avoided. Important here is that each module (ideally) separates its engine from its knowledge base. On the other hand, the common representation has to be ends-based and fulfil the needs of the application.

The latter point leads us to another lesson learned: The application has to be examined and its needs have to be mirrored in the representation. We also have to determine what interactions we are aiming for. Since, e. g., in SmartKom, we pursue a situated delegation-oriented dialog paradigm – meaning that the system is in itself not a dialogue partner as in (Blaylock et al., 2003) but instead the dialogue is mediated by an animated agent – we encapsulate the details of the application APIs in an application manager and hence provide a user-oriented view of the application(s). Additionally, the dialogue plans are represented separately from the ends-based representation in a different knowledge base, i. e., the plan specifications for the action planner. However, the plans refer to the application using the ends-based representation.

We have acquired our knowledge, e. g., ends-based represenations or interpretation rules completely by hand. While we avoid the potentially costly resources for the collection and annotation of corpora for automated learning[4], the question remains whether expanding knowledge sources by hand is feasible. Our approach has indeed allowed for scaling up – in SmartKom we have extended the system to more than 50 functionalities overall (Reithinger et al., 2003a).

In the following, we list the most important lessons we learned, which is by no means exhaustive:

Encapsulation Encapsulate the backbone from the application(s). This was one of the main lessons from the NaRATo and the SmartKom projects. We did not do it in the NaRATo project and spent lots of time interfacing the database. In SmartKom, such a module exists, and the backbone developers could concentrate on more relevant tasks in dialogue processing proper.

Representation Use one representation throughout the backbone. It is a secondary question how exactly it is done, but it is essential that you get there and avoid spending time on converting between different formalisms.

Representation (revisited) There is to be no presentation (system output) without representation on the ends-based representation level. This representation is part of the global dialogue history residing in the discourse module and can be accessed by any module, e. g., for reference resolution at any time during the course of the dialogue.

Interface In the case of a multi-module approach, use one well-defined representation for module communication. In most cases we have used XML and XML Schema which is convenient because a wide variety of infrastructure and tools is available. For instance, most XML processing tools allow for syntactic validation. However, XML is not mandatory. A final remark here: using XML in combination with stylesheets, we can in fact – contrary to the advice in Representation (above) – translate or convert messages to some internal representation easily.

Interface (revisited) Interfaces should be clean and well-defined. One reason for the success of the SmartKom project was the requirement to define every interface formally by XML Schema. These XML Schemata were kept in a project-wide repository and changed at this one place

[4] Supervised as well as unsupervised

after mutual agreement only. Due to the multi-blackboard approach, there are not point-to-point connections, but n-to-m connections, and an interface definition comprises of a precise description of what is supposed to be an allowed message for a specific blackboard.

Integration Our large projects have profited enormously of a dedicated integration group providing infrastructure, integration cycles and – for, e. g., the SmartKom and COMIC systems – a testbed (Herzog et al., 2004).

Multimodality More modalities allow for more natural communication, which normally employs multiple channels of expression, suited to the content to be communicated. For natural language processing per se this raises new and interesting challenges, e. g., cross-modal referential expressions. It is also the case that more modalities constrain interpretation and hence enhance robustness. The ends-based representation allow for modality-independent processing in the backbone.

Standards Standards ease scalability. For, e. g., ends-based representations and tools, we have previously developed custom-built software providing short-lived solutions. In other situations we have chosen standards and standard tools. We claim that the latter is beneficial in at least two ways: It opens up the door for scalability since we can re-use our as well as other's resources. Secondly it is easier to maintain our solution over time and projects.

5 Conclusion

DFKI's dialogue toolbox was used in a number of fully functional, differently sized systems with a variety of interaction paradigms. Vital to its success in terms of reusability and scalability was the choice of a modular design and ends-based representations throughout the complete backbone. Starting from basic functionalities, it is possible to extend the system coverage while incorporating new features. Future work includes reusing (parts of) the backbone in EU and nationally funded large projects like AMI, TALK, Inscape, VirtualHuman and SmartWeb.

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.

Tilman Becker. 1998. Fully lexicalized head-driven syntactic generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, Niagara-on-the-Lake, Ontario, Canada, August.

Nate Blaylock, James Allen, and George Ferguson. 2003. Managing communicative intentions with collaborative problem solving. In Ronnie W. Smith and Jan van Kuppevelt, editors, *Current and New Directions in Discourse and Dialogue*. Kluwer.

Els den Os and Lou Boves. 2003. Towards ambient intelligence: Multimodal computers that understand our intentions. In *eChallenges e-2003*, pages 22–24.

Ralf Engel. 2002. SPIN: Language understanding for spoken dialogue systems using a production system approach. In *Proceedings of 7th International Conference on Spoken Language Processing (ICSLP-2002)*, pages 2717–2720, Denver, Colorado, USA.

Iryna Gurevych, Robert Porzel, Elena Slinko, Norbert Pfleger, Jan Alexandersson, and Stefan Merten. 2003a. Less is more: Using a single knowledge representation in dialogue systems. In *Proceedings of the HLT-NAACL Workshop on Text Meaning*, pages 14–21, Edmonton, Canada, May.

Iryna Gurevych, Robert Porzel, Hans-Peter Zorn, and Rainer Malaka. 2003b. Semantic coherence scoring using an ontology. In *Proceedings of the Human Language Technology Conference - HLT-NAACL 2003*, Edmonton, CA, May, 27– June, 1.

Gerd Herzog, Heinz Kirchmann, Stefan Merten, Alassane Ndiaye, Peter Poller, and Tilman Becker. 2004. Large-scale software integration for spoken language and multimodal dialog systems. *Journal of Natural Language Engineering*. To appear in the special issue on "Software Architecture for Language Engineering".

Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Göteborg University.

Sharon Oviatt. 1999. Ten myths of multimodal interaction. *Communications of the ACM*, 42(11):74–81.

Norbert Pfleger, Jan Alexandersson, and Tilman Becker. 2002. Scoring functions for overlay and their application in discourse processing. In *KONVENS-02*, Saarbrücken, September – October.

Norbert Pfleger, Ralf Engel, and Jan Alexandersson. 2003. Robust multimodal discourse processing. In Kruijff-Korbayova and Kosny, editors, *Proceedings of Diabruck: 7th Workshop on the Semantics and Pragmatics of Dialogue*, Wallerfangen, Germany, September.

Ehud Reiter. 1995. NLG vs. templates. In *5th European Workshop in Natural Language Generation*, pages 95–105, Leiden, May.

Norbert Reithinger, Jan Alexandersson, Tilman Becker, Anselm Blocher, Ralf Engel, Markus Löeckelt, Jochen Müeller, Norbert Pfleger, Peter Poller, Michael Streit, and Valentin Tschernomas. 2003a. Smartkom - adaptive and flexible multimodal access to multiple applications. In *Proceedings of ICMI 2003*, Vancouver, B.C.

Norbert Reithinger, Dirk Fedeler, Ashwani Kumar, Christoph Lauer, Elsa Pecourt, and Laurent Romary. 2003b. MIAMM - A Multimodal Dialogue System Using Haptics. In Jan van Kuppevelt, Laila Dybkjaer, and Niels Ole Bersen, editors, *Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*. Kluwer Academic Publishers.

Wolfgang Wahlster. 2003. Smartkom: Symmetric multimodality in an adaptive and reusable dialogue shell. In R. Krahl and D. Gnther, editors, *Proceedings of the Human Computer Interaction Status Conference 2003*, pages 47–62, Berlin: DLR, June.