IJCAI 2011 Workshop

The 7th IJCAI Workshop on

Knowledge and Reasoning in Practical Dialogue Systems

Proceedings

17 July 2011
Barcelona, Spain

Introduction

This is the seventh IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems. The first workshop was organised at IJCAI-99 in Stockholm, the second workshop took place at IJCAI-2001 in Seattle, and the third workshop was held at IJCAI-2003 in Acapulco. The the fourth workshop was held at IJCAI-2005 at Edinburgh. The fifth workshop was held in Hyderabad, India, 2007 and focused on dialogue systems for robots and virtual humans. The sixth workshop was held in Pasadena, CA in 2009, and focussed on challenges of novel applications of practical dialogue systems.

The seventh IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems focuses on challenges arising when implementing (conversational) dialogue systems for different types of users, such as elderly people and people with special needs.

Topics addressed in the workshop include:

- How can we evaluate dialogue systems or conversational dialogue systems for different types of people, e.g., people with special needs and young people?
- How can we implement dialogue systems in such a way that the target users can also interact with their surroundings?
- How can authoring tools for dialogue systems be developed such that application designers who are not experts in natural language can make use of these systems?
- What are the best ways of representing language resources for dialogue systems.
- What is the role of ontologies in dialogue systems?
- How can one easily adapt a dialogue system to a new application or user?
- What methods are best suited for design and development of dialogue systems?
- What are the most appropriate ways to evaluate dialogue systems for different types of users: what to evaluate and how. How do these systems differ from generic systems?

The workshop contains a collection of 8 papers divided in four chategories: Architecture, Learning Dialogue, Building and Evaluation of Dialogue Systems and, finally, a section we have called Limitations.

An event like this one always need help from additional people. We would particularly like to thank the Program Committee for helping us out and Adele Howe for help and guidance. Finally, we wish all participants of the Workshop a great event.

June 2011

Jan Alexandersson

David Traum

Arne Jönsson

Ingrid Zukerman

Organizers:

Jan Alexandersson, DFKI GmbH, Germany (Chair)
Arne Jönsson, Linkping University, Sweden (Co-chair)
David Traum ICT, USA (Co-Chair)
Ingrid Zukerman, Monash University, Australia (Co-Chair)

Program Committee:

Dan Bohus (USA)
Johan Bos (The Netherlands)
Sandra Carberry (USA)
Maxine Eskenazi (USA)
Kallirroi Georgila (USA)
Joakim Gustafson (Sweden)
Nancy Green (USA)
Phil Green (UK)
Kazunori Komatani (Japan)
Peter Ljunglöv (Sweden)
Kathleen McCoy (USA)
Wolfgang Minker (Germany)
Mikio Nakano (Japan)
Antti Oulasvirta (Finland)
Olivier Pietquin (France)
Ehud Reiter (UK)
Norbert Reithinger (Germany)
Amanda Stent (USA)
Jason Williams (USA)

Table of Contents

Limits of Simple Dialogue Acts for Tactical Questioning Dialogues
Ron Artstein, Michael Rushforth, Sudeep Gandhe, David Traum and Aram Donigian 1

Learning Dialogue Agents with Bayesian Relational State Representations
Heriberto Cuayáhuitl 9

Building Modular Knowledge Bases for Conversational Agents
Daniel Macias-Galindo, Lawrence Cavedon and John Thangarajah 16

SceneMaker: Visual Authoring of Dialogue Processes
Gregor Mehlmann, Patrick Gebhard, Birgit Endrass and Elisabeth Andre 24

Rapid Development of Multimodal Dialogue Applications with Semantic Models
Robert Neßelrath and Daniel Porta 37

Unsupervised Clustering of Probability Distributions of Semantic Frame Graphs for POMDP-based Spoken Dialogue Systems with Summary Space
Florian Pinault and Fabrice Lefèvre 48

Subjective and Objective Evaluation of Conversational Agents
Annika Silvervarg and Arne Jönsson 54

Speaking and Pointing—from Simulations to the Laboratory
Ingrid Zukerman, Arun Mani, Zhi Li and Ray Jarvis 64

Workshop Program - KRPDS11

Sunday, July 17, 2011

10–10:20	Opening remarks
	Session 1: Architecture
10:20–11	*SceneMaker: Visual Authoring of Dialogue Processes* Gregor Mehlmann, Patrick Gebhard, Birgit Endrass and Elisabeth Andre
11–11:30	Morning coffee break
	Session 2: Learning Dialogue
11:30–12	*Unsupervised clustering of probability distributions of semantic frame graphs for POMDP-based spoken dialogue systems with summary space* Florian Pinault and Fabrice Lefvre
12-12:30	*Learning Dialogue Agents with Bayesian Relational State Representations* Heriberto Cuayhuitl
12:30–14:30	Lunch
	Session 3: Building and evaluating dialogue systems
14.30–15:10	*Building Modular Knowledge Bases for Conversational Agents* Daniel Macias-Galindo, Lawrence Cavedon and John Thangarajah
15:10-15:50	*Rapid Development of Multimodal Dialogue Applications with Semantic Models* Robert Neßelrath and Daniel Porta
15:50-16:30	*Subjective and Objective Evaluation of Conversational Agents* Annika Silvervarg and Arne Jönsson
16:30–17	Afternoon coffee break
	Session 4: Limitations
17–17:30	*Speaking and Pointing – from Simulations to the Laboratory* Ingrid Zukerman, Arun Mani, Zhi Li and Ray Jarvis
17:30–18:10	*Limits of Simple Dialogue Acts for Tactical Questioning Dialogues* Ron Artstein, Michael Rushforth, Sudeep Gandhe, David Traum and Aram Donigian
18:10–18:30	General discussion

Limits of Simple Dialogue Acts for Tactical Questioning Dialogues

Ron Artstein and **Michael Rushforth*** and **Sudeep Gandhe** and **David Traum**
Institute for Creative Technologies, University of Southern California
12015 Waterfront Drive, Playa Vista, CA 90094-2536, USA

MAJ Aram Donigian
United States Military Academy
West Point, NY 10996, USA

Abstract

A set of dialogue acts, generated automatically by applying a dialogue act scheme to a domain representation designed for easy scenario authoring, covers approximately 72%–76% of user utterances spoken in live interaction with a tactical questioning simulation trainer. The domain is represented as facts of the form ⟨object, attribute, value⟩ and conversational actions of the form ⟨character, action⟩. User utterances from the corpus that fall outside the scope of the scheme include questions about temporal relations, relations between facts and relations between objects, questions about reason and evidence, assertions by the user, conditional offers, attempts to set the topic of conversation, and compound utterances. These utterance types constitute the limits of the simple dialogue act scheme.

Introduction

In previous work, we presented a spoken dialogue system for tactical questioning simulation which uses a simple scheme of dialogue acts, designed to facilitate authoring by domain experts with little experience with dialogue systems (Gandhe et al. 2009). The dialogue acts are generated automatically from a representation of facts as ⟨object, attribute, value⟩ triples and actions as ⟨character, action⟩ pairs. We found that initially the dialogue act scheme only covered about 50% of the user utterances, but our analysis showed that simple extensions could increase coverage to above 80% (Artstein et al. 2009). This paper puts that claim to test. We took a corpus of user utterances collected in interaction with the system, and mapped it to a set of dialogue acts in two stages: first we mapped half of the utterances to the original dialogue acts used in collecting the corpus, then we added facts to the domain representation in order to address gaps found in the coverage, and afterwards we mapped the held out data to dialogue acts derived from the expanded domain. The conclusion from this process is that the claim of Artstein et al. (2009) was about right – the expanded domain covers about 72–76% of the user utterances. While many of the remaining utterances could also be represented through an additional expansion of the domain, there remains a set of utterances which cannot be represented using the simple scheme. This paper presents a detailed analysis of those utterances that cannot be expected to be handled by the scheme, exploring the limits of this simple dialogue act representation.

Dialogue acts are often used as representations of the meaning of utterances in dialogue, both for detailed analyses of the semantics of human dialogue (e.g., Sinclair and Coulthard 1975; Allwood 1980; Bunt 1999) and for the inputs and outputs of dialogue reasoning in dialogue systems (e.g., Traum and Larsson 2003; Walker, Passonneau, and Boland 2001). There are many different taxonomies of dialogue acts, representing different requirements of the taxonomizer, both the kinds of meaning that is represented and used, as well as specifics of the dialogues and domain of interest (Traum 2000). There are often trade-offs made between detailed coverage and completeness, simplicity for design of domains, and reliability for both manual annotation and automated recognition. A common concern for theories of dialogue acts is representing the mechanisms that regulate the flow of conversation, which determine dialogue properties such as turn-taking, coordination among speakers and cohesiveness of the dialogue.

In our tactical questioning simulator, the scheme is intentionally kept very simple, in order to allow authoring by domain experts who work on the level of the domain representation, without detailed knowledge of dialogue act semantics and transitions (Gandhe et al. 2009). This simplicity results in limited expressibility. We found that in the specific genre of tactical questioning of a virtual character, most of the difficulties faced by the simple dialogue act scheme are not ones of regulating the conversation. Rather, it is the representation of information. The purpose of tactical questioning is to extract specific information through interview, and users consistently employ a richer view of the information than the system can represent. While the gap in coverage only affects a small fraction of user utterances, addressing it would require changes not only to the dialogue act scheme, but to the domain representation as well. This paper provides a characterization of the tactical questioning domain as it appears from an interviewer's perspective, based on an analysis of actual user utterances.

The remainder of the paper describes the tactical questioning genre of dialogue and the dialogue system architec-

*Now at the University of Texas at San Antonio

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 1–8,
Barcelona, July 2011

ture used in collecting the corpus; presents the corpus and the procedure for annotation and domain expansion; and presents the results of the annotation experiment, both in quantitative terms (reliability and coverage) as well as a detailed analysis of the gaps of the dialogue act representation.

Tactical Questioning

Artstein et al. (2009) provides an overview of the Tactical Questioning domain, which is defined as "the expedient, initial questioning of individuals to obtain information of immediate value" (U.S. Army 2006). A tactical questioning dialogue system is a simulation training environment where virtual characters play the role of a person being questioned; these characters display a range of behaviors such as answering questions cooperatively, refusing to answer questions, or intentionally providing incorrect answers (lying). The interviewer (human participant) may work to induce cooperation by building rapport with the character, addressing their concerns, making promises and offers, as well as threatening or intimidating the character.

System architecture

The architecture for our tactical questioning dialogue systems is a compromise between a text-to-text classifier that directly maps questions to responses in a stateless fashion (Leuski et al. 2006) and a full-fledged system with intricate reasoning and inference capabilities (Traum et al. 2008). It employs a fairly basic representation of dialogue acts, which are generated automatically from a simple domain representation. The generated dialogue acts reflect the role of the human participant as an interviewer and the character as a person being interviewed. We thus make a distinction between *user dialogue acts* and *character dialogue acts* – some dialogue act types are made by both user and character, but others are restricted to only one of the participants.

The dialogue acts are employed in conversation through a finite-state representation of local dialogue segments, a set of policies for engaging in the network, and a rule-based dialogue manager to update the context and choose dialogue acts to perform (Gandhe et al. 2008). This functionality allows for short subdialogues where the character can ask for and receive certain assurances (such as protection or confidentiality) and still remember the original question asked by the trainee. The link between dialogue acts and natural language is provided by a statistical classifier (Leuski and Traum 2008).

The domain representation encodes the character's knowledge as a set of facts of the form ⟨object, attribute, value⟩; in addition, the domain specifies a number of actions that the character and interviewer may perform, such as offers, threats, compliments and insults. Dialogue acts are automatically generated from the domain specification, by applying an illocutionary force (or *dialogue act type*) to a semantic content containing the relevant portion of the domain specification. For example, each fact generates 3 dialogue acts – a character dialogue act of type *assert*, a user dialogue act of type *yes/no question*, and a user dialogue act of type *wh-question* which is formed by abstracting over the value. Each object in the domain is considered a topic of conversation, and generates a set of grounding acts used for confirming the topic (*repeat-back* and *request-repair*). Additional dialogue act types include forward function (elicitation) and backward function (response) dialogue acts, as well as some generic dialogue acts that are defined independently of the domain such as greetings, closings, thanks, and special dialogue acts that are designed to handle out-of-domain dialogue acts from the user.

The system architecture was designed to facilitate rapid creation of characters by scenario designers who are experts in tactical questioning, but not experts in dialogue or dialogue systems (Gandhe et al. 2009). The architecture therefore hides much of the dialogue logic from the scenario designer, exposing only the domain and a limited set of policies. The simple structure of the domain representation is intended to provide a minimal amount of structure that would allow automatic creation of dialogue acts, while keeping authoring possible without extensive knowledge of ontologies. The representation is intended to capture just enough information about a user's actual utterance to allow for natural and believable dialogue behavior by the character.

Of course, users are not aware of the system's limited representations, and their models of the domain are richer than what is encoded. In a pilot study (Artstein et al. 2009) we found that the available dialogue acts adequately represented about 50% of the user utterances, and our analysis showed that with some modifications, coverage was expected to increase to 80% or above. The remaining ($< 20\%$) utterances could be dealt with using policies for unrecognized input (such as clarification requests or character initiative), which would result in a believable user experience that is useful for training.

This paper tests the claim of Artstein et al. (2009) using a corpus collected in live interaction with a virtual human with an expanded domain and dialogue act set. We found that coverage has indeed increased to 72–76%. However, there remains a substantial number of user utterances which cannot be represented using the dialogue act scheme, and we provide a detailed characterization of these utterance types.

Scenario details

The experiment reported in this paper used one specific scenario implemented in the dialogue system described above. This is the same scenario described in Artstein et al. (2009), with small modifications based on the results of that experiment. In this scenario, the user plays the role of a commander of a small military unit in Iraq whose unit had been attacked by sniper fire while on patrol near a shop owned by a person named Assad. The user interviews a character named Amani who lives near the shop, was a witness to the incident, and is thought to have some information about the identity of the attackers.

Amani's knowledge about the incident is represented as facts in the domain – triples of the form ⟨object, attribute, value⟩; each fact is either true or false (false facts are used by Amani when she wants to tell a lie). Table 1 gives some facts about the incident. For example, Amani knows that the name of the suspected sniper is Saif, and that he lives

Object	Attribute	Value	T/F
strange-man	name	saif	true
strange-man	name	unknown	false
strange-man	location	store	true
brother	name	mohammed	true

Table 1: Some facts about the incident

in the store. She can lie and say that she doesn't know the suspect's name. She does not have an available lie about the suspect's location (though she can always refuse to answer a question). The facts in the domain give rise to dialogue acts – for example, the fact ⟨strange-man, name, saif⟩ defines a character dialogue act with a meaning equivalent to "the suspect is named Saif" (*assert*), and two user dialogue acts, equivalent in meaning to "is the suspect named Saif?" (*yes/no question*) and "what is the suspect's name?" (*wh-question*).

Since our experiment is intended to check how well the dialogue act scheme represents user utterances, the remainder of the paper will be concerned only with the user dialogue acts generated by the scheme, not with the character dialogue acts or dialogue policies.

Method

We ran a pilot study at ICT, the results of which were reported in Artstein et al. (2009). Based on the pilot study we modified the domain, adding a few facts. We also made some changes to the dialogue act scheme, adding several dialogue act types that are generated from the domain. The character's policies were updated to handle the new dialogue act types.

Corpus collection

We collected a corpus of dialogues between human participants and the Amani character at the United States Military Academy at West Point. The dialogue participants were all cadets enrolled in a negotiation course; they had practiced negotiations in human-human role plays, but had never talked to a virtual character. Dialogue participants were given an instruction sheet with some information about the incident, the character, and suggestions for interaction, but no guidance about particular language to use with the character (see appendix). The character's behavior could be set to either confirm offers and topic shifts explicitly (high grounding) or not confirm them (low grounding). Each participant talked to the character twice (one interaction of each type), with the order of presentation balanced across participants; participants were not informed of the variation, and were instructed to treat the second dialogue as completely separate from the first. Since the current experiment focuses only on the user utterances and not the character behavior, we treat utterances from both conditions as a single corpus. The corpus consists of 68 dialogues (34 participants), comprising of a total of 1854 utterances; dialogue lengths vary from 8 to 46 utterances (mean 27.3, median 28.5, standard deviation 8.5).

Dialogue act annotation

Utterances were matched to fully specified user dialogue acts by 3 experienced annotators, including the first and second authors and a student annotator. The annotation guidelines were to match each user utterance to the most appropriate user dialogue act, and if no dialogue act was close enough, to match to "unknown". Based on the problems reported in Artstein et al. (2009), we added instructions to treat *Do you know* and *Can you tell* questions as wh-questions, and to treat formulaic greetings such as *How are you* and *It's nice to meet you* as greetings rather than questions or assertions.

Matching utterances to dialogue acts was done in two rounds. For the first round, the corpus was split in the following fashion. Whole dialogues were randomly selected until they totaled more than 100 utterances; this portion was annotated independently by all annotators and served as a reliability sample. The remaining dialogues were randomly assigned to annotators in a way that approximately balanced the number of utterances among the annotators. The annotators then matched utterances to dialogue acts from the system employed in collecting the corpus, using the domain creation tool (Gandhe et al. 2009), until about half of the corpus was annotated (annotators worked at different rates, so the number of utterances annotated at this stage was not balanced; see Table 5 below). The resulting annotated corpus will be referred to as the *original domain*, and it contains 768 unique utterances. Due to technical limitations, annotators mapped each utterance text to a single dialogue act, not taking into account context that would disambiguate different dialogue acts for the same text appearing at different times.

Based on the annotation of the original domain, we expanded Amani's domain to include meaning representations for most of the user questions that were not successfully mapped to dialogue acts. This resulted in a doubling of the number of available dialogue acts for interpretation (Table 2). The bulk of the expansion occurred in the representation of user questions through the addition of domain knowledge: each addition of a full ⟨object, attribute, value⟩ triple generated a *wh-question* and a *yes/no question*, while an addition of ⟨object, attribute⟩ without a value generated only a *wh-question* (the latter are questions that Amani can understand but does not know an answer to; such tuples were added in order to expand coverage of the user questions without adding knowledge to the character). In the course of adding domain knowledge, six new objects were created, and thus there were corresponding increases in grounding dialogue acts (*repeat-back* and *request-repair*). The *response* category includes responses to certain acts such as compliments, apologies and thanks; the increase in responses comes from the addition of compliments by Amani. No changes were made to the dialogue act scheme, that is to the rules that generate individual dialogue acts from the domain.

After expanding the domain, we took the remaining (unannotated) utterances and split them among the annotators using a similar method to the first round, creating a reliability sample of just over 100 utterances and splitting the

Dialogue Act Type	Pilot	Original	Expanded
generic acts[a]	10	13	13
closing	3	3	3
compliment	3	1	2
insult	2	2	2
offer	3	3	3
pre_closing	3	3	3
repeat_back	10	9	15
request_repair_attribute		9	15
request_repair_object	10	9	15
response	3	6	12
wh-question	**31**	**42**	**119**
yes/no question	**35**	**43**	**85**
Total	113	143	287

[a]One each of accept, ack, apology, greeting, offtopic, refuse_answer, reject, request_repair, thanks, and unknown; the original and expanded domains added clarify_elicit_offer, yes, and no.

Table 2: User dialogue acts in the Amani domain

remainder evenly among the annotators. These were then annotated by the same 3 annotators from the first round, using the same tools and instructions. The resulting annotated corpus will be referred to as the *expanded domain*, and it contains 799 unique utterances.

Results

Reliability

As a means of checking that the annotators had a similar understanding of the task, we calculated inter-annotator reliability using Krippendorff's α (Krippendorff 2004). Reliability is normally taken as a measure of the reproducibility of the annotation procedure, as codified in an annotation manual. In our case, however, the annotators were not working from detailed written guidelines; any shared understanding must therefore come from their previous experience. Reliability is therefore indicative of how straightforward the task is *before* implementing corrective measures such as detailed guidelines and domain and dialogue act improvements.

In addition to calculating agreement on the actual annotation (fully specified dialogue acts), we calculated the implicit agreement on whether a particular utterance was covered by the domain. This implicit agreement on coverage was calculated by collapsing all of the categories other than "unknown" into a single label. Table 3 shows the results of both calculations on the reliability samples for the original domain and the extended domain; the results from the pilot of Artstein et al. (2009) are also quoted here for comparison.

For the original domain, reliability was essentially the same as in the pilot: substantially above chance, but not as high as typically accepted norms. For the expanded domain we see a marked improvement in reliability, which indicates that the task is easier. The annotators and the guidelines were the same for both the original domain and expanded

	N	Individual acts			Implicit coverage		
		α	$A_o^{(a)}$	$A_e^{(a)}$	α	$A_o^{(a)}$	$A_e^{(a)}$
Pilot	224	0.49	0.55	0.11	0.38	0.74	0.58
Original	90[b]	0.49	0.58	0.19	0.33	0.67	0.52
Expanded	110[b]	0.63	0.65	0.07	0.39	0.79	0.66

[a]Krippendorff's α is defined in terms of observed and expected disagreement: $\alpha = 1 - D_o/D_e$. For expository purposes we have converted these into values representing observed and expected agreement: $A_o = 1 - D_o$, $A_e = 1 - D_e$.

[b]Several items were excluded from the reliability sample because they were not marked by all annotators.

Table 3: Inter-annotator reliability

domain, so the improvement in reliability is probably attributable to the better coverage of the domain.

The improvement in the reliability of matching utterances to specific dialogue acts does not carry over to the decision of whether an utterance is covered by the domain: here, the observed agreement of the expanded domain has gone up but so has the expected agreement, and consequently the reliability is at about the same level as the original domain. Our interpretation is that this remains a difficult decision for human judges – while domain coverage may increase, the boundary between what is covered and what is not remains fuzzy.

As an example of the fuzziness of the boundary we can take a fairly common follow-up on Amani's assertion that the suspect regularly has tea with the shopkeeper.

> Uh when he was having tea, was it close to where we are right now?
>
> Who was he having tea with?

While many such questions were judged to be out of domain, there was disagreement regarding the above two questions (and several others), on whether they were truly out of domain or if they could be mapped to questions about the suspect's location or daily routine, respectively. The expanded domain added several facts about the suspect's tea partner and drinking routine, so the above questions fall squarely within the expanded domain. However, expanding the domain did not make the domain's boundary any clearer: annotators disagreed on whether the following question could be mapped to a general question about the tea partner, or if it was outside the expanded domain.

> Why do you think he was having tea with the set?

We see that while adding facts to the domain increases the character's knowledge and thus its ability to understand user utterances, it does not necessarily make the boundaries of the character's knowledge any clearer.

Similar conclusions come from looking directly at the classification of the utterances in the reliability sample. Table 4 shows how many utterances in the reliability sample were mapped to a specific act as opposed to being judged to be out of domain, and whether the annotators agreed or disagreed about the mapping. In both the original and ex-

	Domain:	Original		Expanded	
		N	%	N	%
Specific act	Agree	32	30	53	45
	Disagree[a]	10	9	20	17
Out of domain	Agree	19	18	9	8
	Disagree[b]	46	43	35	30

[a]Utterances mapped to specific dialogue acts by all coders, where at least two coders disagreed on the dialogue act.

[b]Utterances mapped to specific dialogue acts by some coders and to "unknown" by other coders.

Table 4: Agreement on dialogue acts

Annotator	Original domain			Expanded domain		
	Total	In-domain		Total	In-domain	
		N	%		N	%
All	768	477–523	62–68	799	572–607	72–76
A	185	150	81	308	242	79
B	492	292	59	362	310	86
C	288	176	61	356	217	61

Table 5: Domain coverage

panded domain studies, the majority of disagreements are not on which dialogue act an utterance should be mapped to, but rather on whether an utterance is close enough to an existing dialogue act. The proportion of utterances mapped to specific dialogue acts is greater in the expanded domain, but the proportion of utterances on which there is agreement has not improved by much.

Domain coverage

We can define the overall coverage of a domain as the proportion of user utterances that are mapped to specific dialogue acts rather than "unknown" (we define coverage in terms of unique utterance types without regard to their frequency). Table 5 shows the coverage of the original and expanded domains, broken down by annotator; the overall coverage is reported as a range because sometimes annotators disagree as to whether an utterance is covered by the domain: the lower value considers such disagreements to be out of domain, while the higher value considers them to be in domain. The table shows that expanding the domain has improved the coverage by about 10 percentage points. We also see that annotators differ in their propensity to consider utterances to be in-domain, and that this propensity varies across the samples: the improvement in the overall coverage can be attributed to one specific annotator (coder B) for whom coverage increased substantially, coupled with the fact that the utterances in the expanded domain were more evenly balanced across the three coders.[1]

[1]The person who carried out the domain expansion was coder C, who turned out to be the one least likely to map an utterance to a specific dialogue act, both before and after the expansion.

Overall, we see that domain coverage is in line with the assessment of Artstein et al. (2009), that suitable domain expansion can bring coverage to about 80% of user utterances. Of the utterances that fall outside the expanded domain, many can still be represented using the dialogue act scheme – these constitute the "long tail" of user questions which have not been encountered or anticipated by the domain creators. Among the 227 utterances classified as outside the expanded domain by at least one annotator, we identified 94 (41%) that can plausibly be used to further expand the domain (among utterances classified as out-of-domain by all annotators the proportion is 79/192, also 41%). However, there are several types of user utterances which cannot be given a suitable representation in the scheme. These utterances demonstrate the limits for the simple dialogue act representation used in our tactical questioning system.

Temporal relations A fairly common utterance type encountered in our corpus is a question relating events in time (26 of the 227 out-of-domain utterances, or 11%).

Is Assad in the shop right now?

When have you seen the sniper on the second floor?

Did you see where he went after he had tea?

Questions with a temporal component are probably motivated by the particular scenario, where the task is to find information about a person related to a particular event. However, the representation language of facts as ⟨object, attribute, value⟩ triples does not explicitly encode time. While it is possible to represent certain static temporal facts using this scheme, for example ⟨assad, time-in-shop, now⟩, extensions would be required in order to represent temporal relations between events or perform temporal reasoning. Such an extension could be, for example, adding a temporal index to each fact, though this would increase authoring complexity.

Requests for elaboration Questioners often followed up on the character's responses by asking for additional details. Often such questions ask about facts that can be represented in the scheme; some questions, however, ask explicitly about information in relation to facts that were just provided (17 of 227 utterances, or 7%).

Do you know if there are anyone else in that building?

Have you seen him anywhere else?

OK then, do you think there is another door in the shop?

The representation language derives question dialogue acts from facts consisting of ⟨object, attribute, value⟩ triples; the only relations between facts are those that occur implicitly, when two facts share an object and attribute but differ on value, or share an object but differ on attribute. For example, if the domain representation includes facts of the form ⟨building, occupant, strange-man⟩, ⟨building, occupant, . . . ⟩ then the dialogue manager can interpret the question *Do you know if there are anyone else in that building?* as asking for values that have not yet been provided. Asking for elaboration on objects and attributes while keeping the attribute

or value fixed would require moving from a hierarchical domain representation to a relational one.

Relations between objects A small number of question concern relations between objects (3 of 227 utterances, or 1%).

Could they be found in the same area as him?

Since the domain represents all facts as ⟨object, attribute, value⟩ tuples, any fact about two objects needs to be encoded by specifying one object as a dependent value of the other. Representing relationships between the two domain objects would require a move toward a relational semantics, much like the requests for elaboration above.

Reason and evidence A common type of question is to ask the character about the reasons or evidence for her assertions (19 of 227 utterances, or 8%).

Do you know why he was having tea?

How do you know this?

And did you see him actually pull the trigger

In the current domain representation, facts do not carry any additional information beyond the content of the fact itself. Adding reasons would require an extension of the representation, for example by enriching facts beyond ⟨object, attribute, value⟩, or alternatively by enabling relations between facts.

Assertions Our dialogue act model is geared towards the user questioning the character: each fact in the domain gives rise to question-type user dialogue acts, and assertion-type acts by the character. However, we do find that the users occasionally make assertions (21 of 227 utterances, or 9%).

I have a soldier who was wounded by a sniper.

My men are outside right now and we will be in this area for a long time.

Well, I noticed that you're a school teacher ma'am.

The underlying domain representation is symmetrical, so it is possible to add these facts to the user's domain, which would give rise to user dialogue acts of type *assert* and corresponding character question dialogue acts. However, the above examples show that user assertions in tactical questioning dialogues are more than mere statements of fact; having the character ask questions about these assertions would be pointless. To do something useful with these assertions, the system would require and inference component to capture the intention behind them.

Conditional offers Offers are represented in the domain by ⟨character, action⟩ pairs, where the action is a specific offer; some user offers come with conditions attached (10 of 227 utterances, or 4%).

We can discuss money if you give me more information.

If we were able to supply you with a weapon or armed protection, would you feel safe to tell us information?

Even though the instructions to the participants do not impose any penalty on making an unconditional offer such as providing safety or secrecy, it appears that the participants sometimes attach conditions to their offer as a means of leverage. Interpreting conditions for offers and designing suitable policies would require a richer representation than the current ⟨character, action⟩ form.

Topic setting A small number of utterances were attempts by the user to set the topic (4 of 227 utterances, or 2%).

Can we talk about the shooter?

I wanna talk about the sniper not guns.

The dialogue act scheme does not include moves to set the topic of conversation. This is a straightforward addition, because the system already keeps track of the conversation topic, and the scheme already includes grounding dialogue acts for confirming topics. Dialogue acts of type *set-topic* have been added to the scheme subsequent to the experiment.

Compound utterances A fair number of utterances consisted of multiple questions strung together (20 of 227 utterances, or 9%).

Ma'am how do they look like? Are they tall? Are they short? Do they have black hair or mustache?

Do you know where he was located? Was he in a building or was he in a mosque or something like that?

Since the system assigns a single dialogue act to each user speech event (delimited by a press and release of a button), these compound utterances cannot be represented. The proper way to deal with them is by adding a module that splits them into smaller units that can be interpreted.

Conclusion

Our study has shown that a set of dialogue acts, generated automatically from a domain representation designed for easy scenario authoring by domain experts with little detailed knowledge of dialogue systems, can achieve substantial coverage of actual user utterances employed in live conversation with a virtual character. After an initial domain has been adjusted and augmented based on several hundred user utterance, coverage rises to approximately 72%–76% of unseen utterances. Combined with dialogue management techniques to recover from misunderstandings, this level of coverage should be sufficient to allow a character to sustain a coherent interaction with the user.

Among those utterances that are not covered, the largest group (around 40%, or 12% of the total utterances) are utterances that do fit in the scheme, but have not been encountered or anticipated by the domain creators. It is inevitable that such a "long tail" of rare unseen utterances should exist. The remaining out-of-domain utterances, about 17% of the total, consist mostly of the following types: questions about temporal relations, relations between facts and relations between objects, questions about reason and evidence, assertions by the user, conditional offers, attempts to set the topic of conversation, and compound utterances. Most of these utterance types fall outside the representation capability of the system, and thus constitute the limits of the simple dialogue act scheme.

We end with a caveat about our results. Our corpus of user utterances has been collected using one specific scenario, which may have influenced the questions the users wanted to ask. For example, the large number of questions about temporal relations is probably due to the fact that the users are tasked with finding information related to an event. Our user group was also fairly homogeneous, consisting of military cadets enrolled in a negotiation course, which may have influenced their approach and strategies employed in the interaction. We expect that a different scenario or a different population of users may give rise to a somewhat different distribution of utterances. Nevertheless, we believe that this study is a good start for exploring how far the simple dialogue act representation can take us, and what actual user utterances lie beyond its scope.

Acknowledgments

The project or effort described here has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Allwood, J. 1980. On the analysis of communicative action. In Brenner, M., ed., *The Structure of Action*. Basil Blackwell. Also appears as Gothenburg Papers in Theoretical Linguistics 38, Dept of Linguistics, Göteborg University.

Artstein, R.; Gandhe, S.; Rushforth, M.; and Traum, D. 2009. Viability of a simple dialogue act scheme for a tactical questioning dialogue system. In *DiaHolmia 2009: Proceedings of the 13th Workshop on the Semantics and Pragmatics of Dialogue*.

Bunt, H. C. 1999. Dynamic interpretation and dialogue theory. In Taylor, M. M.; Néel, F.; ; and Bouwhuis, D. G., eds., *The Structure of Multimodal Dialogue, Volume 2*. Amsterdam: John Benjamins.

Gandhe, S.; DeVault, D.; Roque, A.; Martinovski, B.; Artstein, R.; Leuski, A.; Gerten, J.; and Traum, D. 2008. From domain specification to virtual humans: An integrated approach to authoring tactical questioning characters. In *proceedings of Interspeech 2008*.

Gandhe, S.; Whitman, N.; Traum, D.; and Artstein, R. 2009. An integrated authoring tool for tactical questioning dialogue systems. In *6th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Krippendorff, K. 2004. *Content Analysis: An Introduction to Its Methodology*. Thousand Oaks, California: Sage, second edition. chapter 11, 211–256.

Leuski, A., and Traum, D. 2008. A statistical approach for text processing in virtual humans. In *Proccedings of 26th Army Science Conference*.

Leuski, A.; Patel, R.; Traum, D.; and Kennedy, B. 2006. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, 18–27. Sydney, Australia: Association for Computational Linguistics.

Sinclair, J. M., and Coulthard, M. 1975. *Towards an Analysis of Discourse: The English Used by Teachers and Pupils.* Oxford University Press.

Traum, D. R., and Larsson, S. 2003. The information state approach to dialogue management. In van Kuppevelt, J., and Smith, R. W., eds., *Current and New Directions in Discourse and Dialogue*. Dordrecht: Kluwer. chapter 15, 325–353.

Traum, D.; Swartout, W.; Gratch, J.; and Marsella, S. 2008. A virtual human dialogue model for non-team interaction. In Dybkjær, L., and Minker, W., eds., *Recent Trends in Discourse and Dialogue*, volume 39 of *Text, Speech and Language Technology*. Dordrecht: Springer. chapter 3, 45–67.

Traum, D. 2000. 20 Questions on Dialogue Act Taxonomies. *J Semantics* 17(1):7–30.

U.S. Army. 2006. Police intelligence operations. Field Manual FM 3-19.50, U.S. Army. Appendix D: Tactical Questioning.

Walker, M. A.; Passonneau, R.; and Boland, J. E. 2001. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 515–522. Morristown, NJ, USA: Association for Computational Linguistics.

Appendix: participant instructions

The following information sheet was given to all experiment participants, to serve as background while talking to the character.

Situation: You are a 2LT Platoon Leader, stationed in a small village in Iraq. While on patrol yesterday, your platoon came under sniper fire, which seriously wounded one of your soldiers. Local intelligence indicates a woman named Amani witnessed the sniper.

Mission: You will question Amani Omar Al-Mufti in order to determine the location and appearance, and daily activities of the sniper that wounded the soldier.

Execution: You received permission from Amani's eldest brother to question her. He is present during the questioning to act as a chaperone, however, you will not need to speak any further with the brother. Your platoon will provide security outside during your questioning inside. Gather intelligence from Amani and offer to keep her family safe if she shows concern.

If Amani becomes too hostile or indicates that she no longer has time, end the interview before too much ill will is generated, without pressing her on any issues. You may have the opportunity to meet with her in the future.

Service Support: N/A

Command and Signal: N/A

Screening Report

A: Report Number: DTG:

B: Capture Data

N/A

C: Biographical Information

Full Name/ Rank/ Service Number:
  a. Amani Omar Al Mufti
  b. Civilian
  c. N/A

Date/ Place of Birth:
  a. 16AUG1983
  b. Local

Sex/ Marital Status/ Religion:
  a. Female
  b. Single
  c. Islam (Shiite)

Full Unit Designation/ Unit Code:
  a. N/A
  b. N/A

Duty Position:
  a. Housekeeper and Guardian of Siblings
  b. Teacher at private K-12 school

Military Education/ Experience:
  a. N/A
  b. N/A

Civilian Education/ Experience:
  a. Completed Secondary School, some college
  b. She is an English teacher at a K-12 school.

Languages Spoken (Fluency):
  a. Arabic (Native)
  b. English (Fluent)

D: Observations

Physical Condition:
  a. No Issues

Uniform Type/ Condition:
  a. N/A
  b. N/A

Assessment of Knowledgeability:

She is likely to have personal knowledge about the gunman's appearances and his location.

E: Recommendations

Relationship Building:
Begin the questioning with greeting Amani. Gaining her trust and comfort is key to getting any answers from her.

Information Gathering:
Focus on finding out what she knows about the suspected sniper, his location and reasons she suspects him. If being friendly and respectful is not effective, explain to her that she and her family can have protection. If she wants anything in return for information, you are free to make an offer or refuse to make one. Make sure she understands that you value the importance of secrecy due to the sensitive nature of the visit.

Learning Dialogue Agents with Bayesian Relational State Representations

Heriberto Cuayáhuitl
German Research Center for Artificial Intelligence (DFKI)
heriberto.cuayahuitl@dfki.de

Abstract

A new approach is developed for representing the search space of reinforcement learning dialogue agents. This approach represents the state-action space of a reinforcement learning dialogue agent with relational representations for fast learning, and extends it with belief state variables for dialogue control under uncertainty. Our approach is evaluated, using simulation, on a spoken dialogue system for situated indoor wayfinding assistance. Experimental results showed rapid adaptation to an unknown speech recognizer, and more robust operation than without Bayesian-based states.

Introduction

Reinforcement learning dialogue agents have a promising application for adaptive conversational interfaces. Unfortunately, three main problems affect their practical application. The first, *the curse of dimensionality*, causes the state space to grow exponentially in the number of state variables. This problem has been addressed by function approximation techniques (Denecke, Dohsaka, and Nakano 2004; Henderson, Lemon, and Georgila 2005; Chandramohan, Geist, and Pietquin 2010); and by divide-and-conquer approaches (Cuayáhuitl et al. 2010; Lemon 2011). Second, the dialogue agent *operates under uncertainty* (the most obvious source is automatic speech recognition errors, but not the only source). This problem has been addressed by sequential decision-making models under uncertainty (Roy, Pineau, and Thrun 2000; Williams 2006; Thomson 2009; Young et al. 2010). Third, reinforcement learning methods usually require many dialogues to find optimal policies, resulting in *slow learning*. This last problem has been addressed by incorporating prior knowledge into the decision making process (Singh et al. 2002; Heeman 2007; Williams 2008; Cuayáhuitl 2009). Because of such problems, the current practice in dialogue optimization consists in inducing behaviour offline, from a corpus of real dialogues or from simulations. When the learnt policies are then deployed they behave with frozen optimization. The rest of the paper contributes to tackle these problems by proposing a new approach to represent the agent's state-action space.

Dialogue Optimization Under Uncertainty

A human-machine dialogue can be defined as a finite sequence of information units conveyed between conversants, where the information can be described at different levels of communication such as speech signals, words, and dialogue acts. Figure 1 illustrates a model of human-machine interaction. An interaction under uncertainty between both conversants can be briefly described as follows: the machine receives a distorted user speech signal $\tilde{x}_t$ from which it extracts a user dialogue act $\tilde{u}_t$ and enters it into its knowledge base; the machine then updates its belief dialogue state b_t (i.e. a probability distribution over dialogue states) with information extracted from its knowledge base; this dialogue state is received by the spoken dialogue manager in order to choose a machine dialogue act a_t, which is received by the response generation module to generate the corresponding machine speech signal conveyed to the user.

A conversation follows the sequence of interactions above in an iterative process between both conversants until one of them terminates it. Assuming that the machine receives a numerical reward r_t for executing action a_t when the conversational environment makes a transition from belief state b_t to state b_{t+1}, a dialogue can be expressed as $D = \{b_1, a_1, r_2, b_2, a_2, r_3, ..., b_{T-1}, a_{T-1}, r_T, b_T\}$, where T is the final time step. Such sequences can be used by a reinforcement learning agent to optimize the machine's dialogue behaviour. Although human-machine conversations can be used for optimizing dialogue behaviour, a more common practice is to use simulations.

A reinforcement learning dialogue agent aims to learn its behaviour from interaction with an environment, where situations are mapped to actions by maximizing a long-term reward signal (see (Sutton and Barto 1998) for an introduction to reinforcement learning). Briefly, the reinforcement learning paradigm works by using the formalism of Markov Decision Processes (MDPs). An MDP is characterized by a finite set of states S, a finite set of actions A, a probabilistic state transition function, and a reward function that rewards the agent for each selected action. Solving the MDP means finding a mapping from observable states to actions corresponding to $\pi^*(s_t) = \arg\max_{a_t \in A} Q^*(s_t, a_t)$, where the Q-function specifies the cumulative rewards for each state-action pair. The optimal policy can be learnt by dynamic programming or reinforcement learning algorithms.

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 9–15,
Barcelona, July 2011

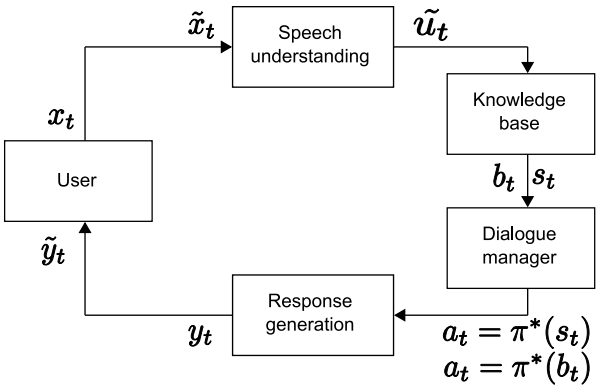

Figure 1: A pipeline model of human-machine interaction, where observable dialogue state s_t or belief dialogue state b_t is used by the dialogue manager to choose action a_t.

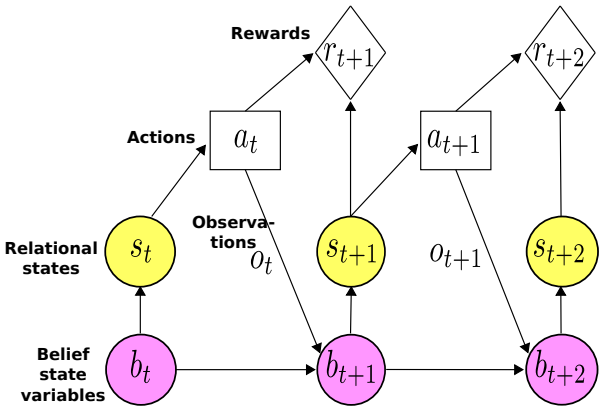

Figure 2: Dynamics of an MDP-based dialogue manager using Bayesian Relational state representations.

An alternative but more computationally intensive model for sequential decision-making under uncertainty is the Partially Observable Markov Decision Process (POMDP) model. In a POMDP the dialogue state is not known with certainty (as opposed to an MDP); i.e. since the agent does not know the state s exactly, it must maintain a belief state over the possible states S (Young et al. 2010). The characterization of a POMDP extends an MDP with a set of observations or perceptions from the environment (e.g. keywords from the user utterances) $\Omega = \{o_1, o_2, ..., o_n\}$, and an observation function $O(s, a, o)$ that specifies a perceived observation o from selecting action a in state s with probability $P(o|s, a)$. Thus, a POMDP can be seen as an MDP over a belief space, where the observable states are replaced by belief states. Solving the POMDP can be described as finding a mapping from belief states to actions corresponding to $\pi^*(b_t) = \arg\max_{a_t \in A} Q^*(b_t, a_t)$, where the Q-function specifies the cumulative rewards for each belief state and action. The rest of the paper describes an approach that extends MDP-based reinforcement learning conversational agents with beliefs states, which can be seen as learning agents with a characterization between MDPs and POMDPs.

A Bayesian-Relational Approach for Dialogue Control Under Uncertainty

Figure 2 shows the presented approach which unifies two concepts: (a) *relational representations* imposed on an MDP state-action space; and (b) *belief state variables* extending the fully-observed state variables by using partition-based Bayesian networks.

Dialogue as a Relational MDP

An MDP is typically represented with propositional representations (e.g. a set of binary features), which result into exponential growth. A relational MDP mitigates that problem by using tree-based and high-level representations resulting in the following benefits: (a) compression and more expressive description of the state-action space, (b) straightforward incorporation of prior-knowledge into the policy, (c) generalization for reusable behaviours, and (d) fast learning.

A relational MDP is a generalization of an MDP specified with representations based on a logical language (van Otterlo 2009). A relational MDP can be defined as a 5-tuple $<S, A, T, R, L>$, where element L is a language that provides the mechanism to express logic-based representations. We describe L as a context-free grammar to represent formulas compounded by predicates, variables, constants and connectives similar to (Russell and Norvig 2003), Chaper 8. Whilst the state set S is generated from an enumeration of all logical forms in grammar L, the actions A available in a given state are constrained by the logical forms in L. A sample relational state is expressed by a set of predicates: '$Salutation(greeting) \wedge Slot(x, confirmed) \wedge SlotsToConfirm(none) \wedge DatabaseTuples(none)$'. This representation indicates that slot x has been confirmed, there are no slots to confirm and no database tuples. A sample relational action is expressed as follows: '$request \leftarrow Salutation(greeting) \wedge Slot(x, unfilled) \wedge SlotsToConfirm(none)$'. This expression indicates that the action 'request' is valid if the logical expression is true.

Relational MDPs with Belief States

Because dialogue states are not known with certainty, POMDPs have been adopted for policy optimization under uncertainty (Roy, Pineau, and Thrun 2000; Williams 2006; Henderson and Lemon 2008; Thomson 2009; Young et al. 2010). Moreover, because POMDPs are computationally intensive and hard to scale up, in this paper we propose to approximate the belief states of a relational MDP with belief state variables. This approximation is used to scale up to more complex conversational systems. The belief states can be defined as $b(s) = \frac{1}{Z}\Pi p(X_i \in s)$, where $p(X_i \in s)$ is the probability distribution of predicate X_i in state s, and Z is a normalization constant.

For the belief states, we maintain a Bayesian Network (BN) for each predicate $X_i \in s$. A BN models a joint probability distribution over a set of random variables and their dependencies based on a directed acyclic graph, where each node represents a variable Y_j with parents $pa(Y_j)$

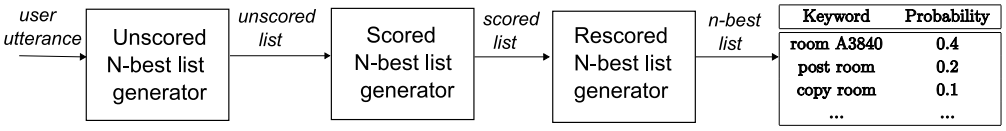

Figure 3: Block diagram for generating N-best list. Whilst scored lists are based on beta distributions and ASR error rates (see Fig. 5), re-scored n-best lists are based on posterior distributions (useful for the belief states) derived from Bayesian Networks.

(Jensen 1996). The Markov condition implies that each variable is only dependent on its parents, resulting in a unique joint probability distribution expressed as $p(Y) = \Pi p(Y_j|pa(Y_j))$, where every variable is associated with a conditional probability distribution $p(Y_j|pa(Y_j))$. Such a network is used for probabilistic reasoning, i.e. the calculation of posterior probabilities given some observed evidence. To that end, we use efficient implementations of the variable elimination and junction tree algorithms (Cozman 2000). In addition, because the size of domain values D for each variable can be large (which results in high computational expense), we use random variables with partitions $D = \{\tilde{D}_i\}$ expressed as

$$D = \begin{cases} \tilde{D}_0 \leftarrow \text{item}_1, \text{item}_2, \text{item}_3 \ldots \text{item}_N, \text{other} \\ \tilde{D}_1 \leftarrow \text{item}_{N+2}, \text{item}_{N+3}, \text{item}_{N+4} \ldots \text{item}_{N'}, \text{other} \\ \ldots \\ \tilde{D}_M \leftarrow \text{item}_{N'+2}, \text{item}_{N'+3} \ldots \text{item}_{N''}, \text{other} \end{cases}$$

where $|\tilde{D}_k| \leq \max$. The entry 'other' is initialized with probability 1, which changes with belief updating during the course of the interaction. At each time step, the networks and corresponding posteriors are updated based on the perceived observations (i.e. ASR N-best lists) from the environment. The N-Best lists were generated according to the procedure shown in Figure 3. Once the posteriors are updated, their 1-best hypotheses are used in the relational states of the MDP.

Belief Updating of the Dialogue State

The partition-based Bayesian Networks (BNs) described above use multiple minimal BNs defined by $p(V_k^i|R_k^i, P_k^i)$, where index i denotes a predicate in the dialogue state and index k denotes a partition in predicate i. The meaning of such random variables is as follows: R_k^i is used for speech recognition at time step t, P_k^i is used for speech recognition at time step $t-1$, and V_k^i is the belief of predicate i. The belief updating procedure is as follows. First, compute an N-best list for each keyword in the user utterance. For each entry in the N-best list, get the partition of the current entry denoted as $\tilde{D}_k^i$. Assign the corresponding probabilities to the random variable R_k^i. Update the probability of entry 'other' according the new observations. If $t = 0$ then assign the probability distribution of R_k^i to P_k^i, else assign the probability distribution of V_k^i to P_k^i so that it can maintain the previous beliefs. Finally, the state with the highest probability in the random variables $V_{\forall k}^i$—computed by combining partitions omitting the entry 'other' and redistributing probability mass accordingly—is used in predicate X_i of dialogue state s. This implies that there is a single belief for each predicate, even if it appears in multiple dialogue states.

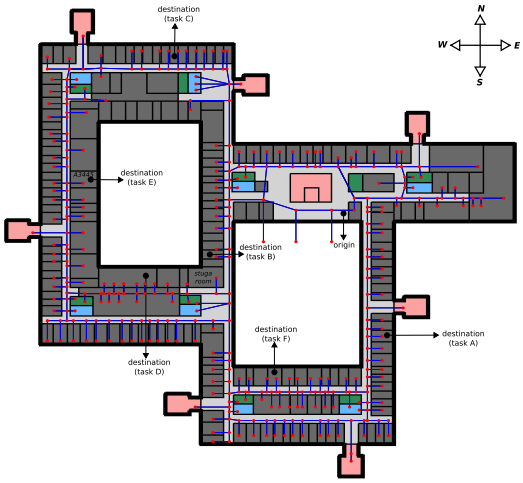

Figure 4: Map of the navigation environment including a superimposed route graph specifying the navigational space. The black circles represent origin and destination locations.

Experiments and Results

We tested our approach in a learning agent that collects information for situated indoor navigation using simulated speech-based interactions. The task of the user is to navigate from an origin to a destination based on instructions received from a dialogue system. After each instruction the user has to say where he/she is and the agent has to guide the user to the goal location (see Figure 4, and (Cuayáhuitl and Dethlefs 2011a) for a dialogue system of this type but without belief monitoring). This scenario represents at least the following sources of uncertainty: What did the user say? Where is the user? What does the user know? This paper focuses its attention on the first source of uncertainty.

The Simulated Conversational Environment

The system and user verbal contributions are based on the Dialogue Act (DA) types shown in Table 1 combined with the attributes {origin, destination}. This makes a set of 10 user DAs and 14 system DAs. We used the conditional probability distribution $p(u|a)$ for simulating user dialogue acts u given the last machine dialogue acts a. The user responses were coherent with probability 0.9 and random otherwise, a speech recognition error rate of 20% was simulated and ambiguity of domain values of 10%.

In addition, we modelled Automatic Speech Recognition (ASR) events from *beta* continuous probability distri-

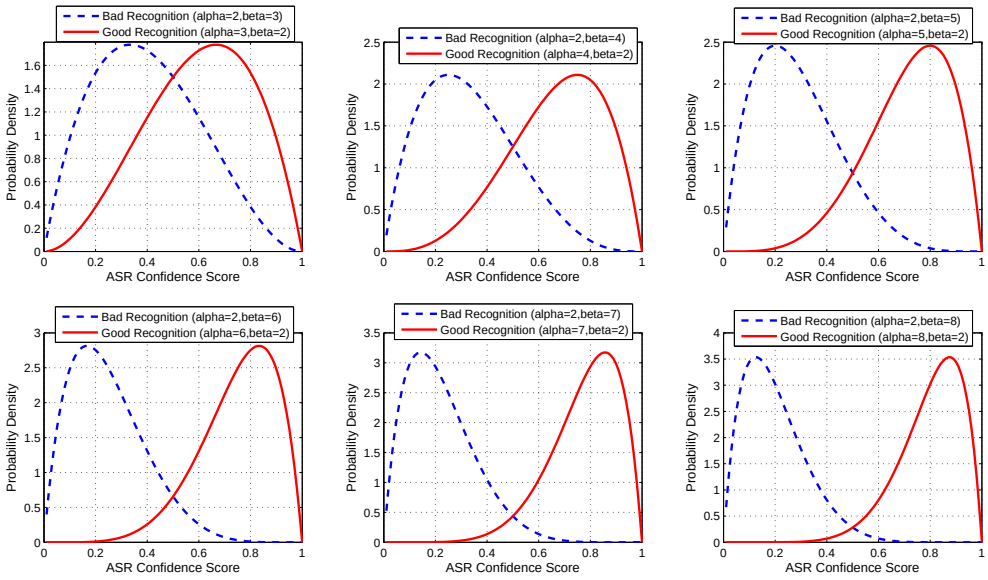

Figure 5: Beta probability distributions for modelling speech recognition events in simulation-based dialogue strategy learning.

butions (see Figure 5), which have been applied to statistical dialogue modelling by (Williams and Balakrishnan 2009; Williams 2010). The *beta* distribution is defined in the interval (0, 1) and it is parameterized by two positive shape parameters referred to as α and β. The probability density function of a *beta* distribution is expressed as

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\,dx},$$

where the denominator represents the beta function, α and β are positive real numbers (which can be estimated from data), and $0 \leq x \leq 1$. Our simulations used (α=2,β=5; α=5,β=2) for bad and good recognition, respectively.

Characterization of the Learning Agent

Figure 6 shows the context-free grammar specifying the language for the relational states in our learning agent. Whilst the enumeration using a propositional representation represents a total of $100^2 \times 3^3 = 270$ thousand states (100^2 recognized locations for each confidence score from 0.01 to 1.0; 3 values for unfilled, filled, confirmed origin; 3 values for unfilled, filled, confirmed destination; and 3 values for ambiguous user dialogue act), the relational representation only required 21 thousand combinations (7.7% of the propositional representation). The actions constrained with the relational states (i.e. logical forms in grammar L) are expressed as

$$A = \begin{cases} \text{Request(origin,destination)} \leftarrow l_{01} \\ \text{Request(origin)} \leftarrow l_{03} \vee l_{12} \\ \text{Request(destination)} \leftarrow l_{02} \vee l_{08} \\ \text{Apology(origin,destination)+} \\ \qquad \text{Request(origin,destination)} \leftarrow l_{04} \\ \text{Apology(origin)+Request(destination)} \leftarrow l_{02} \vee l_{11} \\ \text{Apology(destination)+Request(origin)} \leftarrow l_{03} \vee l_{09} \\ \text{ImpConf(origin)+Request(destination)} \leftarrow l_{02} \\ \text{ImpConf(destination)+Request(origin)} \leftarrow l_{03} \\ \text{ExpConf(origin)} \leftarrow l_{02} \vee l_{11} \\ \text{ExpConf(destination)} \leftarrow l_{03} \vee l_{09} \\ \text{ExpConf(origin,destination)} \leftarrow l_{04} \\ \text{Clarify(origin)} \leftarrow l_{05} \vee l_{13} \\ \text{Clarify(destination)} \leftarrow l_{04} \vee l_{10} \\ \text{Clarify(origin,destination)} \leftarrow l_{08}. \end{cases}$$

It can be observed that whilst the propositional state-action space would use $100^2 \times 3^3 \times 14$ =3.8 million state-actions, the constrained state-action space only uses 32 thousand (less than 1% of the propositional one). The goal state is defined when the origin and destination locations are confirmed (a sample dialogue is shown in Table 2). In addition, the Bayesian networks (with semi-hand-crafted structure and parameters based on the spatial environment) for modelling the beliefs of predicates in the relational states are shown in Figure 7. Since the posteriors can have a large number of probabilities (e.g. the conditional probability table for predicate 'UserOrigin' has $200^3 \times 2$=16 million entries), we partitioned large networks with entries based on locations per navigation segment (from one junction to another) allowing a maximum of domain values $\max \leq 30$ (i.e. multiple instantiations of a Bayesian net with smaller conditional probability tables). Finally, the reward function is defined by the following rewards: 0 for reaching the goal

$L := l_1\ l_2\ l_3\ l_4\ l_5\ l_6\ l_7\ l_8\ l_9\ l_{10}\ l_{11}\ l_{12}\ l_{13}\ l_{14}$
l_1:= UserOrigin(unfilled) $\wedge$ UserDestination(unfilled) $\wedge$ AmbiguousUserDialogueAct(unknown)
l_2:= UserOrigin(filled,score) $\wedge$ UserDestination(unfilled) $\wedge$ AmbiguousUserDialogueAct(no)
l_3:= UserOrigin(unfilled) $\wedge$ UserDestination(filled,score) $\wedge$ AmbiguousUserDialogueAct(no)
l_4:= UserOrigin(filled,score) $\wedge$ UserDestination(filled,score) $\wedge$ AmbiguousUserDialogueAct(no)
l_5:= UserOrigin(filled,score) $\wedge$ UserDestination(unfilled) $\wedge$ AmbiguousUserDialogueAct(yes)
l_6:= UserOrigin(unfilled) $\wedge$ UserDestination(filled,score) $\wedge$ AmbiguousUserDialogueAct(yes)
l_7:= UserOrigin(filled,score) $\wedge$ UserDestination(filled,score) $\wedge$ AmbiguousUserDialogueAct(yes)
l_8:= UserOrigin(confirmed) $\wedge$ UserDestination(unfilled) $\wedge$ AmbiguousUserDialogueAct(no)
l_9:= UserOrigin(confirmed) $\wedge$ UserDestination(filled,score) $\wedge$ AmbiguousUserDialogueAct(no)
l_{10}:= UserOrigin(confirmed) $\wedge$ UserDestination(filled,score) $\wedge$ AmbiguousUserDialogueAct(yes)
l_{11}:= UserOrigin(filled,score) $\wedge$ UserDestination(confirmed) $\wedge$ AmbiguousUserDialogueAct(no)
l_{12}:= UserOrigin(unfilled) $\wedge$ UserDestination(confirmed) $\wedge$ AmbiguousUserDialogueAct(no)
l_{13}:= UserOrigin(filled,score) $\wedge$ UserDestination(confirmed) $\wedge$ AmbiguousUserDialogueAct(yes)
l_{14}:= UserOrigin(confirmed) $\wedge$ UserDestination(confirmed) $\wedge$ AmbiguousUserDialogueAct(no)
score := $0.01\ \vee\ 0.02\ \vee\ 0.03\ \vee 0.04\ \vee 0.05\ \vee ...\ \vee\ 0.97\ \vee\ 0.98\ \vee\ 0.99 \vee 1$

Figure 6: Context-free grammar defining the language L for collecting information in the wayfinding domain. See (Cuayáhuitl and Dethlefs 2011b) for a more complete state representation of the wayfinding interaction (including information presentation).

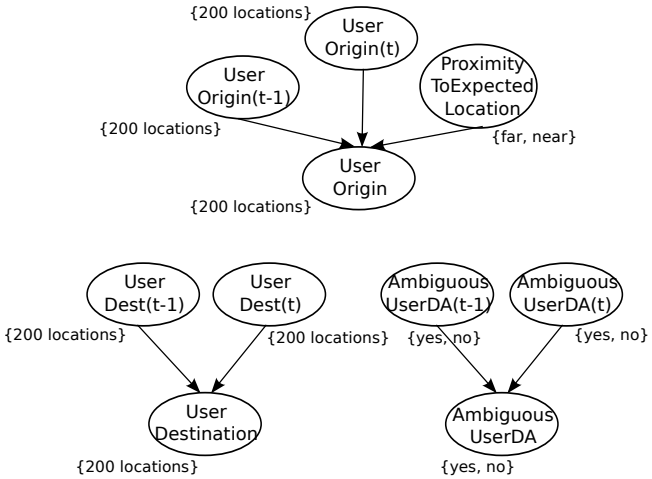

Figure 7: Bayesian networks for modelling the beliefs of predicates in the relational states. The domain values of each random variable is shown in curly brackets. Notice that the top and bottom left networks use multiple networks (partitions) to handle smaller conditional probability tables.

state and -10 otherwise. We used the Q-Learning algorithm (Sutton and Barto 1998). The learning rate parameter α decays from 1 to 0 according to $\alpha = 100/(100 + \tau)$, where τ represents elapsed time-steps. The action selection strategy used ϵ-Greedy with $\epsilon = .01$, undiscounted rewards, and Q-values initialized to 0.

Experimental Results

Figure 8 shows the learning curves of induced behaviour with the proposed approach. One thing to notice is that reinforcement learning with relational representations using a constrained state-action space is dramatically faster than without constraints. Whilst the latter requires five orders of magnitude to learn a stable policy, the former only requires three orders of magnitude. Two key characteristics of relational state-action spaces are (1) they are easy to specify and to read, and (2) they offer the mechanism to generate coherent dialogues (even before learning). Surprisingly, the relational representations have been ignored in the learning dialogue systems field. Another thing to notice is that learnt policies with belief state variables help to improve performance more (due to more accurate recognitions) than without tracking beliefs from the environment. We measured the average system turns of the last 1000 training dialogues and found that constrained learning with belief states outperforms its counterpart (constrained learning without belief states) by an absolute 15% in terms of average system turns. We also compared the average system turns of the first 1000 training dialogues and the last 1000 training dialogues for the best policy (with beliefs), and found that the latter phase outperformed the first one by 2 system turns. This indicates that the hand-coded policy with relational representations was improved by policy learning.

Furthermore, our approach scales up to larger domain values because (a) the size of the relational state-action space is location-independent, and (b) even when the Bayesian Networks (BNs) are slot-dependent, the partitioned approach makes them scalable. It remains to be investigated the scalability limits of our approach with larger and more complex BNs. Nonetheless, the partition-based BNs reduce computational demands for loading, updating and querying beliefs in comparison to non-partitioned BNs. Although the results above require an evaluation in a realistic environment, the proposed approach is promising for optimizing dialogue behaviour in unknown and uncertain environments (which require fast learning with continuous belief tracking).

Conclusions and Future Work

We have described a unified approach for representing search spaces of reinforcement learning dialogue agents, which aims for efficient and robust operation combined with straightforward design. To this end we use logic-based representations in the state-action space, and extend them with belief states by using partition-based Bayesian networks.

Dialogue Acts	Sample Utterance
Provide(ori)	I am in front of room B3090
Provide(des)	How do I get to Dr. Watson's office?
Provide(ori,des)	How do I get from room B3090 to Dr. Watson's office?
Reprovide(ori)	I said in front of room B3090
Reprovide(des)	I meant to Dr. Watson's office?
Reprovide(ori,des)	I asked how do I get from room B3090 to Dr. Watson's office?
Confirm(ori)	Yes, I did.
Confirm(des)	Yes, I said that.
Confirm(ori,des)	Yes, please.
Silence()	[*remain in silence*]
Request(ori,des)	What is your origin and destination?
Request(ori)	Where are you?
Request(des)	Where would you like to go?
Apology(ori,des)+ Request(ori,des)	Sorry, from where to where?
Apology(ori)+ Request(ori)	Sorry, where are you?
Apology(des)+ Request(des)	Sorry, what is your destination?
ImpConf(ori)+ Request(des)	Okay, from room B3090, to where ?
ImpConf(des)+ Request(ori)	Okay, to room B3090, where are you?
ExpConf(ori,des)	Yes
ExpConf(ori)	No
ExpConf(des)	Yes I did
Clarify(ori)	Do you mean James Watson or Peter Watson?
Clarify(des)	Do you mean Copy room or Post room?
Clarify(ori,des)	Do you want to go to the Copy room or Post room?

Table 1: Dialogue Acts for collecting information in the situated navigation domain, where ori=origin and des=destination. The groups correspond to user and system dialogue acts, respectively.

Our experimental results provide evidence to conclude that our method is promising because it combines fast learning with robust operation. By proposing relational state-action spaces, it makes a concrete contribution to conversational interfaces which learn their dialogue behaviour. Although this approach scales up to large domain values, it can be extended with hierarchical control to deal with large relational states and optimization of large-scale conversational interfaces; e.g, hierarchical reinforcement learning dialogue agents such as (Cuayáhuitl et al. 2010; Cuayáhuitl and Dethlefs 2011a) can be extended with Bayesian relational representations.

Related work closest to ours is the following. (Lecoeuche 2001) used reinforcement learning with relational representations, but he did not model beliefs. (Horvitz and Paek 1999; Paek and Horvitz 2000; Bohus and Rudnicky 2005; 2006; Skantze 2007) modelled beliefs in dialogue systems, but they did not optimize conversations using reinforcement learning. In general, our approach lies between the

Agent	Dialogue Act	Utterance
Sys	Request(ori,des)	What is your origin and destination?
Usr	Provide(ori,des)	*I want to go from room B3090 to Dr. Watson's office?*
Sys	ImpConf(ori) +Request(des)	Okay, from room B3090, to where?
Usr	Reprovide(des)	*Dr. Watson's office?*
Sys	Clarify(des)	Do you mean James Watson or Peter Watson?
Usr	Reprovide(des)	*Peter Watson*
Sys		[provides a route instruction]
Usr	Provide(ori)	[executes the route instruction] I am in front of the lifts
Sys	Apology(ori) +Request(ori)	Sorry, where are you?
Usr	Reprovide(ori)	*In the corridor of the lifts*
Sys	Clarify(des)	Do you mean the lifts next to the language learning center?
Usr	Confirm(ori)	*Ehhh, yes*
Sys		[provides a route instruction]
Usr	Provide(ori)	[executes the route instruction] *Okay, now I can see offices B3280 and B3285*
Sys		[provides a route instruction]
...		[and so on until reaching the goal]

Table 2: Fragment of a conversation in the situated wayfinding domain. This dialogue focuses its attention on collecting information as the user carries out the navigation task. We assume that the user carries a hand-held device with him/her to communicate with the system using spoken interaction.

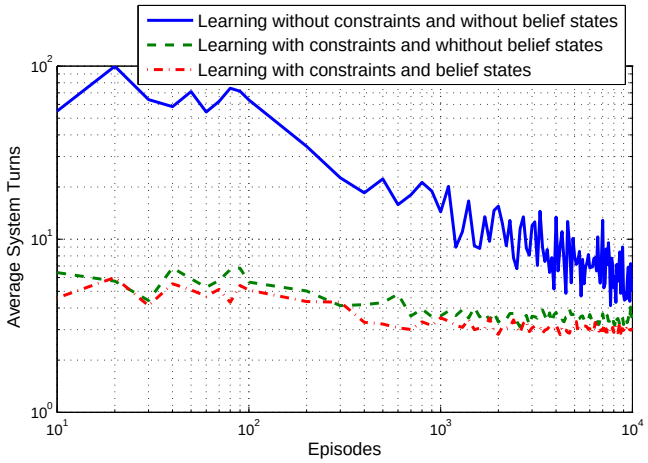

Figure 8: Learning curves of induced dialogue behaviour (averaged over 10 runs), where all agents started to learn after 100 episodes (to plot the performance before learning). Learning with constraints and belief states (i.e. the best learning curve) outperforms its counterpart (learning with constraints and without belief states) by an absolute 15% in terms of average system turns due to more accurate speech recognition. The best learnt dialogue policy improved the hand-coded constraints from 5 to 3 system turns, derived from a comparison of the first and the last 1000 dialogues.

MDP and POMDP models (Roy, Pineau, and Thrun 2000; Williams 2006; Thomson 2009; Young et al. 2010). Since we model beliefs of predicates (with short histories) in the dialogue state instead of beliefs of entire dialogue states (with long histories), our approach is expected to be less robust than the POMDP model but at the same time more scalable. A theoretical and experimental comparison between our and a POMDP-based approach is left as future work. Another future direction is to use (non-)linear function approximation for tackling very large relational state-action-spaces, when hierarchical control would not be sufficient to control the rapid state space growth. Finally, the proposed approach can be assessed in larger, more complex systems.

References

Bohus, D., and Rudnicky, A. 2005. Constructing accurate beliefs in spoken dialogue systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 272–277.

Bohus, D., and Rudnicky, A. 2006. A 'K hypothesis + other' belief updating model. In *AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*, 13–18.

Chandramohan, S.; Geist, M.; and Pietquin, O. 2010. Optimizing spoken dialogue management from data corpora with fitted value iteration. In *INTERSPEECH*, 86–89.

Cozman, F. G. 2000. Generalizing variable elimination in Bayesian networks. In *IBERAMIA/SBIA, Workshop on Probabilistic Reasoning in Artificial Intelligence*, 27–32.

Cuayáhuitl, H., and Dethlefs, N. 2011a. Spatially-aware dialogue control using hierarchical rienforcement learning. *ACM Transactions on Speech and Language Processing (Special Issue on Machine Learning for Robust and Adaptive Spoken Dialogue Systems)* 7(3):5:1–5:26.

Cuayáhuitl, H., and Dethlefs, N. 2011b. Optimizing situated dialogue management in unknown environments. In *INTERSPEECH*.

Cuayáhuitl, H.; Renals, S.; Lemon, O.; and Shimodaira, H. 2010. Evaluation of a hierarchical reinforcement learning spoken dialogue system. *Computer Speech and Language* 24(2):395–429.

Cuayáhuitl, H. 2009. *Hierarchical Reinforcement Learning for Spoken Dialogue Systems*. Ph.D. Dissertation, School of Informatics, University of Edinburgh.

Denecke, M.; Dohsaka, K.; and Nakano, M. 2004. Fast reinforcement learning of dialogue policies using stable function approximation. In *International Joint Conference on Natural Language Processing (IJCNLP)*, 1–11.

Heeman, P. 2007. Combining reinforcement learning with information-state update rules. In *Human Language Technology Conference (HLT)*, 268–275.

Henderson, J., and Lemon, O. 2008. Mixture model POMDPs for efficient handling of uncertainty in dialogue management. In *International Conference on Computational Linguistics (ACL)*, 73–76.

Henderson, J.; Lemon, O.; and Georgila, K. 2005. Hybrid reinforcement/supervised learning for dialogue policies from communicator data. In *Workshop on Knowledge and Reasoning in Practical Dialogue Systems (IJCAI)*, 68–75.

Horvitz, E., and Paek, T. 1999. A computational architecture for conversation. In *International Conference on User Modelling (UM)*, 201–210.

Jensen, F. 1996. *An Introduction to Bayesian Networks*. Springer Verlag, New York.

Lecoeuche, R. 2001. Learning optimal dialogue management rules by using reinforcement learning and inductive logic programming. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Lemon, O. 2011. Learning what to say and how to say it: Joint optimization of spoken dialogue management and natural language generation. *Computer Speech and Language*.

Paek, T., and Horvitz, E. 2000. Conversation and action under uncertainty. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 455–464.

Roy, N.; Pineau, J.; and Thrun, S. 2000. Spoken dialogue management using probabilistic reasoning. In *International Conference on Computational Linguistics (ACL)*, 93–100.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education.

Singh, S.; Litman, D.; Kearns, M.; and Walker, M. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of AI Research* 16:105–133.

Skantze, G. 2007. *Error Handling in Spoken Dialogue Systems: Managing Uncertainty, Grounding and Miscommunication*. Ph.D. Dissertation, KTH - Royal Institute of Technology.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Thomson, B. 2009. *Statistical methods for spoken dialogue management*. Ph.D. Dissertation, University of Cambridge.

van Otterlo, M. 2009. *The Logic of Adaptive Behaviour: Knowledge Representation and Algorithms for Adaptive Sequential Decision Making under Uncertainty in First-Order and Relational Domains*. IOS Press.

Williams, J., and Balakrishnan, S. 2009. Estimating probability of correctness for ASR N-Best lists. In *Workshop on Discourse and Dialogue (SIGDIAL)*.

Williams, J. 2006. *Partially Observable Markov Decision Processes for Spoken Dialogue Management*. Ph.D. Dissertation, Cambridge University.

Williams, J. 2008. Integrating expert knowledge into POMDP optimization for spoken dialogue systems. In *AAAI Workshop on Advancements in POMDP Solvers*.

Williams, J. 2010. Incremental partition recombination for efficient tracking of multiple dialog states. In *ICASSP*.

Young, Y.; Gasic, M.; Keizer, S.; Mairesse, F.; Schatzmann, J.; B., T.; and Yu, K. 2010. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language* 24(2):150–174.

Building Modular Knowledge Bases for Conversational Agents

Daniel Macías-Galindo and **Lawrence Cavedon** and **John Thangarajah**
{daniel.macias, lawrence.cavedon, john.thangarajah}@rmit.edu.au
School of Computer Science and Information Technology
RMIT University, Melbourne, Australia

Abstract

The process of constructing domain-specific ontologies presents challenges in the time and human effort required. Although some efforts have been made to automate this process using hierarchical relations, relatively little has been done on incorporating other types of semantic associations between concepts. We describe *MKBUILD*, a tool that follows a methodology to create domain-specific ontologies containing related concepts, drawing from existing large-scale resources such as WordNet and Wikipedia/DBPedia. The context for this work is to provide Modular Knowledge Bases (MKBs) for a *conversational agent* designed to operate on a mobile platform with a small computational footprint. The MKBs that we generate will be utilised by the conversational agent when processing speech fragments and for generating coherent narrative structure for free-flowing conversations. In order to obtain semantic associations between concepts in the ontology that we generate, we use hierarchical relations and *word senses* which are obtained from WordNet and semantic associations between concepts which are obtained from Wikipedia. As an initial evaluation, we ask human participants to rate the relevance of concepts in constructed domain-specific ontologies to the associated domain, using sample ontologies created using our technique. We obtain promising results, as participants consider above 68% of the concepts of sample ontologies to be relevant to the domain.

1. Introduction

We describe a process for constructing a set of ontologies related to a given domain, drawing from existing large-scale resources. The context for this work is to provide a knowledge base (KB) for a *conversational agent* designed to operate on a mobile platform with a small computational footprint. As such, existing large-coverage ontologies and KBs (e.g., WordNet, Cyc) are unviable. Our conversational agent involves a modular architecture (Adam, Cavedon, and Padgham 2010), in which new conversational capabilities can be uploaded to the conversational agent by adding a new *module*, which supports conversations about a specific "domain", such as a visit to the zoo or a day at the beach.

A central component of a conversational module is the ontology and KB of information about the module's domain. The ontology and knowledge base are used to ground references and associate topics with conversation snippets. In particular, *semantic relatedness* (Budanitsky and Hirst 2006) as measured over the ontology is used to select topics to support coherent discourse (Barzilay and Lapata 2008). However, constructing a domain-specific ontology manually is extremely time-consuming.

We describe *MKBUILD*, a tool that, given a *primary domain concept* selected by a module designer, extracts a set of sub-ontologies that contain concepts relevant to such a domain. As we cannot use the large resources described above on our small platform, *MKBUILD* serves to identify the appropriate portions of these resources to extract and associate with a specified domain. We specifically aim to construct ontologies, including hierarchical taxonomic relations: this contrasts to recent work that extracts term-concept networks (Gregorowicz and Kramer 2006) from resources such as Wikipedia.

The first step of our process is to identify the *core root concepts* associated with a given domain. For example, for the domain about a trip to the ***Zoo***, we identify concepts such as ***Animal*** and ***Amusement Park***[1]. We then construct a sub-ontology for each core root concept identified for a domain, using large-scale knowledge resources, including WordNet. Finally, we explore adding non-hierarchical associations between concepts. Such associations will play a role in topic-switching and conversational coherence.

The paper is structured as follows: Section 2 outlines the architecture of our conversational agent; Section 3 describes the ontology identification and extraction processes; Section 4 presents an evaluation of *precision* of the extracted "root" concepts according to human evaluators, as well as a discussion of the results obtained; finally, Section 5 discusses conclusions and future work.

2. Conversation Management

The setting for this work is an interactive Toy, a specific incarnation of a conversational agent. The Toy interacts with users via spoken language using automated speech recognition (ASR) and speech synthesis (text-to-speech, TTS).

[1]Note, "concepts" for us will basically equate to WordNet synsets and Wikipedia article titles, but will generally be named and referred to by common terms.

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 16–23,
Barcelona, July 2011

Simple robust techniques such as keyword-spotting and lightweight semantic parsing are used to determine enough user input to (hopefully!) select an appropriate conversational path to follow.

ASR output is sent to the *Dialogue Manager (DM)*, which constructs an appropriate response to send to the TTS. There are two interesting aspects of the DM architecture for the purposes of this paper: (1) it is *modular*, in that its capabilities can be extended to new domains and tasks; and (2) it includes a *Conversation Manager (CM)* that combines activity-oriented conversations (e.g., telling a story, discussing a trip to the zoo) and less structured chat. In particular, the CM must handle diversions from an activity and decide whether it is appropriate to steer the user back to the original conversation flow.

Central to the CM is the notion of *Activity* and *Conversation Agenda*. An Activity is a conversational task, such as "telling a (particular) story" or "talking about your day at school". A module provides the Toy with fragments that allow it to run one or more Activities. The Conversation Agenda contains the current set of selected Activities and their status: e.g., an Activity may be suspended by a change in topic and resumed later.

The content of conversations is defined by *Conversational Fragments*, which are short authored snippets of conversation (Adam, Cavedon, and Padgham 2010). Fragments are created by the designer of a topic module; each fragment is associated with one node in a *Topic Network*, that is generated from the domain ontology. Each fragment consists of:

- A *head* containing unique id, topic, type and *applicability condition*;
- A *body* containing output utterance and a list of *expected inputs* coupled with associated processing (e.g., set internal variables).

Each expected input may use a very generic template designed simply to check for certain keywords to allow maximum coverage. For example, a *quiz*-activity fragment (*type=activity*, *subtype = quiz*) on the *lions* topic of the *zoo* module, with the output sentence *"What do lions eat?"*, will expect a number of appropriate answers (*"meat"*, *"zebras"*,...) and a number of wrong answers (*"grass"*,*"lollies"*,...) with different processing and/or responses associated with each. A story fragment has an output utterance representing one line of the story, with the *applicability condition* ensuring that lines of the story are told in order; expected inputs could be questions that may be asked about that line of the story.

Our framework is designed to support both open-ended conversation and activity-oriented dialogue; it is also designed for system-led (as opposed to user-led) dialogues, while still allowing the user to interrupt (e.g., ask a question) or divert topic. The purpose of the Topic Transition Network is used to generate a coherent dialogue structure. This is similar to the use of ontology-based measures of semantic similarity in measuring discourse coherence (e.g., (Lapata and Barzilay 2005)), but used "generatively" in deciding which conversational fragments to be selected for the next portion of conversation.

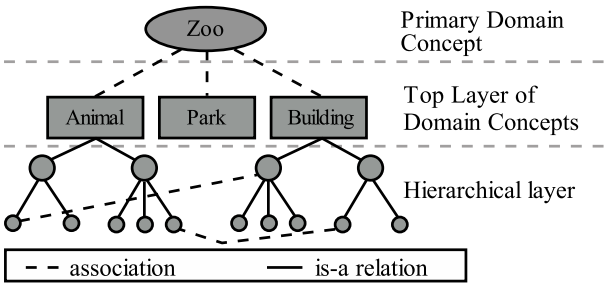

Figure 1: Schema of a MKB.

As mentioned above, the Topic Transition Network is constructed according to the domain ontology featured in the module. However, representing a domain seldom involves a single hierarchical ontology[2]. In the next section, we describe how such a resource, which we call a Modular Knowledge Base (MKB), is constructed using a (semi-)automated process.

3. Building Domain-Specific MKBs

In this section we describe the methodology and details of the steps in building a domain-specific ontology, or MKB.

An MKB is built around a main concept representative of the domain, and features a set of sub-ontologies linked by associations amongst its nodes (concepts), e.g., ***Zoo*** when building an MKB for the zoo domain. Linked to this main concept, several other concepts form a top layer that features the most general concepts associated with the domain; e.g., ***Animal*** and ***Park*** for the ***Zoo*** domain. For each concept at the top layer, we obtain a set of concepts hierarchically related to it, for example, ***Mammal*** and ***Reptile*** for the concept ***Animal***. Some parts of the domain require hierarchical relations; for example, when building a MKB about a ***Zoo***, it is relevant to capture the hierarchy below the concept ***Animal***. In addition to hierarchical relations, the MKBs we develop also feature other concept associations; these may be used for topic transitions by the conversational agent. The target architecture for a MKB is shown in Figure 1.

To build MKBs, we use two information resources: (a) WordNet (Fellbaum 1998), a lexical ontology that contains multiple word senses, grouped by their meaning; and (b) Wikipedia[3], an online encyclopaedia that operates like a collaborative wiki. An example of both resources is presented in Figure 2.

WordNet features a hierarchical structure of concepts, but lacks other kind of relationships that are not lexical (e.g., ***Lion*** `lives in` ***Savannah***). On the other hand, Wikipedia features a set of *wikilinks* in every article. Each *wikilink* links to a concept that helps in understanding a definition[4]. *Wikilinks* can be used as associations between con-

[2]We do link the multiple ontologies obtained below into a single ontology rooted at the domain concept. But this is irrelevant to the processes we describe.

[3]http://en.wikipedia.org/

[4]See http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

cepts, although they do not always describe a positive association: for example, the article about spiders has a link to insects, although their relationship is negative, in the sense that spiders are *not* insects. At the moment we are not interested in the type of association but the existence of such associations. The combination of Wikipedia *wikilinks* ("flexible" in the sense that humans themselves choose what to link in Wikipedia articles) and the WordNet hierarchy ("rigid" because property inheritance cannot be changed by humans) helps us produce richer MKBs.

Wikilinks have been previously analysed as a reliable set of associations between articles (Hepp, Siorpaes, and Bachlechner 2007; Milne, Medelyan, and Witten 2006). (Milne, Medelyan, and Witten 2006) used Wikipedia to compare its coverage against *Agrovoc*, an agriculture thesaurus. To represent associations between concepts, they initially used only *wikilinks* that are mutual or bidirectional; however, coverage of important associations was almost doubled when *unidirectional wikilinks* were also considered. In our approach, we use unidirectional wikilinks as a first step since, as commented earlier, we are interested in using such associations for topic transitions, rather than for determining strict semantic relatedness.

We now describe the process for creating an MKB, which provides the designers with a starting point, which they can then refine, rather than having to manually build the MKB from nothing.

The process of producing domain-specific MKBs consists of the following 3 stages:

1. **Define the *primary domain concept***. To achieve this, manual exploration of Wikipedia articles for an unambiguous concept that describes the domain of interest is required.
2. With a *primary domain concept* selected, the next stage is to **extract the top layer of concepts**. This layer represents the most general and representative concepts that can be associated with the *primary domain concept*.
3. Finally, we **extend the domain and associate its concepts** by adding sub-concepts to each top layer concept and analysing, for each concepts' articles, their corresponding *wikilinks*.

An overview of the process can be seen in Figure 3. The first stage of the process is performed manually by the module designer. For the next two stages, we have developed a tool called *MKBUILD* that performs all the tasks necessary for those stages and produces a MKB *automatically*. *MKBUILD* has been developed in Java and uses the OWL-API Library[5] for handling the ontology.

All these stages may be performed separately using *MKBUILD*, thus allowing intermediate manual modifications to the MKB in order to improve the coverage of the module. This means that the process described below may be performed in separate steps; for example, constructing the top layer of concepts (Stage 2) may be verified and edited by the designer before Stage 3. However, for the examples provided below no manual modifications have been performed.

[5]http://owlapi.sourceforge.net/

Figure 2: Screens of (a) WordNet and (b) Wikipedia. ***Lion*** features four senses in WordNet, each with multiple word forms (underlined); a Wikipedia article contains a set of page links (underlined).

We now describe each stage in detail.

Stage 1. Defining the *primary domain concept*

This stage requires the most interaction and supervision from the module designer. Here, the designer must find a word or phrase that describes the domain that the MKB will be about. We require that an article exists in Wikipedia that matches such a phrase, so that the domain concepts that form the top layer as described above may be extracted.

Finding such a phrase can sometimes be complex, due to multiple word senses that such a phrase can carry. For example, Figure 2.a shows **lion** and four possible senses that share the same word form. Wikipedia deals with word senses via *disambiguation pages*: these are pages that contain different contexts of a given word; for example, the "Museum" disambiguation page has links to the page about the facility where objects are exhibited, as well as to a song, a subway station and other places or streets around the world. Note, however, that articles in Wikipedia have a unique identifier: e.g., "Museum", "Museum (song)", "Museum (TTC)", respectively. Such unique identifiers are important because they are used in both Wikipedia and DBPedia to name articles; the detection of *wikilinks* in the following stages will require such full names. The word sense disambiguation task is alleviated in the automatic process described below in Step 2.3. Depending on the sense that the designer wants to use to build the

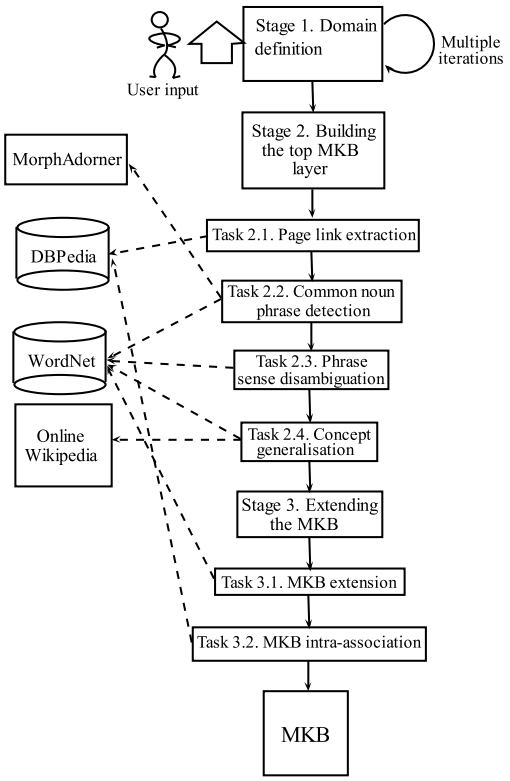

Figure 3: An overview of the process to build MKBs.

MKB, the right identifier must be selected. This identifier is referred below as the *primary domain concept*.

Stage 2. Building the MKB top layer

The *primary domain concept* that is identified in the previous stage is used as the input to *MKBUILD*, which performs the steps in Stages 2 and 3 *automatically*. In Stage 2, the concepts that form the top layer of the MKB are extracted. These concepts should be directly associated with the *primary domain concept*.

MKBUILD first extracts the set of *wikilinks* from the article relevant to the *primary domain concept* in Wikipedia. It then adds concepts to the MKBs that: (a) contain common nouns in their name; (b) are unambiguous; and (c) are the most general that can be associated to the domain. This process is described below.

As we describe the MKB creation process, we illustrate it with the *primary domain concept* ***Museum*** (the facility that exhibits objects).

Task 2.1. Page links extraction. *MKBUILD* searches for all terms that have *wikilinks* included in the article to which the *primary domain concept* refers to. Rather than extracting such terms directly from Wikipedia, *MKBUILD* uses DBPedia (Auer et al. 2008) (version 3.5.1).

DBPedia features the information of Wikipedia in separate files; therefore, extracting information from it is faster than parsing Wikipedia. In this case, *wikilinks* are represented as triples of the form $< article\_source >< wikilinks >< article\_dest >$. *MKBUILD* extracts all the $article\_dest$ terms that $article\_source$ links to. Following our example, *MKBUILD* obtains all the terms that the "Museum" article has *wikilinks* to, such as "Preservation (library and archival science)", "Collection (museum)", and others.

We also extract not only the terms that the article of the *primary domain concept* links to, but also any *redirect links* that accompany each term. Redirect links are used in Wikipedia to reconcile different naming representations of the same article that have been used by its multiple collaborators. A redirect link may exist for a new Wikipedia entry for an existing defined concept, i.e., an existing article. Redirect links are stored in the form $< string >< redirects >< article >$ in DBPedia; we use these to obtain the correct (existing) concept name.

Previous work considered the category structure provided by Wikipedia to be a reliable means for detecting related concepts (Grieser et al. 2011; Herbelot and Copestake 2006; Ponzetto and Strube 2007). However, we have found that categories involve associations that might not be appropriate to a domain: for example, one category for "museum" is "Greek loanwords", which is related to the *word* "museum" but not the *concept* of interest. We do, however, use Wikipedia's category-based hierarchical *folksonomy*, to extend the set of extracted candidate concepts: any concept that is assigned a category c that is also a category for the *primary domain concept* is added as a candidate related concept. This includes the case where c is a super-category of the candidate concept or the *primary domain concept*. We see this as a simpler approach to Grieser et al's RACO criterion (Grieser et al. 2011; Newman et al. 2010).

At the end of this Stage, *MKBUILD* obtains a set of *preliminary concept terms*.

Task 2.2. Common noun term detection. Some of the preliminary concept terms refer to concepts, but some also refer to *instances* of concepts (e.g., specific people or places), as Wikipedia itself does not distinguish between concepts and instances (Hepp, Siorpaes, and Bachlechner 2007). The next step in *MKBUILD* is a process that distinguishes concepts from instances.

The proposal is to only retain terms that are named by common nouns (or compound terms that do not contain proper nouns nor adjectives). To obtain such terms, *MKBUILD* uses a two step process: (i) using a Part-of-Speech (POS) tagger implemented in the Language Technology tool MorphAdorner[6]; and (ii) using WordNet to detect word senses. Prior to this process however, we perform two refinements to the preliminary terms: (a) all terms are changed to be in lower-case –initial experiments showed that MorphAdorner, due to the lack of context[7], would often tag terms incorrectly as proper nouns; and (b) any additional information in the name of terms was removed, such as special characters or words in parentheses, since WordNet cannot handle them accompanying the name.

[6] http://morphadorner.northwestern.edu/

[7] MorphAdorner usually parses sentences.

(i) *POS tagging:* Each term in the refined preliminary terms list is passed by *MKBUILD* through MorphAdorner's POS tagger. Terms are retained for the next step as long as they: (a) contain at least one common noun; and (b) do not feature proper nouns, proper adjectives nor non-English words (e.g., "Musaeum").

(ii) *Mapping to WordNet concepts:* In the second step, *MKBUILD* matches the terms retained in the first step to existing senses in WordNet. For each term, *MKBUILD* verifies that the term exists in WordNet, where at least one sense of it is a noun. If this holds, the next step is to detect if all senses of the term correspond to concepts rather than to instances. To do this, we follow Martin's approach (Martin 2003) whereby instances are detected if the word forms associated to a sense begin with an uppercase letter. Additionally, we leverage DBPedia further for detecting instances. DBPedia provides a useful resource which links Wikipedia articles (or DBPedia concepts) that represent instances to their corresponding WordNet sense. For example, *Twitter* is related to the first sense of the word **website**. This DBPedia file contains triples $< article >< wordnet\_type >< sense >$, from which we detect proper nouns. This approach lets us effectively distinguish certain instances that are named using common nouns, such as *Twitter*.

Task 2.3. Term sense disambiguation. The previous step produces a set of terms with at least one sense in WordNet. Some of the terms may have multiple senses. Consequently, a disambiguation process is required by *MKBUILD* to determine the best sense to be associated with the domain. This is an important step because each sense brings different subconcepts to the MKB. Additionally, there is no text to use as context for disambiguation (since we do not perform text analysis over Wikipedia). This restricts the techniques that can be used for this task.

To address this problem, we use the semantic similarity measure of (Lesk 1986) adapted to WordNet, which finds overlaps between super-classes of each pair of input terms[8]. The algorithm for the disambiguation process is as follows:

```
P ← set of terms
S ← set of unique sense word forms (init. empty)
M ← set of multiple senses word forms (init. empty)
For each term p in P do:
    - Detect how many senses p has in WordNet
    - If senses(p) = 1, add p to S.
    - Else, add p to M.
End For
For each word form m in M do:
    - For each word form s in S do:
        * Compare semantic similarity of each sense
        of m and s
        * Retain the most similar sense of m for s
    - End for
    - Pick the most similar sense of m for all s, and
    add it to S
End for
```

This method requires at least one unambiguous term to seed the process; in practice, this has always been the case: if it were not, user intervention would be used to select a seed term and sense. From the algorithm, it can be observed that each disambiguated multiple-sense term is added to the next disambiguation process as a new unique-sense term. We have found that, for all the tested domains, one sense always overwhelms the others, and the algorithm always terminates[9]. However, in the case that all the candidate concepts have multiple senses, then *MKBUILD* asks the designer to disambiguate one sense, then the process automatically continues for the remaining senses.

To illustrate the process using the ***Museum*** MKB, examples of unique-sense terms are ***Curator*** and ***Natural History***. Similarly, examples of multiple-sense terms are ***Collection*** and ***Sculpture*** (4 and 2 senses, respectively). By applying semantic similarity of multiple-sense and unique-sense terms, *MKBUILD* retrieves the senses "several things grouped as a whole" and "a three-dimensional work of plastic art", respectively, which are relevant to the domain.

Task 2.4. Concept generalisation. Task 2.3 above results in a set of unambiguous concepts. However, in some cases, it is possible that they do not yet represent the level of generality required for the domain (i.e. the domain covers more general concepts than those identified). Due to not being able to find suitable examples of generalisation in the ***Museum*** MKB, we use other examples to illustrate this task. For example, a designer building an MKB about ***Zoo*** may require the concept ***Animal***; however, Wikipedia does not contain this word as a wikilink in the *zoo* related article; rather it contains wikilinks to specific animals, such as *lemur*, *whale* and *marmoset*.

Generalising may lead to concepts being included that are not relevant to a domain, since not every concept in a hierarchy is appropriate—e.g., there are (typically) no chickens in zoos so the ***Animal*** concept is not strictly appropriate. Such concepts can be manually deleted if the module designer feels they should be. In some cases, however, whole subhierarchies are inappropriate and the generalisation should be stopped. For example, an MKB with ***Aquarium*** as the *primary domain concept* should not contain the ***Animal*** hierarchy. In this case, only fish, seals and some birds are relevant to the domain. Therefore, the generalisation must include an appropriate "stopping condition" in order to produce relevant results that require less manual editing.

Concept generalisation is performed by *MKBUILD* using two steps: (i) *generalisation using the initial set of concepts*; and (ii) *generalisation using the extended set of concepts*.

(i) *Generalisation using the initial set of concepts:* To start this process, all *hypernyms* (super-concepts in WordNet) of each concept in the concept list are extracted from WordNet and added to the list of terms. In this step, *MKBUILD* also removes concepts subsumed by another concept in the list, as they will be later added as sub-concepts of the corresponding top-layer concept at a later stage described ahead.

(ii) *Generalisation using the extended set of concepts:* In this step, *MKBUILD* searches for common super-classes between concepts of the list. The super-classes are extracted

[8]This measure is implemented using the Java WordNet:Similarity Library: http://www.cogs.susx.ac.uk/users/drh21/.

[9]This can be seen in the appendix that the authors have placed online at http://mkbuild.wikispaces.com/experiment

from WordNet, and the step consists of exploring level by level such super-classes, starting from the direct parent classes. To apply concept generalisation, we use *concept majority* as follows.

If a common super-class occurs between at least two concepts of the list, then *MKBUILD* searches Wikipedia for the number of articles that contain the *primary domain concept* pdc and either the common super-class h or the set of concepts $t_1, ...t_n$ that are sub-classes of h. Let $wiki\_art\_count(x, y)$ be the number of Wikipedia articles that contain both terms x and y. In order for h to be chosen as a general concept the following must apply:

$$wiki\_art\_count(pdc, h) > \sum_{k=1}^{n} wiki\_art\_count(pdc, t_k)$$

In other words, concept h is more representative (or more commonly associated) to the domain rather than its detected sub-classes. In this case, all concepts $t_1, ...t_n$ are removed from the top list of concepts, and h is added to this list.

This step is illustrated in figure 4 via the two examples mentioned above. In (a), ***Zoo*** represents the *primary domain concept*, whereas concepts ***Chicken***, ***Lemur***, ***Marmoset*** and ***Whale*** are extracted from the ***Zoo*** article. By exploring the hierarchy in WordNet, *MKBUILD* detects that ***Animal*** is a common super-class that may replace these four concepts. The indexes shown beside each concept represent the number of Wikipedia articles that contain the *primary domain concept* and the corresponding concept. Since the sum of the number of articles containing ***Zoo*** and all sub-classes (1524) is not greater than the articles for ***Zoo-Animal*** (5977), *MKBUILD* maintains ***Animal*** and removes its sub-classes. On the other hand, for the MKB about ***Aquarium*** shown in (b) *MKBUILD* will not generalise concepts ***Bird***, ***Dolphin***, ***Fish*** and ***Pinniped*** (the group that is comprised of *seals* and *walruses*), since the sum of articles containing the *primary domain concept*and those concepts (3697) is greater than that for ***Animal*** (2550).

These two steps are repeated until the number of concepts in the list is not reduced in a full iteration (i.e. no reduction is performed over the set of concepts).

Once the generalisation process is applied, the remaining set represents the top-layer concepts of the MKB. Their association to the domain concept is not necessarily hierarchical; however, there may occur some hierarchical relations between the *primary domain concept* and a concept in this layer, which are added. For non-hierarchical associations, *MKBUILD* will create a type of association called `wikilink` between the domain concept and each concept in the top-layer of the MKB, hereafter named the *top-layer domain concepts*.

Stage 3. Building the MKB ontologies

With the top-layer of concepts obtained from Stage 2, two more tasks are performed before an initial version of the MKB is produced. In the first step, sub-classes of the top-layer concepts are extracted from WordNet and attached. In the second step, the resulting concepts of the MKB are

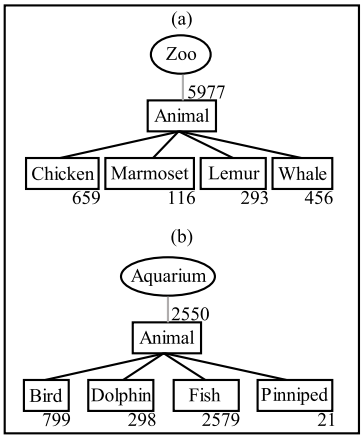

Figure 4: Detection of more general classes via WordNet: (a) shows when the generalisation rule is applied, whereas (b) shows when it is not.

associated with each other using any *wikilink* associations present in their corresponding Wikipedia articles.

Task 3.1. MKB hierarchy extension. For each concept in the top-layer, all of its sub-classes (according to the WordNet hierarchy) are added to the MKB to extend the MKB. In comparison to the previous steps described, this step is straightforward, since ambiguity and generality have been resolved in Stage 2. Nevertheless, as in part (i) of step 2.2, *MKBUILD* takes only WordNet senses that are common nouns as concepts to include in the MKB (i.e. their names contain only lowercase letters). After this step, the MKB contains a set of sub-ontologies associated to the *primary domain concept* via the top-layer concepts.

Step 3.2. MKB concept-association. WordNet proposes a set of hierarchical relations between concepts, as well as other lexical relations (for example, *holonymy* and *meronymy*), that refer to membership of a concept (e.g., ***Museum*** `has part` ***Storage space***). While these relations are useful in determining semantic relatedness between concepts, there are other kinds of associations that are used to identify links between concepts (Budanitsky and Hirst 2006). Since we expect our agent to take control of conversations, it is important to have multiple possible topics as candidates for future conversations. These kind of associations however, are not available in existing knowledge repositories. It has been demonstrated that *wikilinks* show some sort of association between concepts (Hepp, Siorpaes, and Bachlechner 2007); therefore, we use these to derive associations between each available concept.

For this step, *MKBUILD* extracts all the concepts in the MKB. Then, for each pair of concepts c_i and c_j in the MKB, it searches if the wikilink $< c_i >< wikilinks >< c_j >$ exists in DBPedia. If it does, a `wikilink` association between these two concepts is added to the MKB (as long as there is not a hierarchical relation already in place). This process may be improved by using Wikipedia *redirect* links,

for cases where articles are not named after a given MKB concept.

4. Preliminary Evaluation

In this section, we describe a preliminary evaluation of Stage 2 of *MKBUILD*, i.e., identifying the top-layer domain concepts of the sub-ontologies related to the specified domain[10]. We focus on evaluating *precision*: i.e., are the concepts identified in Stage 2 actually valid in that they are pertinent to the specified domain (evaluating *Recall* is discouraged since we do not have a closed set of concepts to choose from). We performed a user study for this, asking users to judge whether the top-layer domain concepts extracted by *MKBUILD* (i.e., in Stage 2) were appropriate to the domain. We only evaluate the precision of top-layer domain concepts and not the hierarchy below, since the concepts in the hierarchy below a top-layer domain concept are assumed to be related to it.

4.1. Setup

We used *MKBUILD* to construct MKBs for three sample domains: ***Internet***, ***Zoo***, and ***Museum***; these contained 56, 13, and 34 top-layer domain concepts and their corresponding hierarchies, respectively. See Table 1 for a list of some top-layer domain concepts that were used in the experiment.

We asked then 8 subjects (6 of them with a postgraduate background in Computer Science) to rate, for each top-layer domain concept in each proposed domain D, whether each concept was related to D. Users scored each domain-concept pair with an integer number, either 1, -1 or 0, where: a score of 1 indicated that the concept is related to D, -1 indicated not related, and 0 indicated don't know/uncertain. We post-processed the data to remove subjects with high levels of uncertainty ($> 20\%$ scores of 0)[11]: for ***Internet*** we retained scores from 7 subjects; for ***Zoo*** and ***Museum***, scores from 4 and 5 subjects were retained respectively. We computed the average Pearson correlation for the retained subjects' scorings of the concepts of each domain to check for reliable agreement between subjects: for *Internet* we obtained a Pearson score of 0.558 (p-value=0.01 for 20 out of 21 combinations) indicating "strong correlation"; for *Zoo* we obtain 0.458 (p-value=0.05 for 2 out of 6 combinations) indicating high "medium correlation"; for *Museum* we obtain 0.178 (p-value=0.05 for 2 out of 10 combinations) indicating low "medium correlation". Given these Pearson scores, we consider the subjects' ratings for *Internet* and *Zoo* concepts as reliable.

4.2. Results

For each concept, we aggregated all (retained) judgements for that concept; e.g., if 3 subjects scored a concept as related to its domain, 1 subject scored the concept as unrelated, and 1 was "uncertain", then the aggregate score is 2. Given the scoring system, an aggregate score greater than zero for a concept indicates more subjects rating the concept as related to its domain than indicating it as unrelated; a negative score indicates the converse. Those concepts that were overall judged positive (i.e., score greater than 0) were considered to be correct extractions; this means that the majority of subjects agree with the system. See Table 1 for some examples with their corresponding aggregation score.

Domain	Concept	Aggregate score
Internet(7)	Modem	7
	Blog	7
	URL	6
	University	-1
	Radar	-6
Zoo(4)	Animal	4
	Zoology	3
	Bamboo	2
	Brazil	-1
	State	-4
Museum(5)	Art	5
	Craft	4
	Musician	3
	Lion	0
	Lake	-2

Table 1: Some examples of top-layer domain concepts. The number of participants retained for each domain is shown in parenthesis.

For domain ***Internet***, we found that subjects agreed that concepts extracted by the system were related to that domain for 38/56 (68%) of the concepts. For ***Zoo***, subjects agreed that 10/13 (77%) of the extracted concepts were related to the domain. For ***Museum***, subjects agreed that 30/34 (88%) of the extracted concepts were relevant. While it cannot be compared directly, our task has some similarity to the *domain-term extraction* task (Liu et al. 2010). For this task, (Liu et al. 2010) reported a precision of 35.4%, which reflects encouraging results using our approach[12].

4.3. Discussion

This evaluation is obviously a preliminary one, however, the above precision scores are promising, although indicate a need for improvement. Concepts extracted by *MKBUILD* for the domain ***Internet*** but deemed not relevant by a majority of subjects for ***Internet*** include other communication technologies, such as ***Television***, ***Radio***, ***Radar***, and ***Compact Disc***.

For ***Zoo***, one of the concepts judged "irrelevant" ***State***. The actual sense selected by *MKBUILD* corresponded to "the way something is with respect to its main attributes", in this case, the state of being of animals. However, the users' judgements for this example were likely clouded by choice of incorrect word sense. Future evaluations will provide clearer directions to users to try to avoid such problems.

This preliminary evaluation also does not give us information on *coverage*, in other words, whether *MKBUILD* ex-

[10]Evaluating other stages would be effectively evaluating WordNet and DBpedia.

[11]Note, we did this only to remove subjects who seemed to have difficulty with the task. This was done to make the judgements more reliable, not to bias them to favourability.

[12]The constructed MKBs are available in the appendix at http://mkbuild.wikispaces.com/MKBs.

tracted all concepts relevant to a given domain. This evaluation neither address the issue of whether the root concepts were extracted at an appropriate level in the hierarchy; i.e., if more general concepts may have been more appropriate as roots of relevant concept hierarchies. We are currently in the process of collecting more extensive judgements from more users that will provide a more comprehensive evaluation of *MKBUILD* (and subsequent extensions of it).

Note that the process is intended to be semi-automatic: as pointed out earlier the module designer can intervene at each Stage of *MKBUILD* and edit the output. In particular, for each of the example domains it is a fairly simple and quick process to remove any inappropriate concepts introduced during the Stage 2 process.

Conclusions and future work

We have described a process for constructing domain-specific ontologies, called Modular Knowledge Bases, to be used by a conversational agent with a modular infrastructure. The process has been programmed in *MKBUILD*, a tool that allows *automatic* extraction of concepts and relations, specific to a given domain, from large resources such as WordNet and Wikipedia/DBPedia. This ontology-construction process we have described saves the module designer significant effort in constructing an ontology specific to a conversational domain, but allows them to intervene at various steps to correct any egregious errors.

A major purpose of the domain ontologies is to generate a *Topic Transition Network* that is used to link conversational fragments together into more coherent longer-running threads, using ontology-based semantic similarity measures. Further developing this technique is a topic for future work.

We have described a preliminary evaluation of the precision of the critical stage of the domain-specific process, i.e., identifying the top-level concepts for specific domains. We are currently performing a more significant evaluation that includes measuring the coverage of the concept-extraction process. We also plan to measure ontology-based semantic relatedness involving sets of relations that go beyond previously considered (i.e., as in (Budanitsky and Hirst 2006; Ponzetto and Strube 2007)), and evaluate its efficacy in topic-transitions in conversational dialogue.

The approach described here is currently limited in coverage to WordNet concepts; future work will investigate extending coverage beyond such concepts, as well as including concepts, relations, and entries from extensive knowledge bases constructed using information extraction techniques using language technologies (e.g., (Yates et al. 2007)).

Acknowledgements: The first author acknowledges scholarship 201228 from the Consejo Nacional de Ciencia y Tecnologia (CONACYT), Mexico. This work was partially supported by Australian Research Council Linkage Project LP0882013. We acknowledge the support and collaboration of our partner, Realthing Pty. Ltd. We also thank the anonymous referees for valuable feedback.

References

Adam, C.; Cavedon, L.; and Padgham, L. 2010. Flexible conversation management in an engaging virtual character. In *Workshop on Interacting with ECAs as virtual characters*.

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2008. DBpedia: A Nucleus for a Web of Open Data. In *The Semantic Web*. Springer.

Barzilay, R., and Lapata, M. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics* 34(1):1–34.

Budanitsky, A., and Hirst, G. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics* 32(1):13–47.

Fellbaum, C. 1998. *WordNet: an electronic lexical database*. The MIT Press.

Gregorowicz, A., and Kramer, M. A. 2006. Mining a Large-Scale Term-Concept Network from Wikipedia. Technical Report 06-1028, The MITRE Corp.

Grieser, K.; Baldwin, T.; Bohnert, F.; and Sonenberg, L. 2011. Using Ontological and Document Similarity to Estimate Museum Exhibit Relatedness. *ACM Journal on Computing and Cultural Heritage (JOCCH)*.

Hepp, M.; Siorpaes, K.; and Bachlechner, D. 2007. Harvesting Wiki Consensus: Using Wikipedia Entries as Vocabulary for Knowledge Management. *IEEE Internet Computing* 11(5).

Herbelot, A., and Copestake, A. 2006. Acquiring Ontological Relationships from Wikipedia Using RMRS. In *ISWC06 Workshop on Web Content Mining with Human Language Technologies*.

Lapata, M., and Barzilay, R. 2005. Automatic Evaluation of Text Coherence: Models and Representations. In *IJCAI*.

Lesk, M. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *SIGDOC*.

Liu, Z.; Huang, W.; Zheng, Y.; and Sun, M. 2010. Automatic Keyphrase Extraction via Topic Decomposition. In *EMNLP*.

Martin, P. 2003. Correction and extension of WordNet 1.7. In *International Conference on Conceptual Structures*.

Milne, D.; Medelyan, O.; and Witten, I. H. 2006. Mining Domain-specific Thesauri from Wikipedia: A case study. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*.

Newman, D.; Lau, J. H.; Grieser, K.; and Baldwin, T. 2010. Automatic evaluation of topic coherence. In *NAACL HLT*.

Ponzetto, S. P., and Strube, M. 2007. Knowledge Derived From Wikipedia For Computing Semantic Relatedness. *Journal of AI Research* 30(1).

Yates, A.; Cafarella, M.; Banko, M.; Etzioni, O.; Broadhead, M.; and Soderland, S. 2007. TextRunner: open information extraction on the web. In *NAACL-Demos*.

SceneMaker: Visual Authoring of Dialogue Processes

Gregor Mehlmann
Human Centered Multimedia
Augsburg University
Universitätsstrasse 6a
D-86159 Augsburg

Patrick Gebhard
DFKI GmbH
Campus D3 2
Stuhlsatzenhausweg 3
D-66123 Saarbrücken

Birgit Endrass
Human Centered Multimedia
Augsburg University
Universitätsstrasse 6a
D-86159 Augsburg

Elisabeth André
Human Centered Multimedia
Augsburg University
Universitätsstrasse 6a
D-86159 Augsburg

Abstract

In this paper, we present a visual authoring approach for the management of highly interactive, mixed-initiative, multi-party dialogues. Our approach enforces the separation of dialog-content and -logic and is based on a statechart language enfolding concepts for hierarchy, concurrency, variable scoping and runtime history. These concepts facilitate the modeling of dialogs for multiple virtual characters, autonomous and parallel behaviors, flexible interruption policies, context-sensitive interpretation of the user's discourse acts and coherent resumptions of dialogues. It allows the real-time visualization and modification of the model to allow rapid prototyping and easy debugging. Our approach has successfully been used in applications and research projects as well as evaluated in field tests with non-expert authors.

1 Introduction

Virtual characters in interactive applications can enrich the user's experience by showing engaging and consistent behavior. To what extent virtual characters contribute to measurable benefits is still fiercely discussed (Heidig and Clarebout 2010; Miksatko, Kipp, and Kipp 2010). Therefore, virtual characters need to be carefully crafted in cooperation with users, artists and programmers. The creation of interactive virtual characters with a consistent and believable dialogue behavior poses challenges such as modeling personality and emotion (Marsella and Gratch 2006), creating believable facial expressions, gestures and body movements (Kipp et al. 2007), expressive speech synthesis (Schröder 2008) and natural language recognition as well as dialog and interaction management (Traum et al. 2008). In this work, we address the tasks of modeling consistent highly interactive mixed-initiative multi-party dialogue behavior and realizing effective interaction management for dialogue situations with embodied conversational characters.

During the last years, several approaches for modeling interactive dialogue behavior of virtual characters have been researched. A variety of systems such as, frame-, plan-, rule- and finite state-based systems were presented. Most of these systems required a substantial degree of expert knowledge and programming skills, thus, being unserviceable for non-computer experts, such as artists and screenwriters that wanted to craft interactive applications with virtual characters. Therefore, as a next step, authoring systems were developed to exploit related expert knowledge in the areas of games, film or theater screenplay.

These systems are created to facilitate the authoring process and to allow non-computer experts to model believable natural behavior for virtual characters. They can be categorized by their conceptual and methodological approaches. On the one hand, character-centric approaches aim on creating autonomous agents for multi-agent systems, while they do not explicitly include support for scripting the behavior of multiple agents in a simple and intuitive way. Examples for character-centric systems are *Improv* (Perlin and Goldberg 1996) or *Scream* (Prendinger, Saeyor, and Ishizuka 2004), where an author defines the agents' initial goals, beliefs and attitudes. These mental states determine the agents' behavioral responses to received communicative acts. In author-centric approaches, on the other hand, a human author can communicate an artistic vision with the primary focus of scripting at the plot level. The user can contribute to the plot within the narrative boundaries defined by the author. Examples for author-centric systems include *Scenejo* (Spierling, Weiss, and Mueller 2006), *Deal* (Brusk et al. 2007) and *Creactor* (Iurgel et al. 2009). Hybrid approaches, as described in (McTear 1998; Gandhe et al. 2008) or (Gebhard et al. 2003), try to bridge the gap between the author-centric and character-centric approach by combining the advantages of both.

So far, none of the mentioned authoring systems supports concepts for dialogue and interaction history and concurrent process modeling for parallel behavior on the authoring level. However, this would facilitate the modeling task and reduce the complexity of the model. It would help to handle typical challenges in the creation of applications with interactive virtual characters, such as the modeling of reactive and deliberate behavior, the use of multiple virtual characters and their synchronization and the handling of user interaction. In this paper, we face these challenges using our new version of the authoring tool *Scenemaker* which pursues a hybrid approach to contribute on the user modeling level for the creation of interactive virtual character applications in a

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 24–36,
Barcelona, July 2011

rapid-prototyping style.

2 Dialogue and Interaction Management

The central concept of our authoring approach with the Scenemaker authoring tool is the seperation of dialogue content and structure. Multimodal dialogue content is specified in a set of *scenes* that are organized in a *scenescript*. The narrative structure of an interactive performance and the interactive behavior of the virtual characters is controlled by a *sceneflow* - a statechart variant specifying the logic according to which scenes are played and commands are executed. Sceneflows have concepts for *hierarchical refinement* and the *parallel decomposition* as well as an exhaustive *runtime history* and multiple *interaction policies*. Thus, sceneflows adopt and extend concepts that can be found in similar statechart variants (Harel 1987; von der Beeck 1994).

Sceneflows and scenescripts are created using a graphical authoring tool and executed by an interpreter software. This allows the real-time extension and modification of the model and the direct observation of the effects without the need for an intermediate translation step. The real-time visualization of a sceneflow's execution and active scenes within the graphical user interface allows to test, simulate and debug the model.

2.1 Creating Multimodal Dialogue Content

A scene resembles the part of a movie script consisting of the virtual characters' utterances containing stage directions for controlling gestures, postures, gaze and facial expressions as well as control commands for arbitrary actions realizable by the respective character animation engine or by other external modules. Scenescript content can be created both manually by an author and automatically by external generation modules. The possibility to parameterize scenes may be exploited to create scenes in a hybrid way between fixed authored scene content and variable content (Figure 1 ①), such as retrieved information from user interactions, sensor input or generated content from knowledge bases. In Section 3.5 we present an application which makes extensive use of parameterized scenes and generated scene content from a domain knowledge module.

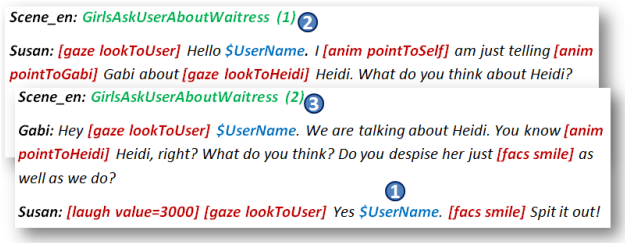

Figure 1: Parameterizable scenes of a scenegroup.

A scenescript may provide a number of variations for each scene that are subsumed in a *scenegroup*, consisting of the scenes sharing the same name or signature (Figure 1 ②,③). Different *blacklisting strategies* are used to choose one of the scenes from a scenegroup for execution. This mechanism increases dialogue variety and helps to avoid repetitive behavior of virtual characters, which would certainly impact the agents' believability.

2.2 Modeling Dialogue Logic and Context

A sceneflow is a hierarchical and concurrent statechart that consists of different types of *nodes* and *edges*. A *scenenode* can be linked to one or more scenegroup playback- or system commands and can be annotated with statements and expressions from a simple scripting language or function calls to predefined functions of the underlying implementation language (Figure 2 ①). A *supernode* extends the functionality of scenenodes by creating a hierarchical structure. A supernode may contain scenenodes and supernodes that constitute its subautomata. One of these subnodes has to be declared the *startnode* of that supernode (Figure 2 ②). The supernode hierarchy can be used for type and variable scoping. Type definitions and variable definitions are inherited to all subnodes of a supernode. The supernode hierarchy and the variable scoping mechanism imply a hierarchy of *local contexts* that can be used for context-sensitive reaction to user interactions.

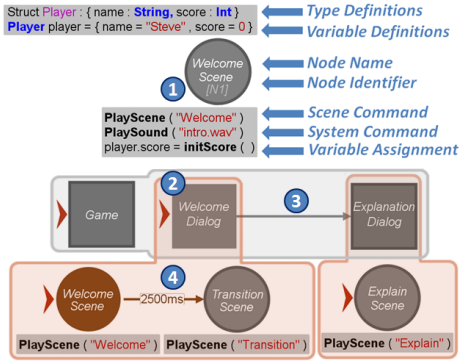

Figure 2: Node statements and supernode hierarchy.

Different *branching strategies* within the sceneflow, e.g. logical and temporal conditions or randomization, as well as different *interaction policies*, can be modeled by connecting nodes with different types of edges. An *epsilon edge* represents an unconditional transition (Figure 2 ③). They are used for the specification of the order in which computation steps are performed and scenes are played back. A *timeout edge* represents a timed or scheduled transition and is labeled with a *timeout value* (Figure 2 ④). Timeout edges are used to regulate the temporal flow of a sceneflow's execution and to schedule the playback of scenes and computation steps. A *probabilistic edge* represents a transition that is taken with a certain probability and is labeled with a *probability value* (Figure 5 ②). Probabilistic edges are used to create some degree of randomness and desired nondeterminism during the execution of a sceneflow. A *conditional edge* represents a conditional transition and is labeled with a *conditional expression*, as shown in Figure 3. Conditional edges are used to create a branching structure in

the sceneflow which describes different reactions to changes of environmental conditions, external events or user interactions. In Section 3.4 we present an application which makes extensive use of a hierarchy nested supernodes to refine the dialogue context for an adequate reaction to the user's interactions.

2.3 Continuous Real-Time Interaction Handling

User interactions as well as other internally or externally triggered events within the application environment can rise at any time during the execution of a model. Some of these events need to be processed as fast as possible to assert certain real-time requirements. There may, for example, be the need to contemporarily interrupt a currently running dialogue during a scene playback in order to give the user the impression of presence or impact. However, there can also exist events that may be processed at some later point in time allowing currently executed scenes or commands to be regularly terminated before reacting to the event. These two different interaction paradigms imply two different *interaction handling policies* that find their syntactical realization in two different types of interruptibility and inheritance of conditional edges:

- *Interruptive conditional edges* (Figure 3 ①,③) are inherited with an interruptive policy and are used for the handling of events and user interactions requiring a fast reaction. Whenever an interruptive conditional edge of a node can be taken, this node and all descendant nodes may not take any other edges or execute any further command. These semantics imply, that interruptive edges that are closer to the root have priority over interruptive edges farther from the root.
- *Non-interruptive conditional edges* (Figure 3 ②,④) are inherited with a non-interruptive policy, which means that a non-interruptive conditional edge of a certain node or supernode can be taken after the execution of the node's program and after all descendant nodes have terminated. This policy is implicitly giving higher priority to any conditional edge of nodes that are farther from the root.

Figure 3 shows a supernode hierarchy with different conditional edges. If the condition "stop" becomes true during the execution of the two innermost scene playback commands, then the scene within the supernodes with the non-interruptive conditions (Figure 3 ②,④) will be executed to its end. However, the scene within the supernodes with the interruptive conditions (Figure 3 ①,③) will be interrupted as fast as possible. In the non-interruptive case the execution of the sceneflow continues with the inner end node (Figure 3 ④) before the outer end node is executed (Figure 3 ②). In the interruptive case the execution of the sceneflow immediately continues with the outer end node (Figure 3 ①) because the outer interruptive edge has priority over the inner interruptive edge (Figure 3 ③).

2.4 Modeling Parallel Dialogue and Behavior

Sceneflows exploit the modeling principles of *modularity* and *compositionality* in the sense of a hierarchical and *parallel decomposition*. Multiple virtual characters and their behavior, as well as multiple control processes for event detection or interaction management, can be modeled as concurrent processes in parallel automata. For this purpose, sceneflows allow two syntactical instruments for the creation of concurrent processes: (1) By defining multiple startnodes for a supernode, as shown in Figure 4, each subautomaton which consists of all nodes reachable by a startnode, is executed by a separate process, (2) by defining *fork edges* (Figure 5 ①) an author can create multiple concurrent processes without the need for changing the level of the node hierarchy.

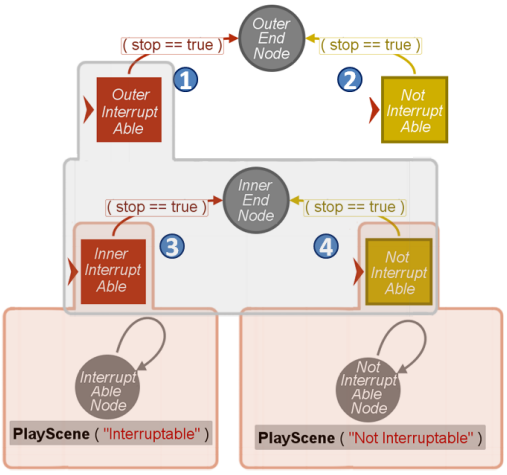

Figure 3: ①,② Interruptive conditional edges. ③,④ Simple non-interruptive conditional edges.

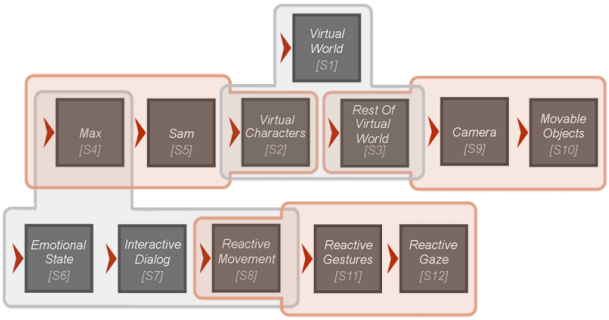

Figure 4: Hierarchical and parallel decomposition.

Following this modular approach, an author is able to separate the task of modeling the overall behavior of a virtual character into multiple tasks of modeling individual behavioral aspects, functions and modalities. Behavioral aspects can be modified in isolation without knowing details of the other aspects. In addition, previously modeled behavioral patterns can easily be reused and adopted. Furthermore, premodeled automata that are controlling the communication with external devices or interfaces can be added as plugin modules that are executed in a parallel process.

Individual behavioral functions and modalities that contribute to the behavior of a virtual character are usually not

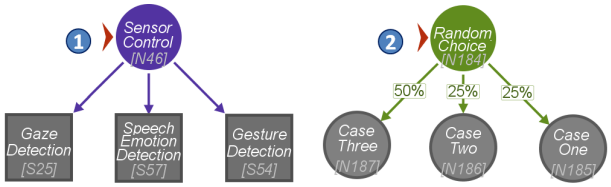

Figure 5: ① Concurrent processes with fork edges. ② Randomization with multiple probability edges.

completely independent, but have to be synchronized with each other. For example, speech is usually highly synchronized with non-verbal behavioral modalities such as gestures and body postures. When modeling individual behavioral functions and modalities in seperate parallel automata, the processes that concurrently execute these automata have to be synchronized by the author in order to coordinate all behavioral aspects. This communication is realized by a *shared memory model* which allows an asynchronous non-blocking synchronization of concurrent processes.

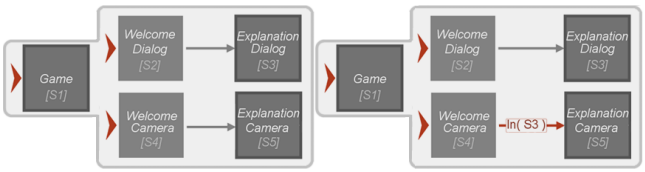

Figure 6: Synchronization over configuration states.

Thereby, sceneflows enfold two different syntactic features for the synchronization of concurrent processes. First, they allow the synchronization over common *shared variables* defined in some supernode. The interleaving semantics of sceneflows prescribe a mutually exclusive access to those variables to avoid inconsistencies. Second, they enfold a *state query condition*, as shown in Figure 6, which represents a more intuitive mechanism for process synchronization. This condition allows to request weather a certain state is currently executed by the sceneflow interpreter during the execution of a sceneflow.

2.5 Consistent Resumption of Dialogue

Our concept of an exhaustive *runtime history* facilitates modeling reopening strategies and recapitulation phases of dialogues by falling back on automatically gathered information on past states of an interaction. During the execution of a sceneflow, the system automatically maintains a *history memory* to record the runtimes of nodes, the values of local variables, executed system commands and scenes that were played back. It additionally records the last executed substates of a supernode at the time of its termination or interruption. The automatical maintainance of this history memory releases the author of the manual collection of such runtime data, thus efficiently reducing the modeling effort while increasing the clarity of the model and providing the author with rich information about previous interactions and states of execution.

The scripting language of sceneflows provides a variety of built-in *history expressions* and conditions to request the information deposited in the history memory or to delete it. The history concept is syntactically represented in form of a special *history node* which is an implicit child node of each supernode. When reexecuting a supernode, the supernode starts at the history node instead of its default startnodes. Thus, the history node serves as a starting point for the author to model reopening strategies or recapitulation phases.

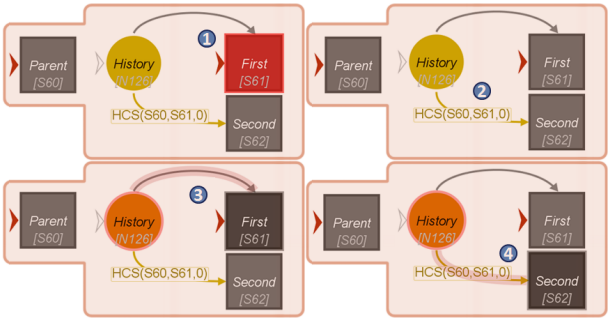

Figure 7: History node and condition.

Figure 7 shows a simple exemplary use of a supernode's history node and a history condition. At the first execution of the supernode "Parent", the supernode starts at its startnode "First" (Figure 7 ①). If the supernode "Parent" is interrupted or terminated at some time and reexecuted afterwards, it starts at the history node "History". The history memory is requested (Figure 7 ②) to find out if the supernode "Parent" had been interrupted or terminated in the node "First" or the node "Second". As the snaphot of the visualized execution shows, depending on the result, the either the node "First" (Figure 7 ③) is executed or the node "Second" (Figure 7 ④) is started over the history node.

3 Applications

The new Scenemaker authoring tool supports the creation of applications with interactive virtual characters on various levels such as the modeling of reactive and deliberate behavior, the use of multiple virtual characters and their synchronization, and the advanced handling of user interactions. In the following, we describe several applications and research projects in which the new Scenemaker tool was used. We present specific aspects of the models created in these applications in order to illustrate the use of certain modeling features of Scenemaker introduced in the previous section.

3.1 IGaze - Modeling Reactive Gaze Behavior

Gaze as an interaction modality has many functions, such like signaling attention, regulating turn-taking or deictic reference (Kipp and Gebhard 2008). An absence of gaze in a virtual character's behavior would be recognized directly by a human interlocutor. Therefore gaze is highly relevant for such characters, especially in human-computer interaction (e.g. COGAIN[1]).

[1] http://www.cogain.org

In the IGaze project, we have modeled a virtual character's gaze behavior as *dominant* or *submissive*. A fine-grained control of gaze can help to improve the overall believability of a virtual character. In one of the first projects that uses the new SceneMaker the gaze behavior for the two characters Sam and Max is represented by a concurrent Sceneflow (see Fig. 8).

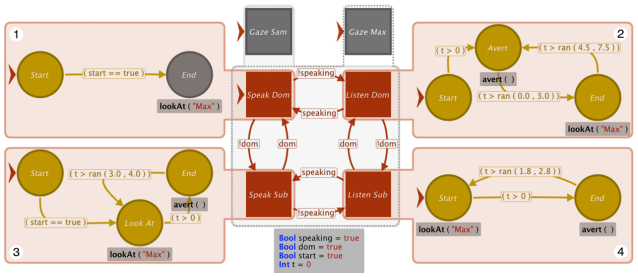

Figure 8: Hierarchical concurrent Supernodes model gaze behavior (*Gaze Sam* and *Gaze Max*)

A separate gaze Supernode for each character holds commands to control its gaze (head) behavior on an abstract level (e.g. **lookat(character)**, **avert**). Those commands define the interface to a characters movement control. In general, the Sceneflow model represents the following gaze behavior:

- Dominant (Dom): High status, according to Johnstone, is gained by outstaring the interlocutor and if a person breaks eye contact and does not look back (Johnstone 1979). The dominant gaze behavior consists of maintaining eye contact while speaking and randomly changing from gazing to averting while listening. More precisely, the character establishes and holds eye contact when speaking (see Fig. 8, ①), and after speaking, immediately looks away. When listening, the character establishes eye contact after 0-3 sec., then holds it for 4.5-7.5 (see Fig. 8, ②).
- Submissive (Sub): Low status, according to Johnstone, means being outstared by the interlocutor or by breaking eye contact and looking back. The submissive gaze behavior makes a character only look briefly every now and then and immediately avert the gaze again. In the submissive gaze mode, a character establishes eye contact when starting to talk but averts his gaze immediately after eye contact. His gaze remains averted for 3-4 sec (see Fig. 8, ③). He then establishes eye contact again and looks away immediately. During listening, the pattern is the same with the difference that the character holds eye contact for 1.8-2.8 sec (see Fig. 8, ④). The submissive avert behavior consists of a head movement away from the user (5° while speaking, 8° while listening) and 15° downward.

A major improvement provided by the Visual SceneMaker is the possibility to verify and alter the timing specifications directly during run-time. This enabled us to carefully adjust the time of a specific gaze aspect in order to achieve an overall compelling result.

On a conceptual level, reactive behavior patterns can be realized as global concurrent Sceneflows or as local concurrent Supernodes that are executed by fork edges at a specific location of a master Sceneflow. Such Supernodes can easily be reused.

3.2 AI Poker - Playing Poker with two Virtual Characters

In the AI Poker, we investigate how modern ECA technologies can help to improve the process of creating computer games with interactive expressive virtual characters. Based on the experience of a computer game company, we identified four main challenges in creating computer games:

- *Fast creation of game demonstrators*. In order to compete with other game companies the implementation of demonstrators has to be fast and reliable.
- *Localization of game content*. To sell games in other countries content has to be translated into the respective language. The more dialogs a game contains, the higher the costs for the translation.
- *Intuitive interaction*. The success of a game is tremendously related to an easy interaction concept.
- *Consistent quality*. The quality of audio and visual presentation should be consistent for the whole game. Every exception lowers its acceptance.

The AI Poker application reuses the gaze control supernodes from the IGaze project. These are extended by two concurrent supernodes, one for automatic camera pan and another for the game interaction control that also controls the two 3d Virtual Characters Sam and Max (see Fig 11), which are in the role of two poker teammates. Sam is a cartoon-like character, whereas Max is a mean, terminator-like robot character. A human user acts as the card dealer and also participates as a regular player.

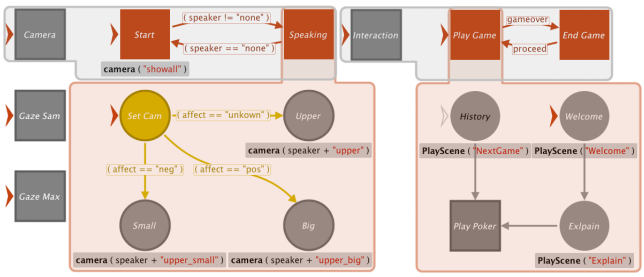

Figure 9: AI Poker's Game Model using hierarchical concurrent supernodes

By using real poker cards with unique RFID tags, a user can play draw poker against Sam and Max. The characters rely on the MARY expressive speech synthesizer using HMM-based and unit-selection-based speech synthesis approaches and the ALMA model for the simulation of affect (Gebhard et al. 2008). It simulates three affect types (emotions, moods, and personality) as they occur in human beings. Based on game events, the affect of each character

is computed in real-time and expressed through speech and body. Both characters are rendered by a 3d visualization engine based on Horde3D (Augsburg University). In order to support Sam's and Max's individual character style, different poker algorithms (realized as separate software modules) are used. Sam relies on a rule based algorithm, whereas Max relies on a brute-force algorithm that estimates a value for each of the 2.58 million possible combinations of five poker cards. The Visual SceneMaker as a central component allows to control of all these techniques and enables the characters to show a consistent emotional expressive behavior that enhances the naturalness of interaction in the game.

Similar to the gaze behavior in the IGaze project, we realized an automatic camera pan that takes into account the affective state (see Fig 11, left side). While a character speaks, the camera shows its upper body. If no character speaks, both characters are shown. Generally, the camera angle is tilted according to the speaker's affective state. If the character is in a positive affective state, the camera shows the upper body with an ascending angle giving the impression that the character appears slightly bigger. In negative affective states, the camera shows the upper body with an descending angle giving the impression that the character appears slightly smaller.

When a user initiates a game, Sam and Max let the user welcome and explain the game setup and as well the general rules (see Fig 10, right side) before the poker game emerges. The use of History nodes at several positions in the *Interaction* supernode has reduced the complexity of the scene graph by reducing the amount of nodes and edges. As the example shows, the scene `"NextGame"` is played, if the *PlayGame* supernode is executed again, skipping the Welcome and Explain nodes and the connected scenes.

Figure 10: The AI Poker demonstrator at the CeBit exhibition.

The use of previously created gaze supernodes and the basic use of history nodes allowed us to create the AIPoker game in 3 months in total. Technically, the content of the poker game consists of 335 scenes organized in 73 groups. The final demonstrator application has been exposed at the CeBit exhibition and was extremely well attended by the visitors of the exhibition, as shown in Figure 10.

3.3 INTAKT - Multiple Interactive Virtual Characters as Shopping Assistants

This project investigates for an future grocery store approach the use of Virtual Characters in two different roles: 1) personal shopping assistant and 2) expert consultant (Kröner et al. 2009). The latter resides in a special display at every shelf and freezer. These characters' purpose is to explain details of food and provide navigation hints for a faster product localisation. Personal shopping assistants resides in a shopping cart display. They guide a user through the grocery store helping her/him to gather all goods of the provided shopping list.

The used dialog and interaction Sceneflow model reuses several Supernodes from the AI Poker Sceneflow model. Necessary was a slight revision of the camera control Supernodes due to the fact that the Virtual Shopping Assistants do not have any affect. However, the camera zooms at the speaking character showing his upper body.

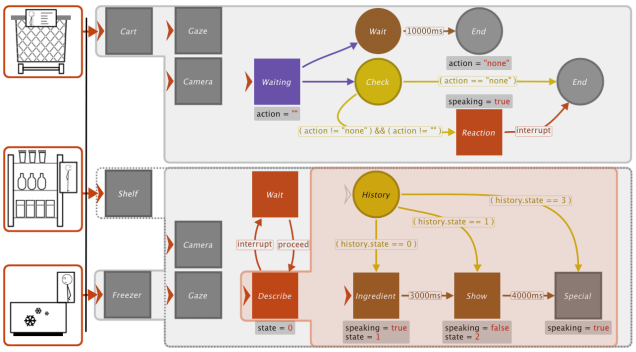

Figure 11: Reuse and extension of hierarchical concurrent Supernodes to model interaction with Virtual Shopping Assistants

Carts and shelves/freezers are equipped with a display showing (different) Virtual Characters that communicate via natural language and natural conversational behaviour with a user and between each other. In addition, the cart display shows a user's shopping list. The characters react every time a product is taken or placed.

The role of the cart character is to guide the user through the shopping list by making suggestions about products to buy (relying on the personal profile, e.g. user prefers ecological products). It serves as a personal advisor that checks every product that is placed in the card against individual needs and individual interests. Therefore, the content of the DPM of each product is used to reason about conflicts with the personality profile. Emerging conflicts are addressed via natural language by the cart character in a low voice - respecting the privacy. Additional information is presented on a display that is attached at the cart. In addition, the cart character may ask the shelf character with a loud (public) voice for help, e.g., if there is a product alternative.

The shelf character provides help by giving in shelf navigation hints for a faster product localisation. In addition, the character provides general information (like price, producer ...) in a natural conversational style.

The user becomes part of this dialog between the characters. Knowledge retrieved from the DPM of the involved products helps to create the illusion that Virtual Characters reacting intelligent to the consumer's interaction with the product.

3.4 SOAP: Modeling Multi-Party Dialogues for an Interactive Storytelling Application

The development of *interactive digital storytelling* systems has been a growing topic of research over the past years. They have been applied for applications in education and training (Marsella, Johnson, and Labore 2003; Si, Marsella, and Pynadath 2005; Swartout et al. 2006) as well as in entertainment and art (Mateas and Stern 2003; Riedl, Saretto, and Young 2003; Cavazza, Charles, and Mead 2001). While some of these systems explore user interaction by putting the user into the role of an observer that can change the world as the story progresses, the majority of them pursues a *dialogue-based* interaction approach. Such systems focus on creating a dramatic experience by offering a selection of dialogue situations in which the user is able to influence the progress and the outcome of the story through interactions.

For the development of interactive storytelling applications it is indispensable to provide authoring software that can be used by non-experts such as artists and screenwriters in order to create highly interactive and consistent multi-party dialogues with the virtual actors. These authoring tools need to have concepts to face challenges such as the continuous real-time processing and context-sensitive interpretation of user interactions, an adequate contemporary reactions to the user's discourse acts and the resumption and revision of dialogue content after unexpected interruptions.

Figure 12: The social game setting in the Virtual Beergarden.

We address these challenges in the social game scenario *SOAP* by using Scenemaker for the dialogue- and interaction management. These ideas have been realized in a demonstrator located in a *Virtual Beergarden* scenario, shown in Figure 12. In the soap-like story, the user and the virtual characters are involved in a romantic conflict. The user, who is represented by an avatar (Figure 12 ①), meets a group of girls (Figure 12 ②) and a group of guys (Figure 12 ③) as well as a waitress (Figure 12 ④). The user can approach the focus groups, listen to their conversations and contribute to the story and thus, influence the progress and outcome of the story.

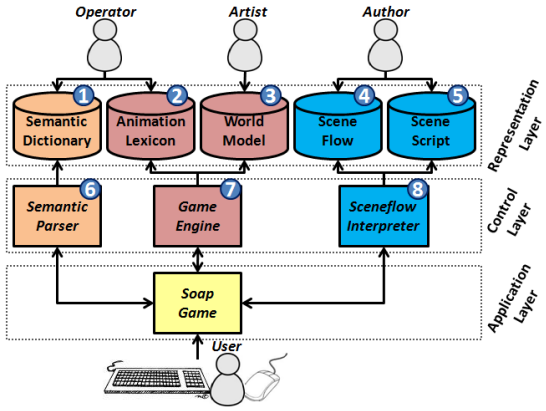

Figure 13: SOAP's component-based system architecture.

The system architecture can be found in Figure 13. The different components are embedded in three independent layers: (1) A representation layer, containing knowledge base and models specifying the scenario content. (2) The control layer, handling the processing of user input and the computation of system output and (3) the application layer enfolding the user interface. Vertically, the components can be categorized into (1) dialogue and interaction management, (2) natural language interpretation and (3) autonomous behavior control, described in the following:

Dialogue and Interaction Management: The behavior modeling as well as the dialog and interaction management of the virtual characters is realized with our modeling tool. An author can specify dialog- and behavior content in a *scenescript* (Figure 13 ⑤) and model the logic of behavior and dialog with a *sceneflow* (Figure 13 ④). An interpreter software executes the model and is, thus, controlling the virtual characters in the game (Figure 13 ⑧).

Figure 14 shows a part of the modeled sceneflow. Each focus group, the user avatar and other game objects are modeled in separate concurrent automata. We also recursively make use of parallel automata in order to model the behavior of individual characters and their behavioral aspects. This procedure reduces the modeling effort and increases the clarity of the model because it prevents the state explosion of the model, which could be observed if we modeled the whole scenario with a simple flat statechart. Furthermore, it allows us to change the behavior of individual focus groups or characters in isolation.

A major requirement in this application was to allow the user to change the focus group, or initiate and terminate a conversation respectivly, at any time. Therefore, each dialogue situation had to be contemporarily interruptible. To create a coherent storytelling experience an interrupted di-

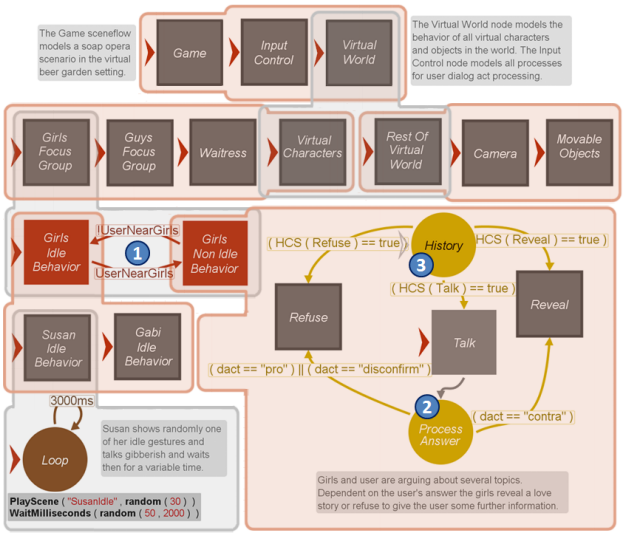

Figure 14: Part of the sceneflow from Soap.

alogue situation had to be consistently resumed after reentering the target group. For these reasons, a highly interactive dialogue structure was modeled and the runtime history was used in order to keep track of previous interactions and the progress of the dialogue. Ongoing dialogues are interrupted whenever the user leaves a focus group and resumed whenever the user reenters the focus group (Figure 14 ①). Consistent resumption or reopening of a previous dialogue is guaranteed by a recursive use of the runtime interaction history (Figure 14 ③). Context-sensitive reaction to the user's interaction is modeled by branching the dialogue structure dependent on the current state of the dialogue and the user's dialogue act provided by the NLU pipeline (Figure 14 ②).

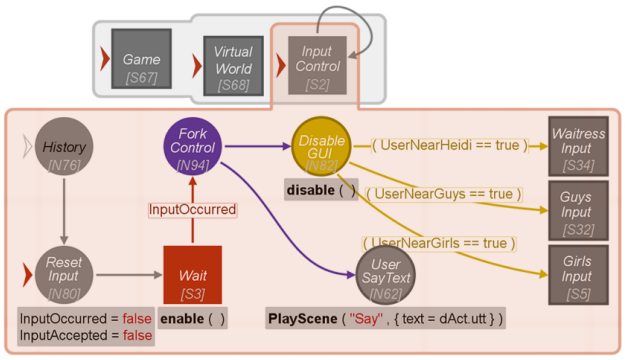

Figure 15: User input processing and control.

To factor out the logic for the detection and the processing of user interactions, we modeled a separate parallel automaton, as shown in Figure 15. This reduces the effort of modeling such logics within the automata for the individual dialog situations to a minimal amount, again effectively increasing the clarity of the model.

Natural Language Interpretation: Our natural language recognition and interpretation pipeline includes a spell checker and the semantic parser *Spin* (Engel 2005) (Figure 13 ⑥) which translates the user's typed-text input into abstract *dialogue-acts* based on the *DAMSL* coding scheme (Core and Allen 1997). The underlying semantic rules are specified in a set of dictionaries (Figure 13 ①) specifying knowledge about the dialogue content. Figure 16 exemplifies a set of rules, syntactic and semantic categories as well as preprocessing steps. The example rule (Figure 16 ③) states that if the user's input contains one of the words "how", "do" or "what" in correlation with the word "you" and any word belonging to the semantic category *location*, the abstract speech act *ask-location* is triggered. The semantic category *location* (Figure 16 ④) contains the words "location", "place", "beergarden", "here" and "party". Thus, different user utterances (Figure 16 ⑤) are parsed into the same dialogue-act. In addition, word stems (Figure 16 ②) and other pre-processing steps can be defined (Figure 16 ①) such as summarizing negations.

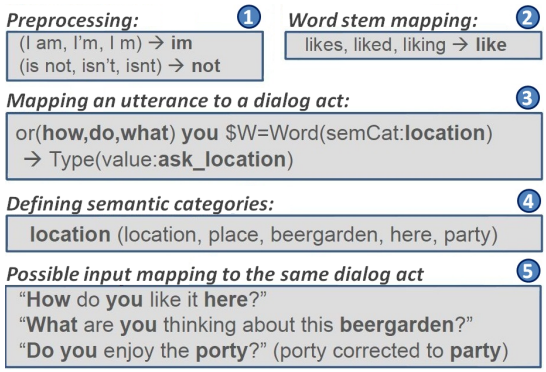

Figure 16: Knowledge defined for the semantic parser.

Autonomous Behavior Control: The scene displayed in the Virtual Beergarden is described by a world model created by an artist (Figure 13 ③). While high-level behaviors of the virtual characters such as speech and gestures are specified using the Scenemaker tool, low-level behaviors such as positioning, agent orientation and proximity or inter-agent gazing are handled automatically by the Virtual Beergarden application. In that manner the author does not need to take care of them. Animations for virtual characters are specified in an *animation lexicon* (Figure 13 ②) including over 40 different gestures and postures for each agent. Following (McNeill 1992), we divide every animation into preparation, stroke and retraction phases, which are used for gesture customization. A Bayesian network can be set for the agents in order to define aspects such as personality or emotional state that influence the manner in which nonverbal behaviors are executed. This has been exemplified for the phenomena of culture-related differences in behavior (Rehm et al. 2007).

3.5 DynaLearn: Modeling Educational Roles for Teaching Assistants

Embodied conversational agents are widely used in educational applications such as virtual learning and training environments (Johnson, Rickel, and Lester 2000). Beside possible negative effects of virtual characters (Rickenberg and Reeves 2000), there is empirical evidence that virtual pedagogical agents and learning companions can lead to an improved perception of the learning task and increase the learners' commitment to the virtual learning experience (Mulken, André, and Müller 1998). They can promote the learners' motivation and self-confidence, help to prevent or overcome negative affective states and minimize undesirable associations with the learning task, such as frustration, boredom or fear of failure. Teams of pedagogical agents can help the learners to classify the conveyed knowledge and allow for a continuous reinforcement of beliefs (André et al. 2000).

Modeling Different Educational Roles In the framework of the *DynaLearn* project, we developed an interactive learning environment in which learners can express their conceptual knowledge through qualitative reasoning models (Bredeweg et al. 2009) and enriched the learning experience with a cast of virtual characters, aiming at increasing learners' motivation and learning success. We considered a variety of teaching methods, learning strategies and ways of knowledge conveyance and verification. These strategies were realized by modeling different virtual hamsters that can play different educational roles (Mehlmann et al. 2010; Bühling et al. 2010). Beside several teachable agents, we modeled a teacher character and a quizmaster character and employed them in various teaching sessions. Figure 17 shows the example of a quizmaster (Figure 17 ①) and two teachable agents (Figure 17 ②) from an educational quiz session as well as an entity diagram representing conceptual system knowledge (Figure 17 ③) from which the reasoning module generates the questions asked by the quizmaster during the quiz session.

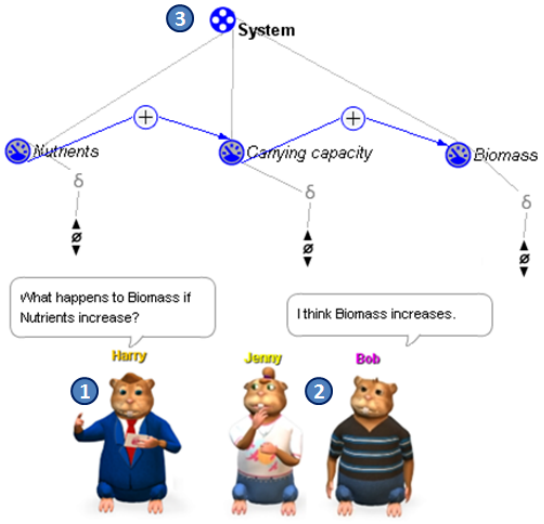

Figure 17: A quizmaster and teachable agents in a quiz

Creating Scenes with Generated Content The logic and the dialogue structure of the different teaching methods were modeled with the Scenemaker authoring tool. Therefore, we integrated the Scenemaker authoring suite with the knowledge reasoning engine by providing a set of functions callable from within the sceneflow model that directly accesses the application interface of the knowledge reasoning module. The possibility to parameterize scenes with arguments from within the sceneflow model allowed the creation of dialogue content consisting both of prescripted content and generated content retrieved from the knowledge reasoning module. Figure 18 shows two examplary scenes (Figure 18 ①, ②) containing generated content that was beforehand retrieved from the knowledge reasoning module over one of the application interface functions (Figure 18 ③).

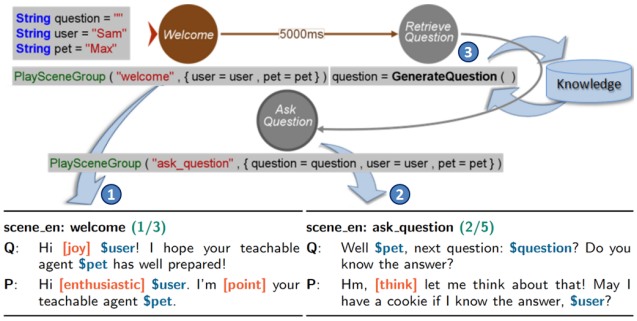

Figure 18: Hybrid scene creation from generated knowledge

4 Field Tests

Success in building different interactive applications with virtual characters for entertainment (Gebhard et al. 2008), education (Mehlmann et al. 2010; Kipp and Gebhard 2008) and commerce (Kröner et al. 2009) permits very promising conclusions with respect to the suitability of our approach. However, mainly computer-experts have been involved in the development of these applications. For this reason, we conducted several field tests and practical workshops with students of different age groups and genders to determine in how far the approach is suited for non-experts. The participating students from various educational levels brought no specific background skills or previous knowledge.

4.1 Nano Camp 2009: School Students Creating Flirting Embodied Conversational Agents

The Scenemaker authoring tool was exposed to a challenging field test in June 2009 at the German Research Center for Artificial Intelligence (DFKI) with secondary and grammar school students (age 12-17). The European broadcasting company *3sat* had invited students from all over Germany to a one-week science camp for hands-on experience with scientific topics. In this context, 12 students were to try out Scenemaker to create an interactive scenario in only 1.5 hrs without prior knowledge or experience. To make this possible, we created a sample scenario where two agents are engaged in a flirt dialogue. Interactivity was given by being able to change the one agent's "flirting strategy" (careful

vs. aggressive). The 12 students were grouped into 6 teams of two people each. After a short introduction (10 mins) including a sample dialogue, the students had some time for brainstorming and sketching the dialogue (40-60 mins). Afterwards, they implemented the sceneflow with minimal assistance (20-40 mins). Every team had to be finished in 1.5 hrs maximum. All 6 teams finished and gave positive feedback about the authoring experience. The resulting scenarios were viewed in the whole group.

4.2 Girls' Day 2010: Teenage Girls Creating a Family Sitcom with Virtual Hamster Characters

A second field test was conducted in the context of Germany's nationwide Girls' Day program (Endrass et al. 2010). This initiative is geared exclusively to female middle school students (age 12-15) and aims at encouraging the students to pursue a career in the natural sciences. Augsburg University invited 9 middle school students to their computer science institute. The students used the Scenemaker tool to create a social game scenario with virtual hamster characters. The 9 students were grouped into 3 teams of three people each and all decided to create some kind of family sitcom episode of about 5 minutes. After a short introduction (20 mins) into the concepts of Scenemaker's modeling approach and the handling of the graphical user interface, the students had some time for brainstorming and sketching the dialogue (40 mins). Afterwards, they modelled the sceneflow with minimal assistance (40 mins), as shown in Figure 19. All 3 teams finished in time and the remarkable resulting scenarios were viewed by the whole group. The students were asked to fill out an evaluation sheet in which they gave positive feedback about the authoring experience.

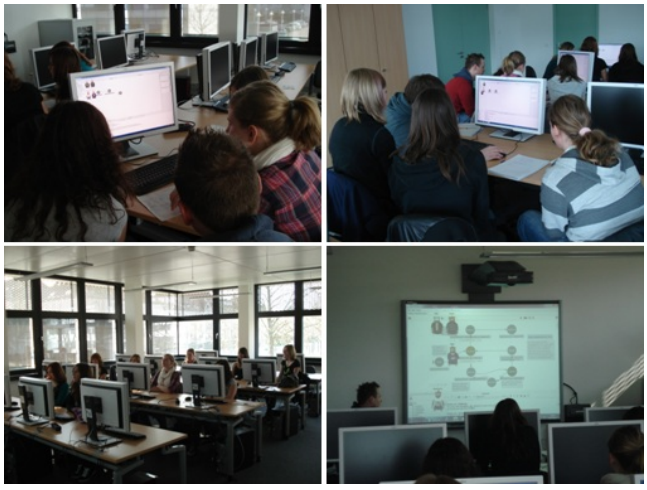

Figure 19: Pictures from the Girls' Day Authoring Session.

4.3 MMP 2011: College Students Creating an Interactive Game with a Virtual Opponent

A third field test was conducted in the context of the multimedia project workshop at the computer science institute of Augsburg University. College students (5th semester) had to model the logic of an interactive battleship game with a virtual opponent (see Figure 20) and the narrative structure of dialogues with the Scenemaker authoring tool. The 9 students were grouped into 2 teams of four people each. The students had been introduced to the modeling concepts of Scenemaker's modeling approach and the handling of the graphical user interface in one lecture session. Most remarkable was, that these students made extensive use of parallel automata for the specification of the virtual opponent's behavior. Each group modeled an automaton simulating the emotional state of the virtual opponent dependend on its success in the game. They synchronized the emotional state with several parallel automata specifying the expressive behavior of the virtual opponent, e.g. for facial expressions and gestures. This showed that the students completely understood and applied Scenemaker's concepts of modularity and compositionality. Already having some previous knowledge in programming, the students claimed that they would have needed much more time and effort to implement the game logic and the agents behavior in a higher programming language, such as Java. They especially praised the intuitive way of creating and synchronizing several concurrent processes, compared with the difficulty they would have had with multi-threading models.

Figure 20: Pictures from the Battleship Game.

4.4 Conclusion

The participants of the field tests were able to quickly pick up most of our concepts for modeling interactive narrative with statecharts and writing scenescripts was promptly and completely understood. This comprehension was directly transfered into the creation of vivid interactive scenarios with virtual characters.

During the field tests, it has been noticeable that the students, with the exception of the college students, in contrast to the computer experts, occasionally had difficulties to apply the more complex concepts of our approach. While the concept of hierarchical and parallel decomposition was

fully understood, the students mostly used concurrent processes to model completely independent parallel behaviors while they rarely utilised the synchronization measures of our language. Furthermore, the history concept was rarely used, because the dialog structure modelled by the students was mostly linear or a tree-like branching structure. They only occasionally had the idea to model dialog situations that could be resumed or reopened after an interruption by using the history concept of our approach.

These observations could be be explained with the short amount of practice time for the students and the schedule of the workshops. We believe that the more complex modeling concepts of our approach will also be fully understood by non-experts after more intensive practice. In the future we plan to do more field test over a longer period of time in order to prove our assumption. This would also allow us to evaluate the quality of the stories and dialogues modeled by the students.

5 Conclusion and Future Work

In this paper, we described a modeling approach to mixed-initiative multi-party dialogues with virtual characters. We presented the integrated authoring tool Scenemaker which allows an author to model complex dialogue behavior and interaction management for multiple virtual characters in a rapid-prototyping style. The statechart language provides different interaction handling policies for an author to handle continuous real-time interaction. The user can interrupt a dialogue at any time and expect a contemporary response. An interaction history allows the author to model reopening strategies for dialogues. After a dialogue was interrupted, it can consistently be resumed and previous dialogue topics can be revised. Our statecharts can be hierarchically refined to create contexts for the interpretation of user input. Parallel decomposition allows to model different behavioral aspects, functions and modalities in isolation. This modular approach reduces the complexity of the model while improving extensibility and reusability. Dialogue content can be authored manually or generated automatically. Blacklisting strategies allow an easy way to provide variability by avoiding repetitive behavior. Autonomous behavior can be specified without the need for an author to explicitly model it.

Success in building different interactive applications with virtual characters for entertainment (Gebhard et al. 2008), education (Mehlmann et al. 2010) and commerce (Kröner et al. 2009) as well as promising feedback from several field tests (Endrass et al. 2010; Kipp and Gebhard 2008) validates the usefulness of our approach. In field tests, students were able to quickly pick up the visual concepts or our UI. In addition, the concept of finite state based modeling of interactive narrative with statecharts as well as the writing scenescripts were promptly and completely understood by the students. This comprehension was directly transfered into the creation of vivid interactive scenarios with virtual characters. Regarding the results of the field tests, we conclude that the Scenemaker software is suitable for rapid prototyping, even for beginners and may be used as an educational device.

Our future work refers on the one hand to technical improvements of the authoring tool based on the user feedback we received so far, such as refinements of our modeling approach, and on the other hand to additional user studies that explore further issues, such as the quality of the scenarios generated with Scenemaker.

For the future we plan to integrate our system with other components, as for example emotion simulation and emotional speech synthesis, nonverbal behavior generation as well as speech recognition. This implies the use of standard languages such as FML, BML and EmotionML (Vilhjalmsson et al. 2007; Kipp et al. 2010; Kopp et al. 2006). Furthermore, we want to integrate a dialog domain knowledge component to the authoring framework, which allows authors to easily define domain knowledge and rules that can map utterances to abstract dialog acts dependent on the dialog domain. This implies the use of an ISO standard for dialogue act annotation (Bunt et al. 2010).

Feedback from users in different field tests and projects has shown that one strength of the modeling approach with Scenemaker is the reusability of already modeled behavioral patterns in the form of sub-models, because this can drastically reduce the modeling effort and complexity. We plan to factor a library of reactive behavior patterns that can be reused for an easy creation of different behavioral aspects for multiple virtual characters. Therefore, one of our main purposes is to identify abstract universal behavioral patterns that appear in multi-party dialogues with multiple virtual characters. We want to provide the author with a library of predefined and parameterizable state chart models, implementing behavioral patterns that can be reused in several projects and can easily be adjusted to the respective context.

6 Acknowledgments

The work described in this paper work was supported by the European Commission within the 7th Framework Programme in the projects IRIS (project no. 231824) and DynaLearn (project no. 231526). In addition, the research was funded in part by the German Federal Ministry of Education and Research under grant number 01 IA 08002 (project SemProM) and under grant number 01 IS 08025B (project INTAKT). We are grateful for the Virtual Character technology and the extensive support provided by the Charamel GmbH.

References

André, E.; Rist, T.; van Mulken, S.; Klesen, M.; and Baldes, S. 2000. The automated design of believable dialogues for animated presentation teams. In *Embodied conversational agents*. Cambridge, MA, USA: MIT Press. 220–255.

Augsburg University. http://mm-werkstatt.informatik. uni-augsburg.de/projects/gameengine.

Bredeweg, B.; Linnebank, F.; Bouwer, A.; and Liem, J. 2009. Garp3 - Workbench for qualitative modelling and simulation. *Ecological Informatics* 4(5-6):263–281.

Brusk, J.; Lager, T.; Hjalmarsson, A.; and Wik, P. 2007. Deal: Dialogue management in scxml for believable game characters. In *ACM Future Play*, 137–144. ACM.

Bühling, R.; Wissner, M.; Häring, M.; Mehlmann, G.; and André, E. 2010. Design Decisions for Virtual Characters in the DynaLearn Interactive Learning Environment. In *Book of Abstracts of the 7th International Conference on Ecological Informatics*, 144–145. Ghent University, Belgium.

Bunt, H.; Alexandersson, J.; Carletta, J.; Choe, J.-W.; Fang, A. C.; Hasida, K.; Lee, K.; Petukhova, V.; Popescu-Belis, A.; Romary, L.; Soria, C.; and Traum, D. R. 2010. Towards an ISO standard for dialogue act annotation. In *7th International Conference on Language Resources and Evaluation (LREC)*.

Cavazza, M.; Charles, F.; and Mead, S.-J. 2001. Agents' Interaction in Virtual Storytelling. In *IVA 2001*, 156–170.

Core, M., and Allen, J. 1997. Coding Dialogs with the DAMSL Annotation Scheme. In *Working Notes of AAAI Fall Symposium on Communicative Action in Humans and Machines*.

Endrass, B.; Wissner, M.; Mehlmann, G.; Buehling, R.; Haering, M.; and André, E. 2010. Teenage Girls as Authors for Digital Storytelling - A Practical Experience Report. In *Workshop on Education in Interactive Digital Storytelling on ICIDS*.

Engel, R. 2005. Robust and efficient semantic parsing of freeword order languages in spoken dialogue systems. In *Interspeech 2005*.

Gandhe, S.; Whitman, N.; Traum, D.; and Artstein, R. 2008. An integrated authoring tool for tactical questioning dialogue systems.

Gebhard, P.; Kipp, M.; Klesen, M.; and Rist, T. 2003. Authoring scenes for adaptive, interactive performances. In *AAMAS 2003*. ACM. 725–732.

Gebhard, P.; Schröder, M.; Charfuelan, M.; Endres, C.; Kipp, M.; Pammi, S.; Rumpler, M.; and Türk, O. 2008. IDEAS4Games: Building Expressive Virtual Characters for Computer Games. In *IVA 2008*, 426–440.

Harel, D. 1987. Statecharts: A visual formalism for complex systems. In *Science of Computer Programming*, volume 8, 231–274. Elsevier.

Heidig, S., and Clarebout, G. 2010. Do pedagogical agents make a difference to student motivation and learning? A review of empirical research. *Educational Research Review*.

Iurgel, I. A.; da Silva, R. E.; Ribeiro, P. R.; Soares, A. B.; and dos Santos, M. F. 2009. CREACTOR - An Authoring Framework for Virtual Actors. In *IVA 2009*, 562–563. Springer.

Johnson, W. L.; Rickel, J. W.; and Lester, J. C. 2000. Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education* 11:47–78.

Johnstone, K. 1979. *Impro. Improvisation and the Theatre*. New York: Routledge/Theatre Arts Books.

Kipp, M., and Gebhard, P. 2008. IGaze: Studying reactive gaze behavior in semi-immersive human-avatar interactions. In *IVA 2008*, 191–199.

Kipp, M.; Neff, M.; Kipp, K. H.; and Albrecht, I. 2007. Toward natural gesture synthesis: Evaluating gesture units in a data-driven approach. In *IVA 2007*, LNAI 4722, 15–28.

Kipp, M.; Heloir, A.; Gebhard, P.; and Schröder, M. 2010. Realizing multimodal behavior: Closing the gap between behavior planning and embodied agent presentation. In *IVA 2010*. Springer.

Kopp, S.; Krenn, B.; Marsella, S.; Marshall, A.-N.; Pelachaud, C.; Pirker, H.; Thórisson, K.-R.; and Vilhjálmsson, H. 2006. Towards a common framework for multimodal generation: The behavior markup language. In *IVA 2006*.

Kröner, A.; Gebhard, P.; Spassova, L.; Kahl, G.; and Schmitz, M. 2009. Informing customers by means of digital product memories. In *1st international Workshop on Digital Object Memories*.

Marsella, S., and Gratch, J. 2006. Ema: A computational model of appraisal dynamics. In *Agent Construction and Emotions*.

Marsella, S.; Johnson, W.-L.; and Labore, C.-M. 2003. Interactive pedagogical drama for health interventions. In *Artificial Intelligence in Education*, 341–348. IOS Press.

Mateas, M., and Stern, A. 2003. Facade: An experiment in building a fully-realized interactive drama. In *Game Developer's Conference: Game Design Track*.

McNeill, D. 1992. *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press.

McTear, M. F. 1998. Modelling spoken dialogues with state transition diagrams: experiences with the cslu toolkit. In *ICSLP 1998*.

Mehlmann, G.; Häring, M.; Bühling, R.; Wissner, M.; and André, E. 2010. Multiple agent roles in an adaptive virtual classroom environment. In Albeck, J.; Badler, N.; Bickmore, T.; Pelachaud, C.; and Safonova, A., eds., *IVA 2010*, 250–256. Springer.

Miksatko, J.; Kipp, K.-H.; and Kipp, M. 2010. The persona zero-effect: Evaluating virtual character benefits ona learning task. In *IVA 2010*. Springer.

Mulken, S. V.; André, E.; and Müller, J. 1998. The persona effect: How substantial is it? In *Proceedings of HCI on People and Computers XIII*, 53–66. London, UK: Springer-Verlag.

Perlin, K., and Goldberg, A. 1996. Improv: A system for scripting interactive actors in virtual worlds. In *Computer Graphics (SIGGRAPH)*, 205–216. ACM.

Prendinger, H.; Saeyor, S.; and Ishizuka, M. 2004. MPML and SCREAM: Scripting the Bodies and Minds of Life-like Characters. In *Life-like Characters – Tools, Affective Functions, and Applications*. Springer. 213–242.

Rehm, M.; Bee, N.; Endrass, B.; Wissner, M.; and André, E. 2007. Too close for comfort? Adapting to the user's cultural background. In *Workshop on Human-Centered Multimedia*.

Rickenberg, R., and Reeves, B. 2000. The effects of animated characters on anxiety, task performance, and evaluations of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '00, 49–56. New York, NY, USA: ACM.

Riedl, M.; Saretto, C.-J.; and Young, R.-M. 2003. Managing interaction between users and agents in a multi-agent storytelling environment. In *AAMAS 2003*, 741–748. ACM.

Schröder, M. 2008. *Emotions in the Human Voice, Culture and Perception*, volume 3. Pural. chapter Approaches to emotional expressivity in synthetic speech, 307–321.

Si, M.; Marsella, S.; and Pynadath, D. 2005. Thespian: An architecture for interactive pedagogical drama. In *AIED 2005*.

Spierling, U.; Weiss, S.-A.; and Mueller, W. 2006. Towards accessible authoring tools for interactive storytelling. In *Technologies for Interactive Digital Storytelling and Entertainment*. Springer.

Swartout, W.; Gratch, J.; Hill, R.; Hovy, E.; Marsella, S.; Rickel, J.; and Traum, D. 2006. Toward virtual humans. *AI Magazine* 27(2):96–108.

Traum, D.; Leuski, A.; Roque, A.; Gandhe, S.; DeVault, D.; Gerten, J.; Robinson, S.; and Martinovski, B. 2008. Natural language dialogue architectures for tactical questioning characters. In *Army Science Conference*.

Vilhjalmsson, H.; Cantelmo, N.; Cassell, J.; Chafai, N.-E.; Kipp, M.; Kopp, S.; Mancini, M.; Marsella, S.; Marshall, A.-N.; Pelachaud, C.; Ruttkay, Z.; Thórisson, K.-R.; van Welbergen, H.; and van der Werf, R.-J. 2007. The behavior markup language: Recent developments and challenges. In *IVA 2007*.

von der Beeck, M. 1994. A comparison of statecharts variants. In *ProCoS 1994*, 128–148. Springer.

Rapid Development of Multimodal Dialogue Applications with Semantic Models

Robert Neßelrath and **Daniel Porta**
German Research Center for Artificial Intelligence
Stuhlsatzenhausweg 3
66123 Saarbrücken
{firstname}.{lastname}@dfki.de

Abstract

This paper presents a model-driven development approach to rapidly create multimodal dialogue applications for new domains. A reusable and consistent base model and generic processes inside a multimodal dialogue framework enable advanced dialogue phenomena and allow for a scenario- and domain-specific customization without the necessity to adapt the core framework. We introduce declarative adaptation and extension points within the discussed models for input interpretation, output presentation, and semantic content in order to easily integrate new modalities, domain-specific interactions, and service back-ends. Three multimodal dialogue applications for different use-cases prove the practicability of the presented approach.

1 Introduction

Speech-based applications gain more and more acceptance. On mobile devices, users can, e.g., search the Internet, dictate short messages, or even maintain shopping lists by uttering speech commands. This circumvents typing on small screen devices. In contrast, multimodal dialogue user interfaces still get less public attention, despite the benefits for a more natural human computer interaction.

Typical usage scenarios for applications with multimodal dialogue user interfaces comprise all kinds of mobile situations where users have to cope with an eyes-busy primary task, e.g., driving a car or walking through a shopping street, or intelligent environments that ease the user's daily life. Common to these scenarios on an abstract level is the support for several input and output modalities allowing for advanced dialogue phenomena, e.g., the use of deictic, elliptic, spatial, or temporal references.

On a closer look, however, every concrete application domain in one of the usage scenarios has its own mixture of interaction patterns. For example, in the ambient assisted living domain, as one incarnation of an intelligent environment, the focus lies on command and control of home appliances whereas multimodal dialogue infotainment applications implement searching or even question answering interaction patterns. While even different interaction patterns can be implemented to some extent in a generic reusable way, such that they can be applied in different domains, the actual application data and its formalism is totally domain-specific, ranging from, e.g., simple XML schema definitions to complex semantic models in the Web Ontology Language (OWL).

With their high complexity, multimodal dialogue user interfaces inherently require more development effort than traditional graphical user interfaces. We are investigating a generic framework for multimodal dialogue user interfaces which shall ease the development of such interfaces for various application domains in various usage scenarios, and hence reduce complexity by encapsulating recurring tasks and functionalities.

We already adopted our framework for implementing multimodal dialogue user interfaces. (Sonntag and Möller 2010) describe a medical system that supports a radiologist in finding a diagnose, asking for a second opinion, and deciding for an appropriate medical treatment. The system allows semantic image annotations and presents former patients with similar findings. The collaborative kiosk infotainment system described by (Bergweiler, Deru, and Porta 2010) can be deployed in museums or exhibitions. Visitors can ask for relevant information by interacting with a large tabletop surface and their mobile devices which can also be used for sharing media assets. The necessary data is retrieved by accessing a large heterogeneous service back-end which also comprises semantic Web services. (Porta, Sonntag, and Neßelrath 2009) describe a mobile business application that enables decision-makers on the go to still participate in important business processes. The mobile user can handle purchase order requisitions in an enterprise resource planning system and can search for alternative products. Found alternatives can be sorted according to different criteria and visualized in a 3-dimensional space.

The selected implemented systems show that different application domains have already been covered with our generic framework. But adaptations and extensions for new domains have been and will always be necessary. So far, such customizations needed a vast amount of time. In order to reduce this time, a reusable and consistent base modelling and generic processes are inevitable. At the same time, transparent scenario- and domain-specific customization points are required without affecting the base system. We tackle these requirements by applying a model-driven development approach. Multimodal dialogue user interfaces involve a large number of models that incorporate or depend

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 37–47,
Barcelona, July 2011

on each other. Model-driven development is a software development methodology for automatically deriving running applications from formal representations of a domain and system. A formal description of system parts by models pays off in better readable documentation, easier reusability and adaptation, and hence in the reduction of development time.

This paper is outlined as follows. We begin with an overview of processing in multimodal dialogue systems (chapter 2). In chapter 3, we describe important models we use in multimodal dialogue applications and explain how we benefit from them in terms of a rapid development process. Chapter 4 highlights three applications that were recently developed with our generic framework. Finally, we conclude in chapter 5.

2 Processing in Multimodal Dialogue Systems

Our generic framework for building multimodal dialogue user interfaces is called the Ontology-based Dialogue Platform (ODP) (Schehl et al. 2008) and includes interfaces to relevant 3rd-party ASR/NLU (e.g., Nuance) and text-to-speech (TTS, e.g., SVOX) components. It also provides a runtime environment for multimodal dialogue applications supporting advanced dialogical interaction. The central component is a dialogue system which uses a production rule system (Pfleger 2004) for a context-aware processing of incoming requests (e.g., display and discourse context) and events. It is based on domain-specific models, e.g., the UI and discourse model. The models include the reaction to pointing gestures, the natural language understanding process, the representation of displayed graphics, and the speech output. Furthermore, the dialogue system provides a programming model for connecting multiple clients (session management) for presentation and interaction purposes. The external and application-specific components in the backend layer can also be accessed easily.

Additionally, the ODP supports the development process with a set of Eclipse-based integrated tools for editing, debugging and testing of semantic objects, rules and grammars (Sonntag et al. 2009). Experience from several research projects like SmartKom (Wahlster 2006) and SmartWeb (Sonntag et al. 2007) influenced the design of the framework. It is based on the abstract reference architecture for multimodal dialogue systems as introduced by (Bunt et al. 2005). Typically, an ODP application consists of one or more (thin) clients, the server-side ODP runtime environment, and the domain-specific application back-end. Hence, the ODP runtime environment acts as a middleware between the actual user interface and the back-end in order to hide the complexity from the user by presenting aggregated data. The internal workflow is divided into three processing phases: (i) understanding, (ii) dialogue management and action planning, and (iii) output generation. Figure 1 shows how these phases are realized within the ODP in groups of standard components for monomodal input interpretation, multimodal fusion and discourse resolution, dialogue management and action planning, service access, modality fission, and modality specific output generation.

The components for monomodal input interpretation are specialized recognizers for single modalities like a speech recognizer or a gesture classifier. References are resolved in the multimodal fusion and discourse resolution engine. The result of the understanding phase is a list of weighted hypotheses of user intentions. The dialogue manager selects the best fitting intention taking into account the computed weights and plans and executes the appropriate actions. Often, access to back-end services is necessary for task-driven interaction such as command and control or searching. Presentation planning computes a high level presentation of the result that is distributed to the output modalities by the fission component.

Figure 1 also emphasizes the adaptation work a developer has to perform. Green marked components execute generic processes that operate on domain-independent concepts. Actually, they do not require any adaptation for new applications. In detail, these are the fusion and discourse resolution engine and the multimodal fission component. Yellow marked components (input interpretation, interaction management, presentation planning) have to be adapted to a new domain. But this can be done in a purely declarative way. Although they already contain generic processes that operate on abstract concepts, concrete domain-specific concepts have to be derived. Optionally, the generic processes of these components might need to be refined. The red marked service access and semantic mapping component is rather domain-specific. Additional to the work required for the yellow components, implementation work for accessing domain-specific services is necessary.

3 Models in Multimodal Dialogue Systems

Throughout the framework, we use a semantic modelling approach to achieve a consistent description of content. (Araki and Funakura 2010) examine some possibilities to use ontologies in spoken dialogue systems. Benefits are found for language models, semantic analysis of utterances, frame-driven dialogue management and user modelling. In (Milward and Beveridge 2003), ontologies support dialogue management, generation of language models for speech recognition and generation and input interpretation. We adopt this idea and introduce semantic models that support the development process. Semantic models and appropriate abstraction layers enable reasoning algorithms to support generic processes throughout the system. In this paper, we focus on:

- The **Content Model** is the semantic representation of the domain-specific data. New content from services is integrated into the system after it is lifted onto a semantic level (section 3.1). In the figures of this paper, concepts, properties, and instances of this model are painted in yellow. Basic concepts are marked with the namespace prefix `base:`. Derived domain-specific concepts wear a different prefix indicating their originating source, e.g., `fb:` for Facebook or `dbpedia:` for DBPedia.
- The **Reference Model** provides means to describe references to content in a generic way throughout the whole framework (section 3.2). Concepts, properties, and in-

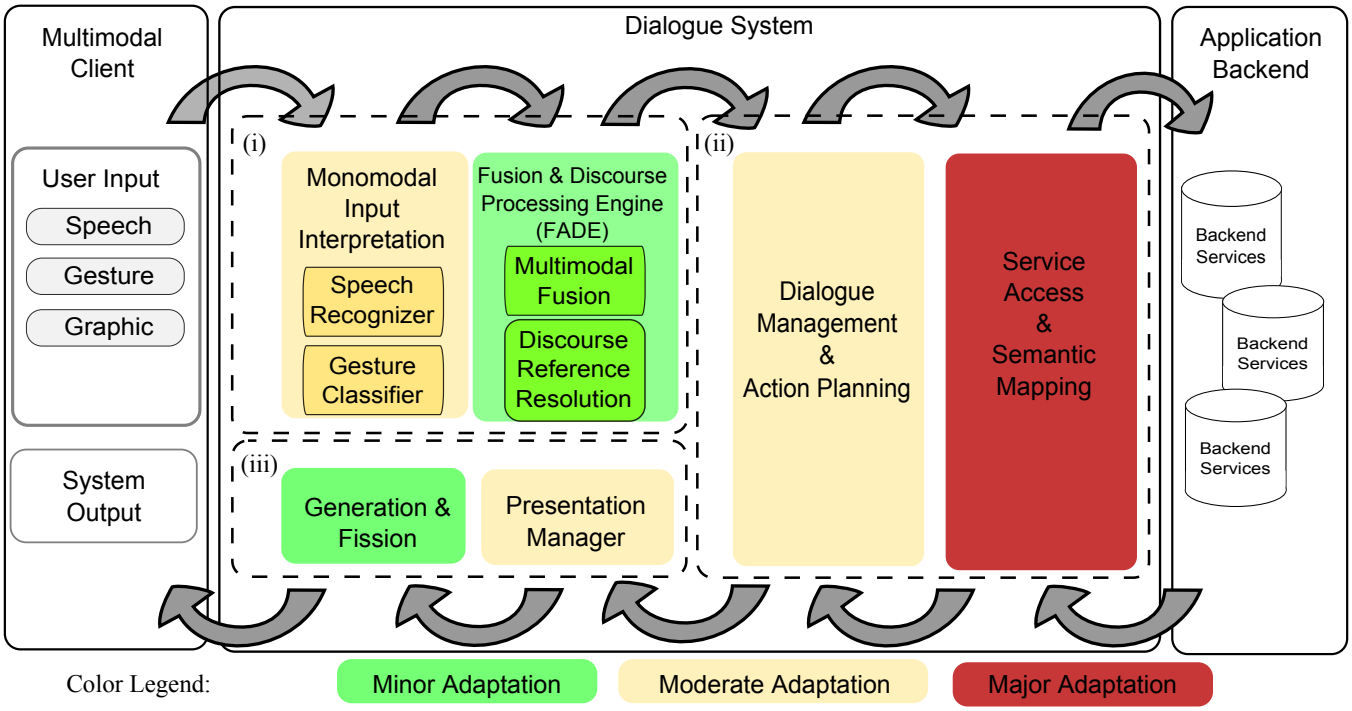

Figure 1: The dialogue processing cycle and development costs for building a new application

stances of this model are marked with the namespace prefix `ref:` and painted in red.

- The domain-independent **Input Interpretation Model** encapsulates the results of the monomodal input interpretation components and is processed in the fusion and discourse resolution engine and later in the dialogue management (section 3.3). Concepts, properties, and instances of this model are painted in green. Basic concepts are marked with the namespace prefix `base:`. Derived domain-specific concepts wear a different prefix indicating the application domain, e.g., `uch:` for the smart kitchen control application (section 4.3) or `isis:` for the ISIS information system (section 4.3).
- The **Graphical Presentation Model** describes the presentation, content and behaviour of graphical user interfaces (section 3.3). Concepts, properties, and instances of this model are marked with the namespace prefix `gui:` and painted in blue.

3.1 Semantic Content Model

With the years, the idea of Web content shifted from a technical domain only accessible by experts to an open community where anyone can contribute. Nowadays, more and more applications depend on Internet access in order to retrieve and store their data in the Web. The "Internet of Services" and "Cloud Computing" introduce lots of smart Web applications and service mashups exposing terabytes of heterogeneous and unstructured content. Hence, an important task is to improve consistency and availability of Web content in order to integrate it into applications (Allemang and Hendler 2008).

We attended to this task during the development of semantic-based multimodal dialogue applications that acquire their content from different heterogeneous Web services. Typically, it is worthwhile to hide the prevalent form of heterogeneity from end users in order to avoid complexity and raise their acceptance for the application. For example, it is incomprehensible for a user that presentation and interaction with an entity of type *Person* that is retrieved from Wikipedia is fundamentally different to the interaction with an entity of same type from a personal addressbook. Here, a consistent pattern for presentation and interaction is preferred.

Knowledge engineers envision the Semantic Web (Berners-Lee, Hendler, and Lassila 2001) which not only describes content but also the meaning and interrelationship of that content. It allows to combine common facts from different sources but also leaves room for perhaps contrary opinions. This follows the *AAA Slogan* "**A***nyone can say* **A***nything about* **A***ny topic.*" mentioned in (Allemang and Hendler 2008). Languages for knowledge and ontology representation like OWL support the merging of semantic content from different sources by offering mechanisms to define equivalent classes (`owl:equivalentClass`) and instances (`owl:sameAs`).

Unfortunately, a lot of relevant Web content is still not available in a semantic representation or full-fledged semantic processing at interactive speeds is hard to achieve. Work goes towards a more lightweight Semantic Annotated Web, e.g., RDFa is an extension for XHTML to embed RDF in Web documents (Adida et al. 2008). But until we reach a

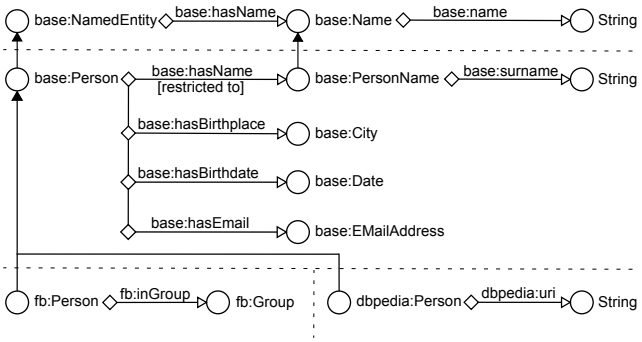

Figure 2: The domain-independent concept `base:Person` and the inherited concepts for DBpedia and Facebook contacts.

```
<object type="ref#ReferenceModel">
  <slot name="ref#hasPattern">
    <object type="uch#Appliance">
      <slot name="uch#hasState">
        <object type="uch#PowerModeState" />
      </slot>
    </object>
  </slot>
  <slot name="ref#hasType">
    <object type="base#DeicticReference" />
  </slot>
</object>
```

Figure 3: Example of a reference model.

level at which most content is represented in a semantic form, more persuasion work and tooling support is needed to convince a larger community of this idea. Adopted from the Web 2.0, service mashups already found their way into the Semantic Web (Ankolekar et al. 2007). Usually, these (composed) services can be queried via standard interfaces like WSDL/SOAP or REST and return XML structures. If not offered by the service itself, these results must be lifted from a pure syntactic onto a semantic level in order to integrate them into a semantic-based application. Often, such service mashups provide valuable data for context-sensitive dialogue applications (Sonntag, Porta, and Setz 2010). Despite the advantages, e.g., making implicit knowledge accessible by inferencing, new problems arise in terms of semantic mediation of content from heterogeneous sources. Some semantic Web services like DBPedia (Bizer et al. 2009) are queried by SPARQL expressions and return their content in RDF bindings that conform to the underlying ontology, whereas not every semantic Web service is operating on the same (upper) ontology.

We have to cope with that in the ODP. Due to the lack of a sufficient semantic infrastructure in Web services, we still need a mapping process that integrates service answers into the running application in order to present them to the user and make them available for interaction. For this, we need well chosen concepts for knowledge representation within our base dialogue ontology which cover most aspects of the content we process (e.g., NamedEntity, Person, Location). Additional domain-specific data must be specified in inherited concepts. Content structures from heterogeneous Web services are mapped onto this extended ontology by applying a rule-based mediation process. Since content representation strongly depends on the domain and the set of accessed services, required mapping rules have to be written manually or in the best case semi-automatically.

As an example, we imagine an interaction system that handles person data from different back-end services. One is DBPedia, that makes Wikipedia content accessible in a semantic form. Second are contacts extracted from facebook. Most properties like the name or date of birth are common properties that are delivered from both information sources. Some others are very domain-specific. Figure 2 depicts the inheritance tree of the concept `base:Person` which bundles the intersection of the most common properties of a person. Since these properties are modelled in the base ontology, rules for interaction, dialogue management and presentation that handle abstract person instances can also handle instances from both sources in the same way. The person instances retrieved from DBPedia (`dbpedia:Person`) and facebook (`fb:Person`) contain additional mutual exclusive properties (Facebook groups or DBPedia resource URIs) which become relevant for very domain-specific interaction or back-end retrieval tasks.

3.2 Reference Model

Referring expressions are a key linguistic phenomenon of verbal interaction. One of the first systems that challenges multimodal fusion is the "Put-That-There" system (Bolt 1980), where the speech command *"Put that there."* gives information about the act itself but contains two placeholders, the first for an item on a screen, the second for a position where the item should be placed. These placeholders are filled with information given by pointing gestures.

In other cases, user intentions refer to a previous interaction or situational context. A dialogue system has to intelligently integrate this context information for the interpretation of the user's intention. A user that gives a speech command *"Turn on the lamp"* to turn on the lamp in the room in which he currently resides, gives an incomplete command insofar that the action but not its target is defined assuming that there are more rooms in the apartment. Nevertheless, the system retrieves information about the type of the target, in this case a lamp. Humans directly understand the actual intention of the command and switch on the lamp of the correct room. A system that takes a context model for the user's location and the type of controllable devices nearby into account can apply reasoning algorithms to infer the same conclusion.

In the ODP, we use a reference model for describing partially defined information. A semantic pattern restricts the type of the missing instances and their properties. Additionally, it comprises linguistic information like case, gender, part-of-speech, or number that is valuable information for the resolution of the reference. We build on the approach from (Pfleger 2007) that describes the rule-based fusion and

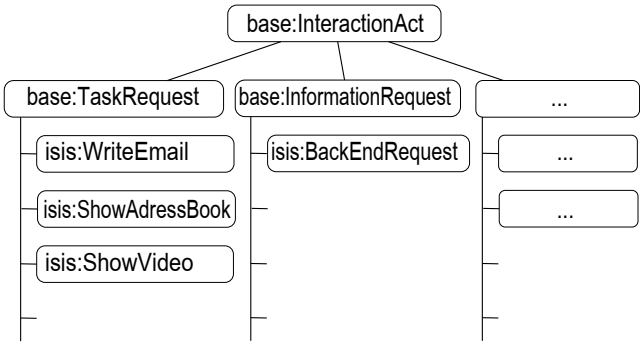

Figure 4: Upper level ontology of the interaction act model.

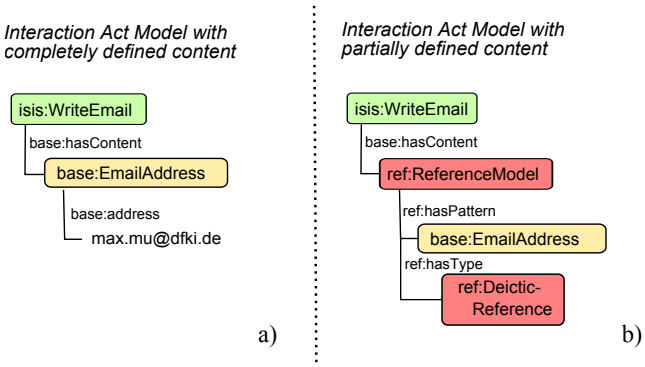

Figure 5: Instances of interaction act models with a) completely and b) partially defined semantic content.

discourse resolution engine FADE. Here, besides the pattern for semantic content, reference types define expectations about where to find the missing instances in the discourse context. These types are: spatial references, deictic references/multimodal fusion, elliptic expressions, temporal relations, and references to items in presented collections. Generic rules in the FADE component are responsible for reference resolution by applying unification and overlay algorithms (Alexandersson, Becker, and Pfleger 2006) on the references and the semantically represented discourse context.

Imagine a pointing gesture to a kitchen appliance combined with the command *"Turn this on!"* Here, the user's speech command is ambiguous. The missing and sufficient information is given with the pointing gesture. Figure 3 shows an instantiated reference model that comprises two details about the referred instance. First is a semantic pattern that only allows instances of type `uch:Appliance` with a power-mode functionality. Second is the type of the reference, in the example a deictically introduced instance.

For our model-driven development approach, we identified an additional purpose for referring expressions and their resolution strategies. By introducing the `ref:DataModelReference`, we enable a loose coupling in model definitions for content, interaction acts, and graphical presentation. Here, the role of a reference model is to describe the connections between these models by patterns (figure 7 provides an example). In the next two subsections, we will show how the reference model helps to accelerate the development process of semantic-based multimodal dialogue applications.

3.3 Input Interpretation Model

Depending on the situation, users prefer different input modalities to interact with an application. For every modality, a monomodal input interpreter hypothesizes user intentions. Their results are integrated into the ODP by means of the interaction act model.

Interaction Act Model (Bunt 2000) distinguishes between two content types in the meaning of an utterance. First is the information that is introduced into the dialogue. Second is the communicative function that describes the way, how the new information is introduced. Their combination is a dialogue act.

The base ontology comprises concepts for the representation of these content types. According to (Bunt et al. 2010) and (Alexandersson et al. 1998), these are communicative functions like grounding, informing, questioning, answering, task offering, turn taking, etc. They are enriched with semantic content that delivers more information about the user's intention and instances that are introduced with the act. Figure 4 shows an extract of the upper level ontology of this model. All interaction act concepts are derived from the base concept `base:InteractionAct`. Inherited concepts like `base:TaskRequest` or `base:InformationRequest` describe the communicative function of the act. The idea of domain-independent dialogue acts was already presented and applied in several related projects like TALK (Becker et al. 2006) and has proven of value for our development approach.

The interaction act example in figure 5 a) represents a task request to write an e-mail to a person like "I'd like to write an email to Max Mustermann". The communicative function is given by the domain-specific concept `isis:WriteEmail` that is derived from the abstract concept `base:TaskRequest`. The semantic content is delivered in the inherited property `base:hasContent`. It can be underspecified ("Write an email to this contact"). In this case, models for referring expressions give additional information about the missing content, e.g., linguistic or semantic information. In figure 5 b) the content is restricted to instances that unify with the concept `base:EmailAddress`. The content of the property `ref:hasType` describes the type of the reference, in this case a deictic one.

The agreement on an interaction act model enables the easy extension of applications with new input modalities. Input interpreters that define their results with interaction act models can be rapidly integrated into a multimodal dialogue application without adapting the internal framework components.

Fusion and Discourse Processing The fusion and discourse resolution engine resolves interaction acts with partially given content. In most cases, missing content must

be filled with information from other modalities or the discourse context. In other cases, a new data instance is introduced and the missing user intention must be resolved. We observe the following sequence of utterances:

(1) "Turn on the lamp."
(2) "And the TV."
(3) "Increase the volume."

In this example utterance (2) introduces a new appliance, the TV, into the dialogue. It does not provide the user's intention what to do with the TV, i.e., turn it on. This must be reasoned from the previous turn (1) of the dialogue. Utterance (3) gives no information about the appliance of interest. Besides the TV, a radio could be running that also contains a volume control. This ambiguity can only be clarified in the discourse context.

Figure 6 shows three possible combinations for the completeness of the interpretation of a user's intention. Example a) is the complete interpretation for utterance (1). No further resolution work is necessary. Example b) shows an interaction model of an utterance with a deixis that refers to a pointing gesture or to an elliptic expression, like in utterance (4):

(4) "Turn this appliance on."

Here, the deixis is represented with a deictic reference model that restricts the referred instance to an `uch:Appliance` with a power-mode functionality. Example c) introduces a lamp entity to the discourse. This occurs when the user mentions the lamp by speech or points with his finger on a graphical object that represents it.

The fusion and discourse resolution engine follows several resolution strategies to fill the missing content for an incomplete interaction act. When receiving input from two different modalities it tests whether the semantic information of one modality unifies with the reference model pattern of the other one. So, in figure 6 the interaction act b) and c) would unify to the complete interaction act a). Notice that the reference pattern in example b) wouldn't unify with an object that is no appliance with power mode functionality, like a chair. Then the fusion and discourse resolution engine would try to find models from preceding turns for interaction act completion. When a television was introduced in a previous turn, the reference pattern would unify with it, based on the fact that this is an appliance with a power mode functionality.

3.4 Graphical Representation and Presentation

In most scenarios, multimodal dialogue applications support a graphical user interface. Hence, a model for graphical output representation is a central component in application development. We introduce a presentation model that describes the actual graphical output of an application.

Graphical Representation Model According to the principle *"No presentation without representation."* (Maybury and Wahlster 1998), we include a set of

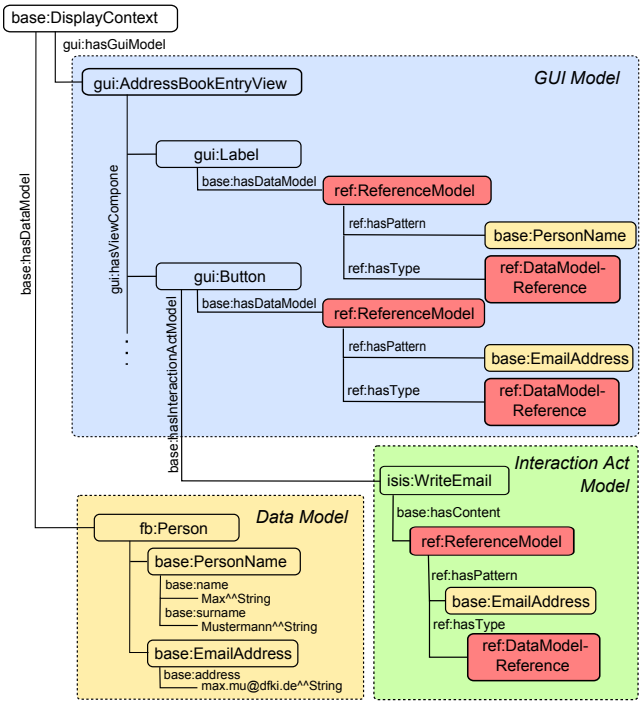

Figure 7: The display context as a combination of gui, data and interaction act models

abstract concepts for view components in the base dialogue ontology. Every view component is derived from the concept `gui:ViewComponent`. Derived concepts `gui:InactiveViewComponent` and `gui:ActiveViewComponent` indicate the interaction capability of inherited view components. Active components are components the user can interact with, like a button or lever. Inactive components only present information, e.g., a label or an image. For creating hierarchical structures, the concept `gui:ViewComponent` owns a property `gui:hasViewComponent` that allows graphical components to contain others.

Similar to the idea of RDFa, we enrich the view components with the semantic content they represent. Active view components can additionally embed an interaction act that defines the interpretation of the user's intention when interacting with the view component. We call the model for the graphical user interface together with the additional information about semantic content and interaction acts the *display context*.

Figure 7 shows an example of a concrete display context. The GUI displays an address book entry (represented by `gui:AddressBookEntryView`) that contains a label for the person's name. An interactive button displays the e-mail address. Pushing the button triggers the underlying interaction act (write an e-mail with the defined e-mail address). Models for user intentions are directly integrated into the definition of the GUI. So an interaction of the user with a view component can later directly be mapped onto an interaction act hypothesis.

```
<object type="base#TaskRequest">
  <slot name="base#hasContent">
    <object type="uch#ManipulateTargetTask">
      <slot name="uch#hasTarget">
        <object type="uch#Lamp">
          <slot name="uch#identifier">
            <value type="String">lamp01</value>
          </slot>
          <slot name="uch#hasState">
            <object type="uch#PowerModeState" />
          </slot>
        </object>
      </slot>
    </object>
  </slot>
  <slot name="base#hasContent">
    <object type="uch#BooleanStateCommand">
      <slot name="base#identifier">
        <value type="String">powerMode</value>
      </slot>
      <slot name="uch#commandType">
        <value type="String">set</value>
      </slot>
      <slot name="uch#hasValue">
        <object type="base#BooleanValue">
          <slot name="base#hasBooleanValue">
            <value type="Boolean">true</value>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
</object>
```

(a)

```
<object type="base#Inform">
  <slot name="base#hasContent">
    <object type="uch#Lamp">
      <slot name="uch#identifier">
        <value type="String">lamp01</value>
      </slot>
      <slot name="uch#hasState">
        <object type="uch#PowerModeState" />
      </slot>
    </object>
  </slot>
</object>
```

(c)

```
<object type="base#TaskRequest">
  <slot name="base#hasContent">
    <object type="uch#ManipulateTargetTask">
      <slot name="uch#hasTarget">
        <object type="ref#ReferenceModel">
          <slot name="ref#hasPattern">
            <object type="uch#Appliance">
              <slot name="uch#hasState">
                <object type="uch#PowerModeState" />
              </slot>
            </object>
          </slot>
          <slot name="ref#hasType">
            <object type="DeicticReference" />
          </slot>
        </object>
      </slot>
      <slot name="base#hasContent">
        <object type="uch#BooleanStateCommand">
          <slot name="base#identifier">
            <value type="String">powerMode</value>
          </slot>
          <slot name="uch#commandType">
            <value type="String">set</value>
          </slot>
          <slot name="uch#hasValue">
            <object type="base#BooleanValue">
              <slot name="base#hasBooleanValue">
                <value type="Boolean">true</value>
              </slot>
            </object>
          </slot>
        </object>
      </slot>
    </object>
  </slot>
</object>
```

(b)

Figure 6: Instances of interaction act models with diverse complexities.

Display Context Generation In the first generation of our framework, display context representations were created manually and filled with semantic content by handwritten rules. This implied a lot of manual work for adapting, extending or creating new views. In a next step, we introduced templates for views that can be connected to arbitrary data models. Figure 7 shows an example for such a template. A view for an address book entry contains several view components for the visualization of contact information. A label shows the person's name and a button displays the e-mail address. The actual content that is presented to the user is connected to the view components by means of the reference model. It refers to the data model attached to the display context. For this, we extended the reference model with a concept `ref:DataModelReference`. The `ref:hasReferenceObject` restricts the possible content items for the view component to concepts that unify with a semantic pattern, here a `base:PersonName` for the label and an `base:EMailAddress` for the button. The references are resolved at run-time. The process applies a breadth first search on the data model structure to find an instance that matches the given restrictions in the reference model. Finally, the referred content for the label and the button are filled with the actual name and e-mail address of Max Mustermann from the attached data model.

Also, the description of the interaction act can be defined by a template. The e-mail button of the addressbook view contains an interaction act that describes the system's behaviour when the user pushes the button. Here, the data model is also defined by a reference. The resolution component retrieves and fills the missing content of the interaction act by searching an instance in the `base:hasDataModel` slot that unifies with the given pattern.

The loose coupling of the data model and the interaction act allows developers to rapidly change the content of the presented data and the behaviour of the system even during runtime by just exchanging the connected models. Even data of different types can be attached as long as it contains instances that match the patterns specified by the reference models.

4 Demonstrators

We already applied our described model-driven development approach and the ODP as semantic-based multimodal dialogue application framework for several demonstration systems and applications. In the following sections, we will briefly describe three of them, covering a wide range of usage scenarios and domains. Although the presented demonstrators do not retrieve missing or resolve ambiguous information by asking the user, the former Babble-Tunes system (Schehl et al. 2008), also developed with ODP, implements this functionality. A future work is to generalize the knowledge we gained from this demonstration system in order to derive concepts for the easy integration of clarification dialogues into new systems.

4.1 Smart Kitchen Control

The smart kitchen at the German Research Center for Artificial Intelligence in Saarbrücken is a completely equipped kitchen where all appliances are accessible via a network connection. The type of connection and protocols are a various set of different technologies. Hood, oven, hob, fridge and freezer are connected via powerline technology and can be controlled via a WSDL/SOAP interface. The light in the kitchen and additional sensor technology like movement, light, temperature or contact sensors for windows and doors are accessed via the battery and wireless radio technology EnOcean. A television, in form of the Windows Media Center, runs on a desktop PC. All appliances of the smart kitchen are integrated in the middleware technology UCH that implements the ISO/IEC 24752 Universal Remote Console standard (Zimmermann and Vanderheiden 2007) and provides the back-end service for a multimodal dialogue appliance control.

For this use-case, the dialogue ontology is extended by concepts for appliance and functionality description. That makes it possible to distinguish between discrete and continuous functionalities. E.g., a power-mode functionality is represented as a boolean value, so it only allows the commands on, off and toggle. A continuous value can be increased, decreased, and set to a certain value. This consistent modelling approach allows the dialogue system to connect user intentions to the correct appliance functionalities by unification.

Figure 8 shows the remote kitchen control, a mobile client running on the android platform. It enables multimodal dialogical interaction with the kitchen by (i) providing a graphical user interface that supports pointing gestures, (ii) streaming capabilities for sending and receiving audio data, and (iii) integrating a toolkit for gesture classification (Neßelrath and Alexandersson 2009). The gesture classifier exploits the accelerator sensor data of the device and enables control with hand gestures. Independent of the modality, all interpreters deliver interaction act models as described in section 3.3. This allows the system to resolve several dialogue phenomena discussed in the previous chapter:

- **Multimodal Fusion:** Speech commands are combined with pointing gestures. A user can point to an appliance symbol on the screen and give the speech command: "Turn this on."
- **Elliptic Reference:** The context for an incomplete command can be reasoned from previous turns. This allows the system to understand the following sequence of commands:
  "Turn on the Hood."
  "Increase the light setting."
  "Brighter."
- **Context resolution for hand movement gestures:** Movement gestures are interpreted as an appliance functionality manipulation without defining an appliance. Imagine that describing a circle is the meaning to turn on an appliance. The appliance of interest is reasoned from a previous turn or the actual display context of the screen.

4.2 Mobile Claims Notification

Car drivers that sustain a collision with, e.g., a game animal have to inform their car insurance of the incident if they

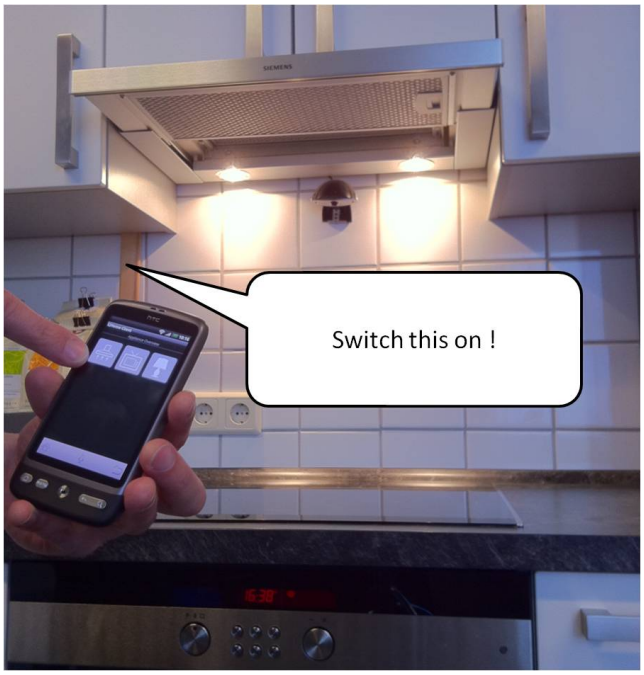

Figure 8: Multimodal dialogue remote control for the smart kitchen.

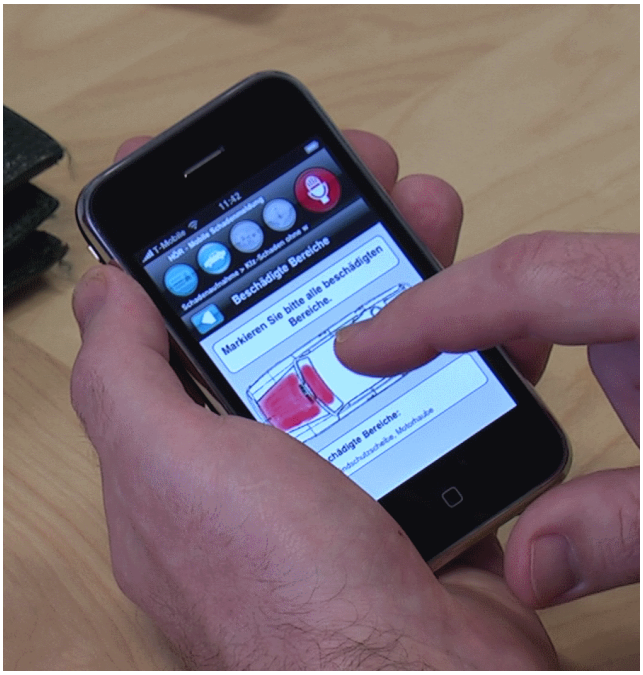

Figure 9: Multimodal dialogue user interface for mobile claims notification.

want to get the car repaired and a rental car in between without excessive costs. Usually, the initiation of such a rather complex and long-lasting business process takes some time, because various stakeholders participate in the process. Focussed on the claimant (the car owner), a mobile application allows an easy and fast initiation of the claims notification right from the location of the accident and further enables a user to participate and observe the subsequent claims adjustment. This is achieved by providing access to the back-end business process via a mobile multimodal dialogue user interface. Here, the GUI layout as shown in figure 9 reflects the relevant process structure from the claimant's perspective (identification, damage survey & claims notification, claims adjustment, and feedback) such that the current progress can be perceived immediately.

Technically, the mobile client runs on an iPhone and consists of a full-screen Web browser component for rendering the DHTML GUI and (analog to the remote kitchen control) a streaming component for sending and receiving audio data. The following dialogue phenomena are supported:

- **Multimodal Fusion:** Speech commands are combined with location information. A user can ask for the nearest repair shop and a route how to get there.
- **Elliptic Reference:** The context for an incomplete command can be reasoned from a previous turn. This allows the system to understand the following sequence of commands:
  "The front fender on the left is damaged."
  "And also the outside mirror on that side."
- **Mixed-initiative dialogue**: Since the application communicates with the back-end business process, it can receive notifications (*"Your car has been repaired."*) and information requests (*"Is your mobile phone number still valid?"*).

4.3 ISIS Information System

ISIS (Interacting with Semantic Information Services) is a multimodal dialogue information system that allows one or more users to interact with semantically annotated Web content from different sources (figure 10). In detail, it processes Wikipedia content via DBPedia, a music ontology and contact information from a personal addressbook. It supports natural language understanding of spoken language and typed input, pointing gestures, and graphical interactions. Additionally it allows to interact with multimedia content and web-based services like Google Maps and YouTube. The system interprets diverse input modalities and allows fast access to an ontological representation of extracted information. The client is running on a PC with touchscreen and renders its screen with HTML5. Again, several dialogue phenomena are supported:

- **Multimodal Fusion:** Speech commands are combined with pointing gestures. A user can point on an entry or a picture of a town and say: "Show this city on the map."
- **Elliptic Reference:** The context for an incomplete command can be reasoned from a previous turn. This allows the system to understand the following sequence of commands:
  "What is the name of Barack Obama's wife?"
  "Where is (s)he born?"
  "Show the city on the map."

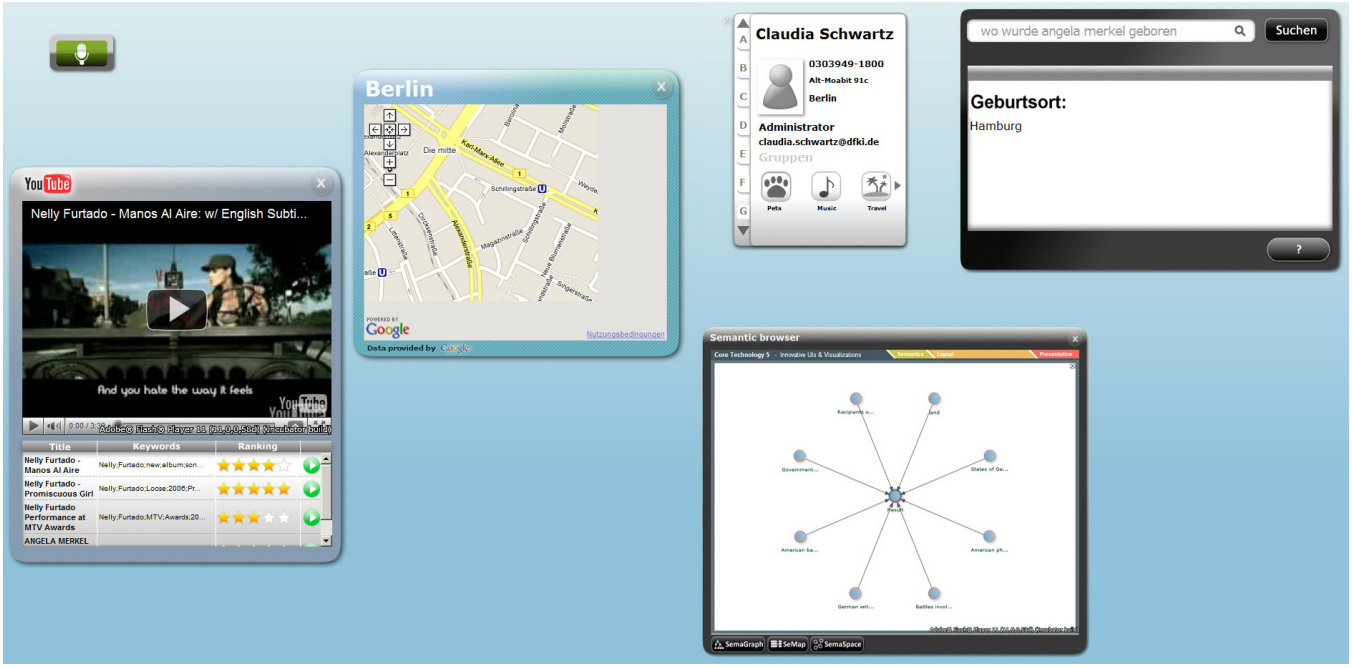

Figure 10: Graphical user interface of the multimodal dialogue ISIS information system.

- **Identical interactions with content from different services**: Because the semantic objects for person instances from Wikipedia and the personal addressbook are derived from a common base ontology, rules for task processing of content from one service work also with content from the the other one.

5 Conclusion & Future Work

The paper presents a model-driven development approach for multimodal dialogue systems. Semantic models are introduced that represent semantic content, referring expressions, user intentions and graphical user interfaces. These models act as a contract for the integration of new modalities and domains into multimodal dialogue applications. The model-driven approach allows a loose coupling of content, interaction act and presentation models and enables the rapid adaptation of content and behaviour, even during runtime. Deriving concepts for domain-specific content from well-chosen basic concepts still allow generic processes to operate in new domains. Reasoning on semantic content supports advanced dialogue phenomena. Similar techniques are used to fill context-dependent placeholders in patterns for user intentions and graphical representations. The development approach was adopted by three multimodal dialogue applications for different domains. We plan to evaluate the system by giving the authoring tools to some groups outside our own research group to collect more information about usability and the learning curve.

In the paper, we do not deal with models for dialogue management. Future work will extend the resolution of referring expressions with the option of asking clarifying questions to the user. Semantic models can help the system to identify missing information and to select suitable situation-adapted callbacks to the user. Semantic models and the connection to lexical databases like WordNet can be exploited to generate language models for speech generation and recognition. Future investigations will also include more fine-grained coordination of multimodal input and output. A further important point is a user model that should be taken into account in all steps of the dialogue process to provide a user-centered application with personalized appearance and behaviour.

6 Acknowledgement

Parts of this work have been carried out in the scope of the Theseus programme funded by the German Federal Ministry of Economics and Technology (01MQ07016) and the i2home project funded by the European Commission under the grant 033502. The opinions herein are those of the authors and not necessarily those of the funding agency. Special thanks go to Matthieu Deru.

References

Adida, B.; Birbeck, M.; McCarron, S.; and Pemberton, S. 2008. RDFa in XHTML: Syntax and processing – a collection of attributes and processing rules for extending XHTML to support RDF. W3C Recommendation. http://www.w3.org/TR/rdfa-syntax/.

Alexandersson, J.; Becker, T.; and Pfleger, N. 2006. Overlay: The basic operation for discourse processing. In Wahlster, W., ed., *SmartKom: Foundations of Multimodal Dialogue Systems*, Cognitive Technologies. Springer Berlin Heidelberg. 255–267. 10.1007/3-540-36678-4$_1$7.

Alexandersson, J.; Buschbeck-Wolf, B.; Fujinami, T.; Kipp, M.; Koch, S.; Maier, E.; Reithinger, N.; Schmitz, B.; and Siegel, M. 1998. Dialogue acts in verbmobil. Verbmobil report, DFKI, Saarbrücken.

Allemang, D., and Hendler, J. 2008. *Semantic Web for the Working Ontologist - Effective Modeling in RDFS and OWL*. Morgan Kaufmann.

Ankolekar, A.; Krötzsch, M.; Tran, T.; and Vrandecic, D. 2007. The two cultures: mashing up web 2.0 and the semantic web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, 825–834. New York, NY, USA: ACM.

Araki, M., and Funakura, Y. 2010. Impact of semantic web on the development of spoken dialogue systems. In *IWSDS*, 144–149.

Becker, T.; Blaylock, N.; Gerstenberger, C.; Kruijff-Korbayová, I.; Korthauer, A.; Pinkal, M.; Pitz, M.; Poller, P.; and Schehl, J. 2006. Natural and intuitive multimodal dialogue for in-car applications: The Sammie system. In *Proceedings of ECAI 2006, 17th European Conference on Artificial Intelligence, and Prestigious Applications of Intelligent Systems (PAIS 2006), Riva del Garda, Italy*, 612–616.

Bergweiler, S.; Deru, M.; and Porta, D. 2010. Integrating a multitouch kiosk system with mobile devices and multimodal interaction. In *ACM International Conference on Interactive Tabletops and Surfaces*, ITS '10, 245–246. New York, NY, USA: ACM.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web: Scientific American. *Scientific American*.

Bizer, C.; Lehmann, J.; Kobilarov, G.; Auer, S.; Becker, C.; Cyganiak, R.; and Hellmann, S. 2009. Dbpedia - a crystallization point for the web of data. *Web Semant.* 7(3):154–165.

Bolt, R. A. 1980. Put-that-there: Voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '80, 262–270. New York, NY, USA: ACM.

Bunt, H.; Kipp, M.; Maybury, M.; and Wahlster, W. 2005. Fusion and coordination for multimodal interactive information presentation. In *Multimodal Intelligent Information Presentation*, volume 27 of *Text, Speech and Language Technology*. Springer Netherlands. 325–339. 10.1007/1-4020-3051-7$_1$5.

Bunt, H.; Alexandersson, J.; Carletta, J.; Choe, J.-W.; Fang, A. C.; Hasida, K.; Lee, K.; Petukhova, V.; Popescu, A.; Soria, C.; and Traum, D. 2010. Towards an iso standard for dialogue act annotation. In *Proceedings of 7th International Conference on Language Resources and Evaluation(LREC-2010), May 19-21, Malta*.

Bunt, H. 2000. *Abduction, Belief and Context in Dialogue*, volume 1 of *Natural Language Processing*. Amsterdam, The Netherlands: John Benjamins. chapter Dialogue Pragmatics and Context Specification, 81–150.

Maybury, M. T., and Wahlster, W., eds. 1998. *Readings in intelligent user interfaces*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Milward, D., and Beveridge, M. 2003. Ontology-based dialogue systems. In *Proceedings of the 3rd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Acapulco, Mexico*, 9–18.

Neßelrath, R., and Alexandersson, J. 2009. A 3d gesture recognition system for multimodal dialog systems. In *Proceedings of the 6th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems (KRPD-09)*, 46–51. Pasadena, California, United States: IJCAI 2009.

Pfleger, N. 2004. Context based multimodal fusion. In *Proceedings of the 6th international conference on Multimodal interfaces*, ICMI '04, 265–272. New York, NY, USA: ACM.

Pfleger, N. 2007. *Context-based Multimodal Interpretation: An Integrated Approach to Multimodal Fusion and Discourse Processing*. Ph.D. Dissertation, Universtität des Saarlandes.

Porta, D.; Sonntag, D.; and Neßelrath, R. 2009. A Multimodal Mobile B2B Dialogue Interface on the iPhone. In *Proceedings of the 4th Workshop on Speech in Mobile and Pervasive Environments (SiMPE '09) in conjunction with MobileHCI '09*. ACM.

Schehl, J.; Pfalzgraf, A.; Pfleger, N.; and Steigner, J. 2008. The babbletunes system: talk to your ipod! In Digalakis, V.; Potamianos, A.; Turk, M.; Pieraccini, R.; and Ivanov, Y., eds., *ICMI*, 77–80. ACM.

Sonntag, D., and Möller, M. 2010. A multimodal dialogue mashup for medical image semantics. In *IUI '10: Proceeding of the 14th international conference on Intelligent user interfaces*, 381–384. New York, NY, USA: ACM.

Sonntag, D.; Engel, R.; Herzog, G.; Pfalzgraf, A.; Pfleger, N.; Romanelli, M.; and Reithinger, N. 2007. *SmartWeb Handheld - Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services (extended version)*. Number 4451 in LNAI. Springer. chapter 14, 272–295.

Sonntag, D.; Sonnenberg, G.; Neßelrath, R.; and Herzog, G. 2009. Supporting a rapid dialogue engineering process. In *Proceedings of the First International Workshop On Spoken Dialogue Systems Technology (IWSDS)*.

Sonntag, D.; Porta, D.; and Setz, J. 2010. Http/rest-based meta web services in mobile application frameworks. In *Proceedings of the 4nd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. (UBICOMM-10)*. XPS.

Wahlster, W., ed. 2006. *SmartKom: Foundations of Multimodal Dialogue Systems*. Berlin, Heidelberg: Springer.

Zimmermann, G., and Vanderheiden, G. C. 2007. The universal control hub: An open platform for remote user interfaces in the digital home. In Jacko, J. A., ed., *HCI (2)*, volume 4551 of *Lecture Notes in Computer Science*, 1040–1049. Springer.

Unsupervised clustering of probability distributions of semantic frame graphs for POMDP-based spoken dialogue systems with summary space

Florian Pinault and Fabrice Lefèvre
University of Avignon, LIA - CERI, France
{florian.pinault,fabrice.lefevre}@univ-avignon.fr

Abstract

Due to errors from the speech recognition and spoken language understanding modules, dialogue managers in spoken dialogue systems have to make decisions in uncertain conditions. In this paper a framework to interface efficient probabilistic modeling for both understanding and manager modules is described and investigated. After each speaker turn, a full representation of the user semantics is first inferred by SLU in the form of a graph of frames. This graph, after completion with some contextual information, is projected to a summary space into which a stochastic POMDP dialogue manager can perform planning of actions taking into account the uncertainty on the current dialogue state. Tractability is ensured by the use of an intermediate low-dimension summary space. To reduce the development cost of an dialogue system a new approach based on clustering is proposed to automatically derive the master-summary mapping functions. A preliminary implementation is presented in the MEDIA domain (touristic information and hotel booking) and several configurations are tested with a simulated user under varying noise conditions.

1 Introduction

During the last decade a lot of research have been done to shift the idea of learning optimal strategies for spoken dialogue systems from theory to practical ground. As a consequence it is now affordable to train policies on data from real-world corpora (collected either with online systems or by means of the Wizard-of-Oz setup) and is no more limited to toy examples. Anyhow trying to reproduce the human behavior using machine learning techniques generally entails to collect large corpus of spoken data and can be very costly. To address this issue efforts have been recently devoted to develop user simulators able to generate synthetic data with good characteristics (Georgila, Henderson, and Lemon 2005; Schatzmann and Young 2009).

However whenever enough data can be made available with respect to the number of trainable parameters, the size of the models itself remains an issue. Even in the case of simple slot-filling problems the number of possible dialogue states can be huge and thus makes their enumeration intractable. Despite regular improvements of training algorithm efficiency, compression of the dialogue state space remains the only viable approach as soon as several thousands states have to be accounted for: a function is elaborated which maps the initial (master) space to a compressed (summary) space. This latter is built so as to contain all and only the pertinent information for the decision-making process. Designing the mapping function is pretty difficult, requiring expert skill. Although some propositions emerge, none of the techniques proposed so far appeared a convenient general-enough answer to the underlying problem. They mainly rely on a prior structuring of the master space, for instance grouping states into partitions as in the HIS model (Young et al. 2010) or factorizing states in Bayesian networks (Thomson and Young 2010), which facilitate the definition of the summary space by means of simple features. One limitation of these approaches anyhow is that, to ensure good performance, they can only be applied to a task or domain with a well-defined ontology, either to form partitions iteratively or to define the Bayesian network structure. In our work an attempt is made to define efficient mapping functions without imposing a rigid structure for the master space.

Following the line of research initiated in (Lefèvre and de Mori 2007), we propose an unsupervised procedure to determine the summary space from the characteristics of the master space. In (Lefèvre and de Mori 2007), an automatic clustering process was applied directly at the level of flat semantic concepts (before any step of composition). In this paper a clustering process is also investigated but this time at the level of semantic frame graphs. These graphs incorporate all the information available to the system in the current turn (principled composition of all the fragmented pieces of information with contextual awareness). Clearly they represent a better candidate for the definition of the master dialogue space from which is obtained the summary space. Anyhow in this case the clustering procedure should be generalized to the case of n-best lists of frame graphs. To solve this issue, a distance is proposed to measure the similarity between 2 graphs which is extended afterwards in a new distance between n-best lists seen as probability distributions over the graphs of semantic frames.

After mapping functions have been defined to con-

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 48–53,
Barcelona, July 2011

vert master states to summary states, the master-summary POMDP framework also requires a final mechanism to transform the summary actions (broad dialogue acts such as `inform`, `request` etc) back to effective master actions (fully-specified dialogue acts such as `inform(name="Ibis Montmartre",phone="121322313")`, `request(location)` etc). Commonly a handcrafted heuristic is applied to find the most suitable pieces of information upon which the summary action can be applied. This heuristic can integrate a backoff mechanism to deal with un-matching situations, when no master action can be derived from the best hypothesised summary action (Gasic et al. 2009). In our system at each dialogue turn a set of rules is used to derive all the possible actions considering the current dialogue situation (with an associated score to manually guide the manager towards a desirable behavior). Then the POMDP, instead of picking up an action, is used to score all the proposed master actions based on its evaluation of the summary actions. Eventually a mixed policy is obtained from the weighted sum of the scores from the rules and the POMDP which allows to make a final decision in the master space. The effect of rule-based and POMDP decisions can thus be balanced and this allows to benefit from both approaches in an integrated way.

The main originalities of the developed approach can be summed up as:

- the spoken language understanding (SLU) module is based on a frame hierarchy and stochastic. In our case it implies that no strong assumption is made during the semantic extraction process on the domain ontology. No hard-coded constraints (such as logical rules) are involved. So users can express themselves more naturally and nonetheless be understood in the context of the task. Of course at the cost of providing the dialogue manager (DM) with rather complex and never observed structured semantic information.
- a standard POMDP model is used in the summary space: as a comparison, in the HIS approach, belief update is performed in the master space but then the planning in the summary space is based only on a MDP model (with the state being defined by few features from the master dialogue state, such as the probability of the best hypothesis, number of matching items in the database etc). In the proposed approach a belief tracking is performed in both the master and summary spaces.

This paper is organised as follows. Section 2 presents the semantic frame representation used as the SLU interface. In Section 3 a novel summary POMDP method is proposed. In Section 4, clusterings of semantic graphs and n-best lists are presented, including specific distance definitions. Practical application of the framework and results are finally analysed in Section 5 on a tourist information and hotel booking task.

2 Graphs of semantic frames

Dialogue managers in dialogue systems have to make decisions based on the available information about the user's

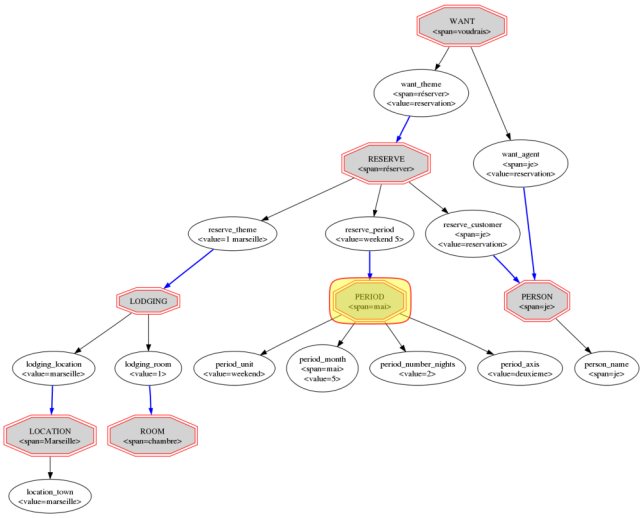

Figure 1: *Example of a semantic frame graph from the* MEDIA *corpus.*

goal. From the system point of view, the user's goal is a compound of specific pieces of semantic information gathered during the interaction process. Depending on the domain, this compound can be expressed through structures of various complexity: from simple flat (or parallel) slots to graphs of semantic frames.

Amongst the available semantic representations, the semantic frames (Lowe, Baker, and Fillmore 1997) are probably the most suited to the task of interest here, mostly because of their ability to represent negotiation dialogs in addition to pure information seeking. Semantic frames are computational models describing common or abstract situations involving roles, the frame elements (FEs).

The FrameNet project (Fillmore, Johnson, and Petruck 2003) provides a large frame database for English. As no such resource exists for French, we elaborated a frame ontology to describe the semantic knowledge of the MEDIA domain. As an illustration this ontology is composed of 21 frames (LODGING, HOTEL, LOCATION, PERIOD etc) and 86 FEs (Lodging_type, Hotel_facility etc), based on a set of around 140 elementary concepts (day, month, city, payment-amount, currency etc). All are described by a set of manually defined patterns made of lexical units and conceptual units (frame and FE evoking words and concepts). The training data are semi-automatically annotated by a rule-based process. Pattern matching triggers the instantiations of frames and FEs which are composed using a set of logical rules. Composition may involve creation, modification or deletion of frame and FE instances. After the composition step a graph of frames is associated to each training utterance. This process is task-oriented and is progressively enriched with new rules to improve its accuracy. Once a reference annotation is established the models of a two-step stochastic frame annotation process (Meurs, Lefèvre, and De Mori 2009) are trained and used afterwards in the system to annotate new utterances.

3 Summary POMDP

Following the idea of (Williams and Young 2005), the summary POMDP method consists in defining mapping functions from master spaces into summary spaces. A way to derive a fully-specified system action a_t from a summary action must be also defined.

Bold face notation will be used hereafter to distinguish between variables which are simple graphs (s_t, $u_t^\star$, a_t and o_t^i) and n-best lists of graphs ($\mathbf{o_t}$ and $\mathbf{b_t}$). The index t, time or turn number, will be dropped when not useful. All graphs considered here are frame graphs.

3.1 Master space

At the master level, which is the intentional level, the user generates an exact utterance $u_t^\star$ (unobserved), and the speech recognizer provides the system with n noisy versions of $u_t^\star$ along with some scores p_i as an n-best list:

$$\mathbf{o_t} = [(o_t^1, p_1), \ldots, (o_t^n, p_n)] \quad (1)$$

During a simulation the exact $u_t^\star$ can be known. In an annotated corpus, $u_t^\star$ is obtained from a reference annotation. In real conditions, $u_t^\star$ is not available. For perfect non-noisy speech recognition and understanding, $\mathbf{o_t} = [(u_t^\star, 1.0)]$.

From $u_t^\star$ (resp. $\mathbf{o_t}$), which depends on the current turn only, we define the cumulative state s_t (resp. $\mathbf{b_t}$) which depends on the full dialogue history. The update formulas, being essentially compositions of semantic structures, are considered parts of the SLU module and are only briefly described here.

The master state s_t is the exact dialogue state. Each utterance $u_t^\star$ is accumulated to a unique frame graph using the state-update formula:

$$s_t = Update_s(s_{t-1}, u_t^\star, a_t) \quad (2)$$

$Update_s$ includes a FSM which is used to maintain a grounding state of each piece of information. Moreover, the database is used to include in s_t the number of venues matching the information of s_t. The same method used in the SLU module to compose semantic tuples into graphs at the turn level (Meurs, Lefèvre, and De Mori 2009) is applied for the update operation between turns.

The master belief $\mathbf{b_t}$ represents the uncertainty on the current state s_t. As for states, each observation $\mathbf{o_t}$ is accumulated into a unique n-best list of frame graphs using the belief-update formula:

$$\mathbf{b_t} = Update_{\mathbf{b}}(\mathbf{b_{t-1}}, \mathbf{o_t}, a_t) \quad (3)$$

It follows that $\mathbf{b_t}$ is an n-best list of graphs and can be written as:

$$\mathbf{b_t} = [(b_t^1, q_1), \ldots, (b_t^m, q_m)] \quad (4)$$

$Update_{\mathbf{b}}$ uses an approximation similar to the one presented in (Young et al. 2010) to process the score of the n-best list. The update is performed as a cross product of the two lists (1) and (4): computing all $Update_s(b_t^i, o_t^j, a_t)$, associated with probability $q_i.p_j$. Then, identical graphs are removed and their weights summed, followed by a pruning and a re-normalisation step.

3.2 Summary space

Our summary POMDP uses two mapping functions M_s and M_o defining the summary state $\tilde{s}_t$ and observation $\tilde{o}_t$. They can be handcrafted as described in (Pinault, Lefèvre, and De Mori 2009) or learned by classifiers (clustering, see *infra*).

$$\tilde{s}_t = M_s(s_t) \quad (5)$$

$$\tilde{o}_t = M_o(\mathbf{b_t}) \quad (6)$$

Note that the summary observation $\tilde{o}_t$ is *not* computed from the master observation $\mathbf{o_t}$, but from the master belief $\mathbf{b_t}$.

The summary POMDP is defined as a classic POMDP on the states $\tilde{s}_t$ and observations $\tilde{o}_t$. In this POMDP, a summary belief $\tilde{\mathbf{b}}_t$ is monitored and represents a true distribution over $\tilde{s}_t$. The summary belief update is performed using a complete probability model learned from a corpus (transition and observation probabilities).

3.3 Summary to master actions

Not all system actions are possible at each turn depending on the current dialogue situation. Then some generic rules are used to generate the set of possible master actions.

The premises of the rules are clauses composed of logical connections of features extracted from the n-best list $\mathbf{b_t}$ (master belief), such as those described in (Pinault, Lefèvre, and De Mori 2009). The set of possible actions is data-driven and relies on the information available in the semantic structure of the master belief. To each master action is associated a summary action. Note that the action list could be easily constrained through the rules, whereby addressing the problem of VUI-completeness (Pieraccini et al. 2009).

3.4 Policy mixer

Adding scores to the rules defining the action list ensures a complete ordering of the action list in a very simplistic manner which allows to define an hand-crafted policy (referred to as *baseline* hereafter). The *baseline* score Q^0 for each master action does not depend on the belief and is designed using simple heuristics such as:

$$Q^0(\text{AskCity}) > Q^0(\text{AskContraints}) > Q^0(\text{AskDate}) \quad (7)$$

The summary POMDP policy π also provides scores but for summary actions (using the set of α-vectors (Spaan and Vlassis 2005) to approximate the Q-value function). These scores are transfered to the corresponding master actions (possibly several). As a result summary policies can be combined with the *baseline* policy through a linear policy mixer at the state-action value function level:

$$Q^{mixed}(\tilde{\mathbf{b}}, \mathbf{b}, a) = \lambda Q^\pi(\tilde{\mathbf{b}}, a) + (1 - \lambda) Q^0(\mathbf{b}, a)$$

4 Graph clustering

The design of the summary variables $\tilde{s} = M_s(s_t)$ and $\tilde{o} = M_o(\mathbf{b_t})$ is crucial to obtain an efficient compression of the master space (preserving the useful information to distinguish between dialogue situations requiring different system actions). In this work we propose to perform an unsupervised k-means clustering, as an alternative to the manual definition of features in M_s and M_o.

In this purpose, a distance between graphs (master states s_t) must be defined but also a distance between n-best lists of graphs (master beliefs $\mathbf{b_t}$). The notion of mean (or center) of a cluster is introduced too.

Let denote G the set of all graphs and B the set of n-best list of graphs for any $n \in \mathbb{N}$.

4.1 Graph edit distance

A widespread measure of similarity between graphs is the Graph Edit Distance (GED) (Gao et al. 2010), a generalisation of the string edit distance. The GED is defined as the shortest edition path to transform on graph into another, allowing 6 atomic editions: node and edge deletions, substitutions and insertions. We use the fast implementation of the GED as a binary linear programming problem, successfully used in (Justice and Hero 2006) for molecule classification with small graphs (of size < 30).

Let denote d_1 the GED between two master states s_t.

4.2 Belief space $\mathbb{B}$

In order to perform a k-means clustering, it is necessary to define the mean of a cluster. In this purpose, we use a space $\mathbb{B}$ for which the mean is well-defined as it has an addition and a scalar product. Then we identify B to $\mathbb{B}$ and G to a subset of B.

Let $\mathbb{B}$ be the set of all probability distributions over G, which are nonzero only for a finite number m of graphs, for any $m \in \mathbb{N}$. The mean of elements of $\mathbb{B}$ is well-defined and belongs to $\mathbb{B}$.

The bijection between B and $\mathbb{B}$ can be expressed as follows: for any $\mathbf{b} \in B$, written as (4), the probability distribution associated to $\mathbf{b}$ is $P_\mathbf{b} \in \mathbb{B}$ such that

$$P_\mathbf{b}(g) = q_i \quad \text{if } \exists i / g = b^i \tag{8}$$

$$P_\mathbf{b}(g) = 0 \quad \text{for any other } g \tag{9}$$

From this bijection, it follows that any $\mathbf{b} \in B$ has an associated $P_\mathbf{b} \in \mathbb{B}$, therefore the mean of a cluster of n-best lists of graphs is well-defined.

To embed G into $\mathbb{B}$, we use δ the canonical injection associating any graph g to δ_g, the Dirac distribution at point g, which is the n-best list with only one element of score 1:

$$g \mapsto \delta_g = [(g, 1.0)] \tag{10}$$

4.3 n-best list distance

It is then possible to define a distance d_2 on the space $\mathbb{B}$:

$$d_2(\mathbf{b}, \mathbf{b'}) = \inf_{X \sim P_\mathbf{b} \text{ and } X' \sim P_{\mathbf{b}'}} \mathbb{E}(d_1(X, X')) \tag{11}$$

with $\mathbb{E}$ the expectation. $X \sim P_\mathbf{b}$ denotes a random variable X which follows a probability distribution $P_\mathbf{b}$.

As the distance d_2 is too hard to compute directly, we approximate d_2 with the measure of similarity d_3, assuming the independence between X and X', defined as: (12), assuming independence between X and X':

$$d_3(\mathbf{b}, \mathbf{b'}) = \mathbb{E}(d_1(X, X')) = \sum_{i,j} q_i q'_j d_1(b^i, b'^j) \tag{12}$$

with $\mathbf{b}$ and $\mathbf{b}'$ are two beliefs, written as in (4).

Algorithm 1 Batch POMDP policy training

1: Collect transition and observation conditional probability tables from corpus C_i.
2: Learn optimal policy π_i with model-based POMDP training algorithm Perseus (Spaan and Vlassis 2005).
3: Generate new data using an ϵ-greedy policy π_i (mixed with *baseline*).
4: Merge new data with corpus C_i to create corpus C_{i+1}. Iterations are repeated until the observed reward function improvement is below a certain threshold.

If d_2 is a genuine distance (the proof is too long to be given here), d_3 is *not* a distance. But it has the advantage of being a linear function. Thus the "distance" d_3 to a cluster mean is the average of the "distances" d_3 to each point of this cluster.

5 Experiments and results

The task considered in our experiments is the MEDIA task which consists in informing about the prices of hotels with some constraints on facilities and making a reservation if any eligible venue is found. About 100k different user goals are possible involving 13 binary slots for facilities and 3 non-binary slots (location, price and date).

5.1 Evaluated policies

As described in Section 3, the summary variables $\tilde{s}_t$ and $\tilde{o}_t$ are extracted from the exact master state s_t and the observed n-best list $\mathbf{b_t}$ (master belief) using the mapping functions M_s and M_o. Two different configurations of POMDP systems are evaluated differing by the mapping functions.

POMDP with hand-crafted summary: *POMDP-HCsum*
In the POMDP-HCsum, M_s is factored into two features: $M_s(s) = (rules(s), db(s))$. The rule-based category $rules(s)$ can take 9 values. The number of database matches $db(s)$ can be 0, 1 or many.

M_o uses the same features applied to its first best hypothesis and possesses an additional feature: $M_o(\mathbf{b}_t) = (rules(b^1), db(b^1), entr(\mathbf{b}_t))$. The entropy feature $entr(\mathbf{b}_t)$ is binary (high/low).

Removing states never encountered in the corpus, the final summary system has only 18 states and 36 observations (and 8 actions).

POMDP with clustered summary: *POMDP-clusterSum*
In the *POMDP-clusterSum* system, $M_s(s_t)$ and $M_o(\mathbf{b}_t)$ are defined by an unsupervised k-means clustering using the distances d_1 and d_3 defined in Section 4. The data consist in 10k graphs from simulated dialogues with the *baseline* system.

Costs associated to edition operations are chosen in order to emphasize, in this order: the information from the database, the grounding states and some frame/FE known to be relevant for the tasks. This human decision can be avoided by using uniform edition costs.

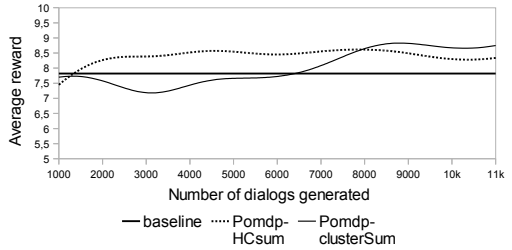

Figure 2: *Average reward during training epochs.*

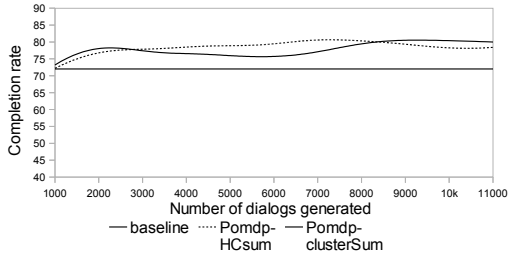

Figure 3: *Task completion rate during training epochs.*

Due to the intense computation cost of linear programming for the GED, only 14 state clusters and 10 observation clusters have been used (still with 8 actions).

5.2 Policy training

The summary POMDP policies are optimized iteratively from an initial corpus C_0 created using the *baseline* policy, as described in Algorithm 1.

At each iteration, 1000 dialogs are generated using the agenda-based user simulator from (Schatzmann and Young 2009) on which the slot-based representation have been replaced by a semantic frame-based representation. Exploration coefficient (ϵ) is set to 0.1 and the noise level is 0.5 (when not stated otherwise). Rewards are -1 at each dialogue turn and +20 for a final success. λ in policy mixer of Eq. 8 is set to balance the score dynamics between the *baseline* and the used POMDP policy so as to put them on a par.

5.3 Evaluation

Evolution of the average reward and completion rate along training epochs are presented in Figure 2 and 3. Iteration step is 1000 dialogues.

It can be observed that both POMDP policies give a clear improvement over the *baseline* alone in terms of average reward and task completion rate. Both policies seem to have comparable behavior even though on the first iterations, the handcrafted *POMDP-HCsum* performs slightly better than the automatic *POMDP-clusterSum*. A possible reason is the influence of the initial corpus generated with the *baseline* policy and thus more coherent with the design choices of the HC summary space.

Table 1 is populated with task completion rates and average rewards measured on 10k new dialogues. The two policies used are those obtained after 10 iterations (trained with

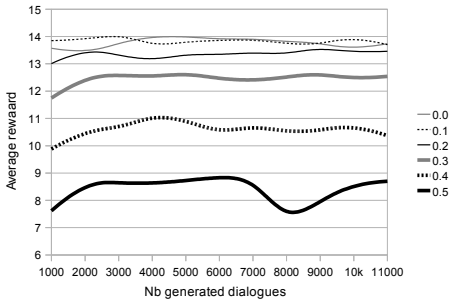

Figure 4: *Average reward of HCsum for error rates varying from 0.0 to 0.5.*

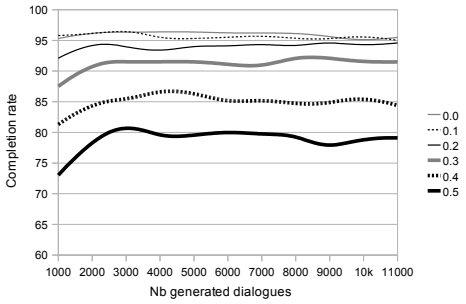

Figure 5: *Task completion of HCsum for error rates varying from 0.0 to 0.5.*

11000 generated dialogues). The results confirm the interest of using summary POMDPs for dialogue systems (gains are around +1 on rewards, 7-8% on completion rates at the expense of a small increase in dialogue lenths 6%) and show that automatic compression of master space is a pertinent alternative to costly and complex expert design.

Influence of the noise level Figures 4 and 5 show the ability of the POMDP policy to outperform the baseline under noisy conditions. The plots were generated with noise level in the user simulator varying from 0 to 0.5 (more details on how the noise is introduced in the simulations can be found in (Schatzmann and Young 2009)) and an exploration coefficient of 0.1.

Influence of the numbers of summary clusters Figures 6 and 7 are generated with a noise coefficient of 0.5 in the user simulator and the exploration coefficient is still 0.1.

The plots tend to show that relevant information cannot be completely represented into too few automatic states. Only 6 states (and 6 observations) as summary clusters is not enough to keep all the information : the $6/6$ policy does not perform as well as the others. Nevertheless it still beats the baseline, showing that the summary POMDP is able to give good advices about the situation handling cautiously the information it has been provided with.

The *clusterSum* systems which use a larger number of clusters (14 states and 10 observations, or 27 states and 19 observations) perform as well as the *HCsum* system which uses an handcrafted summary mapping. All relevant information extracted in the HC summaries seem to be extracted

Policy	Task completion rate	Average reward	Average length
baseline	72.0	7.8	7.17
POMDP - HC Summary Space	78.9	8.7	7.78
POMDP - Clustered Summmary Space	80.3	9.2	7.62

Table 1: *Evaluation on 10k dialogues.*

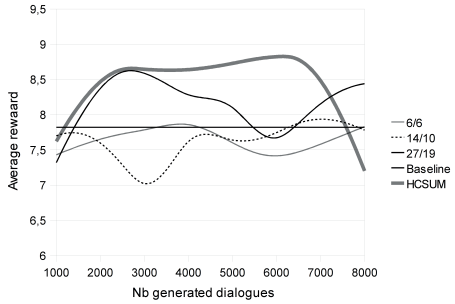

Figure 6: *Average reward of clusterSum during training epochs for different cluster numbers.*

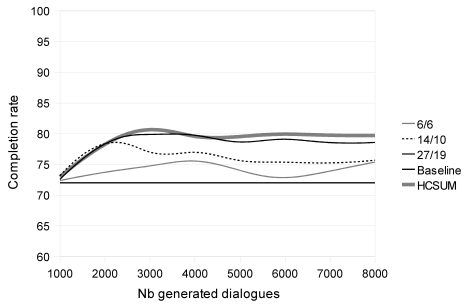

Figure 7: *Task completion rate of clusterSum during training epochs for different cluster numbers.*

also with $14/10$ clusters, adding more clusters does not improve further the performance.

6 Conclusion

In this paper we proposed and investigated several ways to interface a rich semantic representation with a POMDP-based dialogue manager. The manager is mixing ruled-based and POMDP policies. The summary POMDP is based on n-best lists of observations using either handcrafted mapping functions or automatically derived functions from clustering of the frame graphs with appropriate distances. Experiments with a simulated user showed a performance improvement when using a summary POMDP compared to using only a manually defined policy, and the policy based on automatic compression of the dialogue state performs as well as one based on a handcrafted summarisation. Evaluation with real users are in progress, and the preliminary results (with 20 users) tend to confirm the results with simulated users.

References

Fillmore, C. J.; Johnson, C. R.; and Petruck, M. R. 2003. Background to framenet. *International Journal of Lexicography* 16(3):235–250.

Gao, X.; Xiao, B.; Tao, D.; and Li, X. 2010. A survey of graph edit distance. *Pattern Analysis and Applications* 13(1):113–129.

Gasic, M.; Lefèvre, F.; Jurcicek, F.; Keizer, S.; Mairesse, F.; Thomson, B.; Yu, K.; and Young, S. 2009. Back-off action selection in summary space-based pomdp-based dialogue systems. In *IEEE ASRU*.

Georgila, K.; Henderson, J.; and Lemon, O. 2005. Learning User Simulations for Information State Update Dialogue Systems. In *Eurospeech*.

Justice, D., and Hero, A. 2006. A binary linear programming formulation of the graph edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(8):1200 –1214.

Lefèvre, F., and de Mori, R. 2007. Unsupervised state clustering for stochastic dialog management. In *IEEE ASRU*.

Lowe, J.; Baker, C.; and Fillmore, C. 1997. A frame-semantic approach to semantic annotation. In *SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*

Meurs, M.; Lefèvre, F.; and De Mori, R. 2009. Spoken language interpretation: On the use of dynamic bayesian networks for semantic composition. In *ICASSP*.

Pieraccini, R.; Suendermann, D.; Dayanidhi, K.; and Liscombe, J. 2009. Are we there yet? Research in commercial spoken dialog systems. In *Text, Speech and Dialogue TSD*.

Pinault, F.; Lefèvre, F.; and De Mori, R. 2009. Feature-based summary spaces for stochastic dialogue modeling with hierarchical semantic frames. In *Interspeech*.

Schatzmann, J., and Young, S. 2009. The hidden agenda user simulation model. *IEEE Trans. Audio, Speech and Language Processing* 17(4):733–747.

Spaan, M., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for pomdps. In *JAIR*.

Thomson, B., and Young, S. 2010. Bayesian update of dialogue state: a POMDP framework for spoken dialogue systems. *Computer Speech and Language* 24(4):562–588.

Williams, J., and Young, S. 2005. Scaling up pomdps for dialog management: The "summary pomdp" method. In *IEEE ASRU*, 177 –182.

Young, S.; Gašić, M.; Keizer, S.; Mairesse, F.; Schatzmann, J.; Thomson, B.; and Yu, K. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language* 24(2):150–174.

Subjective and Objective Evaluation of Conversational Agents in Learning Environments for Young Teenagers

Annika Silvervarg, Arne Jönsson
Deprtment of Computer and Information Sceinec
Linköping University, Linköping, Sweden
annika.silvervarg@liu.se, arne.jonsson@liu.se

Abstract

In this paper we present results from a study of subjective and objective evaluation metrics used to asses a conversational agent. Our study has been conducted in a school setting with students, aged 12 to 14 years old, who used a virtual learning environment that incorporates social conversation with a pedagogical agent. The subjective evaluation metrics capture the students' experiences of different aspects of the conversations, while the objective evaluation metrics are based on an analysis of the logs of the actual conversations.

Our results show that there are no correlations between subjective and objective metrics that are supposed to measure the same aspects, for example, to what extent the system can correctly interpret and give responses to user utterances. They also indicate that different categories of users need to be considered, for example based on their attitude towards or engagement in the system.

Introduction

We are developing a learning environment to be used by 12 to 14 year old students. The learning environment includes an embodied agent capable of both task-directed and social interaction with users. The starting point is an existing educational math game (Pareto 2004), in which children train basic arithmetic skills through board games that intertwine game play with learning content through visualizations of arithmetic operations. A crucial part of the game is a pedagogical agent, more specifically a Teachable Agent (TA) (Biswas et al. 2001). The TA is a peer rather than a tutor and the student's goal is to teach the agent to play the game. This is mainly done by responding appropriately to different multiple-choice questions posed by the agent during game play, which is called the on-task dialogue. Each question have four candidate answers, one correct, two incorrect, and one "I do not know". These questions are the basis for teaching the agent how to play the game.

A novel part of the learning environment is the ability to have a social conversation with the teachable agent, called off-task dialogue. The off-task conversation is a socially oriented chat-like written conversation where the agent and the student can discuss both domain-oriented topics, such as school and math, and off-domain topics like music, friends and family. Reasons for inclusion of such a conversational mode is to increase overall engagement and receptivity of the students (Cooper and Baynham 2005), to improve recall of the learning material through emotional engagement (Hamann 2001), to promote trust and rapport-building (Bickmore 2003), and to make students feel more at ease with a learning task or topic (Kim et al. 2007). A previous study of the learning environment by Gulz, Haake, and Silvervarg (2011) showed trends that indicate that students who played the game with off-task interaction had a more positive experience of the game and that they also learnt more, as reflected in the learning outcomes of their teachable agents.

The system uses the metaphor of regular breaks between lessons in school for switching between on-task activities (i.e. playing the game and on-task dialogue) and off-task activities (i.e. social conversation), see Figure 1 for screen shots of the system. Thus, the conversation in our learning environment has a different purpose from those in traditional intelligent tutoring systems, where the conversational agent often acts as a teacher that guides the user through a task, cf. (Graesser et al. 2005; Litman and Forbes-Riley 2006). Our agent has more in common with virtual humans as described by e.g. Traum et al. (2008), in that it combines social conversation with some task-oriented aspects. As a consequence, the knowledge representation and processing of the dialogue can be less extensive and simpler than in, for instance, traditional task-oriented or tutorial dialogues.

The aim of this paper is two-fold; to evaluate the conversational skills of the agents as perceived by the specific user group of young teenagers, i.e 12 to 14 year old students, and to investigate and compare different evaluation metrics. We do this by performing both a subjective evaluation, based on questionnaires, and an objective evaluation, based on tagged dialogue logs, and by investigating how the objective and subjective metrics correlate. We first present previous work on objective and subjective measures for evaluation of dialogue system and chatbots, then we describe the off-task conversational abilities of our agent, and finally present and discuss our own empirical findings.

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 54–63,
Barcelona, July 2011

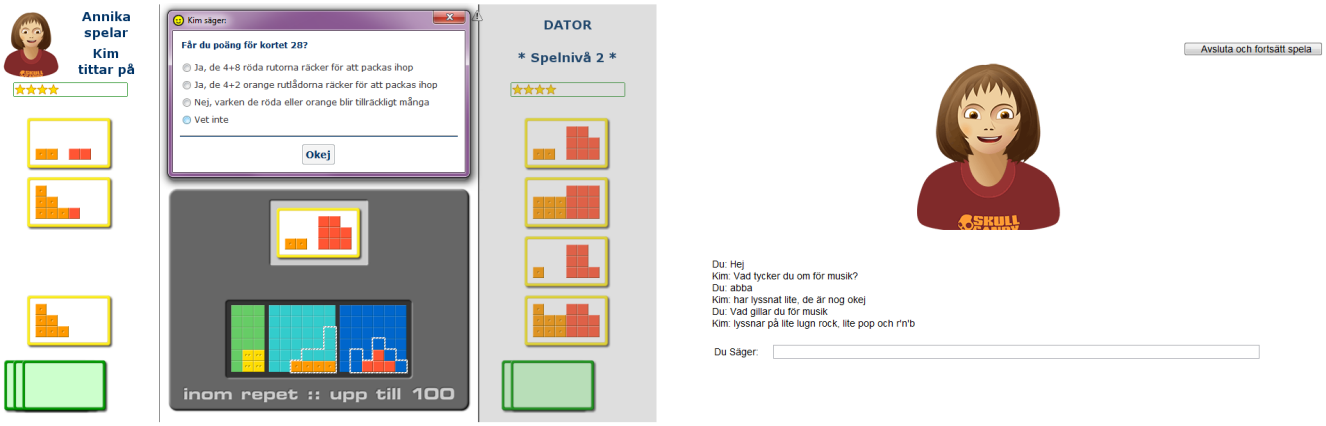

Figure 1: Screenshot of the educational system. On the left side is a screen shot of the educational math game where the agent has asked a multiple choice on-task question. On the right side is a screen shot of the agent engaged in off-task social conversation.

Subjective and objective evaluations of dialogue systems

Evaluation of dialogue systems is mainly done either by distributing a questionnaire to the users trying to the reveal their subjective assessment of using the dialogue system or by studying the resulting dialogue. Artstein et al. (2009) call it ”soft” numbers versus ”hard’ numbers and propose a ”semi-formal” evaluation method combining the two.

PARADISE (Walker et al. 1998), is one prominent evaluation framework that tries to capture both these perspectives for task-based interactions by combining user satisfaction, task success, and dialogue cost into a performance function. Studies using PARADISE indicate, for instance, that interaction quality is more important than efficiency (Walker, Kamm, and Litman 2000). They also show that there indeed are certain factors that correlate to user satisfaction for task oriented dialogues, but that these do not account for all factors correlating to user satisfaction. They show, for instance, that elapsed time is not a good predictor of user satisfaction (Walker, Boland, and Kamm 1999). PARADISE is developed for task-oriented interactions and requires controlled experiments (Hajdinjak and Mihelič 2006).

For non-task interactions, other factors than task success and dialogue cost are important to achieve user satisfaction, e.g. naturalness. Hung et al. (2009) present a variety of methods that aim at capturing naturalness in interactive conversational agents, or chatbots. Their final method is a framework derived from PARADISE.

Chatbot evaluations have also been conducted using a variety of instruments that tries to combine subjective and objective measures. Shawar and Atwell (2007), for instance, present such investigations, but they never assess the quality of the different measures. Instead they conclude that evaluations should be adapted to user needs and the application at hand.

Foster, Giuliani, and Knoll (2009) performed a study of subjective and objective evaluation measures using a robot dialogue system for task assembly. They used the objective measures: dialogue efficiency, dialogue quality (interaction problems and number of gazes at the objects and robot) and task success. The subjective measures were collected in a questionnaire on opinions of the robot as partner, instruction quality, task success and feelings. Contrary to Walker et al. (1998) they found that dialogue length has a positive contribution to user satisfaction. This is probably because the system is not supposed to find information as efficient as possible, and also because robot-interaction is new and enjoyable. They also found that repetitions lowered user satisfaction. They conclude that user satisfaction factors were affected by other factors than those conveyed in the objective measures.

The Conversational Agent

The conversational dialogue system used in our studies performs off-task conversations with students in the context of an educational math game (Silvervarg, Gulz, and Sjödén 2010). The conversational capabilities of the agent have been developed through an iterative process of design – implementation – evaluation, in close cooperation with the target user group, 12 to 14 year old students. Each iteration has led to addition of more topics and extended vocabulary, but also more sophisticated strategies for dialogue management (Silvervarg and Jönsson 2010).

In the current version the agent is able to handle greetings, ask and respond to questions and statements on various topics, ask and respond to follow up questions, deal with utterances outside its domain of knowledge, handle flaming/hazing/abuse and other out of scope utterances, know when and how to change topic, and to tell mini narratives, see Figures 2 and 3 for example dialogues[1].

The off-task conversation is implemented using AIML, Artificial Intelligence Markup Language (Wallace 2010),

[1]All dialogues have been translated to English by the authors, misspellings and ungrammatical constructs are intentionally kept to better reflect the original.

User1:	hi
Kim1:	hi
User2:	what are you doing
Kim2:	talking to you
User3:	hahah that's a good one
Kim3:	I do not understand, can you ask something else
User4:	aa
Kim4:	I am so happy today, when I woke up I heard one of my favorite songs on the radio, and it always puts me in a good mood, how are you today?
User5:	do you have a girlfriend?
Kim5:	nope
User6:	ok
Kim6:	how are you?
User7:	it's fine, you
Kim7:	ok
User8:	what do you usually do in the weekends?
Kim8:	can we talk about hobbies instead?
User9:	okey
Kim9:	what do you do during the weekends?
User10:	I practice with the choir ;) you?
Kim10:	aha hang out with my friends
User11:	it's only this weekend that I have practice with the choir
Kim11:	did not understand, what?
User12:	nothing
Kim12:	ok
User13:	do you have an animal
Kim14:	I did not get that, can you ask something else
User15:	so you have a dog?
Kim15:	yep
User16:	what's it's name
Kim17:	Tessy
User18:	okey
Kim18:	What type of music do you like?
User19:	I like pop and r&b you?
Kim19:	mhm
User20:	What music do you like?
Kim20:	I like rnb and easy rock
User21:	okey
Kim21:	I listen to rock and rnb I think that's the sweetest music
User22:	like I di
Kim22:	I think coldplay are so amazingly good, who's your favorite artist?
User23:	I like , b.o.b coldplay ,akon chrisbrown
Kim23:	mhm okey mhm

Figure 2: Example dialogue where the user has very positive opinions of the conversational agent.

User1:	hi there u doin
Kim1:	talking to you
User2:	okey how old are you?
Kim2:	I am 12 year
User3:	want to go steady with me?
Kim3:	I do not understand, ask something else
User4:	are you stupid?
Kim4:	Haha come on, lets talk about something else
User5:	are you gay?
Kim5:	can we talk about hobbies instead?
User6:	do you like the number 6?
Kim6:	did not understand, what?
User7:	can we hang out today?
Kim7:	hm, lets talk about something else
User8:	diss!!!!!!!!!!!!!!!!
Kim8:	this weekend I am going to the movies. I like to forget about everything and disappear into another world, in case it's a good movie. what about you?
User9:	a together with your mom
Kim9:	okey

Figure 3: Example dialogues where the user has very negative opinions of the conversational agent.

more specifically the open source java-based interpreter called programD. The interpreter supports the use of substitutions, i.e. the opportunity to exchange words or phrases in the user input with other words or phrases. Otherwise only standard AIML has been used.

AIML works on the surface level of utterances, and map user utterances, <*pattern*>, to system responses, <*template*>, which together form a <*category*>. Patterns can consist of words, which in turn consist of letters, numerals, and the wildcards _ and *, which function like words. Templates consist in their simplest form of only plain text. However, a template can include AIML tags that allow it to set or get data in variables and predicates, give conditional responses, choose a random response from a set of responses, or recursively call the pattern matcher to insert the responses from other categories. AIML also allows for handling a limited context through the optional tags <*that*>, which refers to the systems last utterance, and <*topic*>, which can span multiple exchanges.

To deal with the variation in user input, synonyms are handled using substitutions and grammatical variants through several different patterns for the same type of question and topic. The agent's replies are often randomly chosen from a set of 3-5 variants. To be able to correctly respond to follow-up questions and answers to questions posed by the agent, <*that*> and <*topic*> are used. To deal with recurring types of utterances, such as greetings, hazings, and flamings a number of variables are used to keep track of repetitions. To be able to choose new topics the agent has a topic model implemented as a set of AIML predicates including 17 topics that are linked to questions or narratives.

The conversational behaviour is described by a dialogue grammar. The dialogue acts used for the conversation differ from task-oriented dialogue acts, c.f. Bunt et al. (2010), as our agent is not supposed to carry out a task as efficiently as

possible, nor are tutoring-specific dialogue acts, c.f. Litman and Forbes-Riley (2006), applicable as the teachable agent do not have the traditional role of a tutor, and the conversation is more socially oriented. The conversational behaviour more resembles that of virtual humans (Traum et al. 2008) and combine dialogue acts that are task-related as well as more socially oriented. They comprise: Gr (Greeting), Q (Question), A (Answer), Ack (Acknowledgement), Follow Up (FU), Narrative (N), Not Understood (NU), Not Understood Answer (NUA), Abuse (Ab), Abuse Answer (AbA), and Laughter (L). Figure 4 depicts the dialogue grammar based on the dialogue capabilities and dialogue acts described above. Aspects of dialogue behaviour is described in more detail in the following sections.

Greet ::= Gr_U Gr_A [Gr_U (AgentQ|AgentN)]
AgentN ::= N_A [Ack_U AgentQ]
AgentQ ::= Q_A A_U [AgentAck]
AgentQ ::= Q_A A_U [Ack_A UserFU]
AgentQ ::= Q_A A_U FU_A [UserAck]
AgentQ ::= Q_A UserAFU
UserAFU ::= A_U FU_U A_A [UserAck]
UserFU := FU_U A_A [UserAck]
UserQ ::= Q_U A_A [UserAck]
UserQ ::= Q_U AgentAFU
AgentAFU ::= A_A FU_A A_U [AgentAck]
UserAck ::= Ack_U AgentAck
AgentAck ::= Ack_A [Ack_U AgentN| AgentQ]
Abuse ::= Ab_U AbA^1_A [Abuse2]
Abuse2 ::= Ab_U AbA^2_A [Abuse3]
Abuse3 ::= Ab_U (AgentN|AgentQ) [Abuse4]
Abuse4 ::= Ab_U AbA^4_A
NotUnderstand ::= NU_U NUA^1_A [NotUnderstand2]
NotUnderstand2 ::= NU_U NUA^2_A [NotUnderstand3]
NotUnderstand3 ::= NU_U NUA^3_A [NotUnderstand4]
NotUnderstand4 ::= NU_U|Ack_U AgentN|AgentQ
Laughter ::= L_U L_A [L_U AgentN|AgentQ]

Figure 4: Dialogue Grammar. The dialogue grammar describes the agent's conversational behaviour. The subscript letter is used to indicate if the user (U) or the agent (A) performs the dialogue act and the superscripts (1, 2, 3, 4) on the agent responses to Abuse and NotUnderstand utterances indicate that the response comes in different variants. Brackets, [ ], are used for optional items and alternatives are separated by |.

Initiative

The agent is designed to allow for a mixed-initiative dialogue where both agent and user can initiate new topics and ask general follow up questions, e.g. "what about you". A new topic is initiated in situations where interaction is repeated or the user does not take the initiative. For example, when greetings, acknowledgments or laughters are repeated the agent asks a question or tells a mini narrative, as exemplified in Figure 5.

The agent can also take the initiative when the user repeatedly abuse Kim, or there have been many consecutive user utterances that have not been understood, as described in the two following sections.

User:	Hi (Gr_U)
Kim:	Hi (Gr_A)
User:	Hello (Gr_U)
Kim:	What type of music do you like? (Q_A)
User:	much u? (AFU_U)
Kim:	everything but dance band music hehe (A_A)
User:	hehe (L_L)
Kim:	lol (L_L)
User:	lol (L_L)
Kim:	I listen to rock and rnb I think that's the sweetest music (A_N)

Figure 5: Example of Kim taking the initiative. Dialogue acts are shown in parenthesis after each utterance.

Repair strategy

A recovery strategy to deal with the utterances that the system is not able to interpret correctly is needed to help the user understand what they can talk about and avoid loosing track of the conversation. The strategy used is similar to the one used by SGT Star (Artstein et al. 2009); first a simple *clarification* then, a *stall*, followed by a *move on*, and finally a *prompt* that directs the user to a new topic. In our system the agent stall is a general prompt for the user to initiate a new topic, if this fails the agent proposes a topic, and as a last resort takes the initiative and asks a question or tells a narrative, Figure 6.

User:	fgd (NU_U)
Kim:	I did not understand that (NUA^1_A)
User:	cb (NU_U)
Kim:	I did not catch that, can we talk about something else? (NU^2_A)
User:	s (NU_U)
Kim:	Lets talk about school (NUA^3_A)
User:	s (NU_U)
Kim:	My school is okey, a little to much homework, wbu? (NU^4_A)

Figure 6: Example of how Kim handles utterances that are not understood.

Hazing/Flaming/Testing strategy

In the dialogues in the corpus, hazing, testing and flaming occurs rather frequently, for example comments and questions regarding the agent's appearance, intellectual capacities, and sexual orientation and activities. To fend this off, the agent recognizes such utterances and uses a strategy similar to the repair strategy where the two first responses are general, followed by an attempt to change the subject, see Figure 7.

Subjective and objective evaluations

The conversational agent was objectively and subjectively evaluated at a Swedish School, where students from three classes, 12-14 years old, used the learning environment with the conversational agent during three lectures. Data from

User:	you are ugly (Ab_U)
Kim:	eh, lets talk about something more interesting (AbA_A^1)
User:	you smell (Ab_U)
Kim:	come on, lets talk about something fun (Ab_U^2)
User:	ugly (Ab_U)
Kim:	Lets talk about music (Ab_A^3)
User:	ugly (Ab_U)
Kim:	Stop that, or I will tell a teacher (Ab_U^4)

Figure 7: Example of how Kim handles abuse.

questionnaires and dialogue logs from a total of 19 students, 8 females and 11 males, were used in this study. The students played the game for about a total of 120 minutes and after every second game session a break was offered. During the first three breaks the students had to chat with the agent until the break ended, after that chatting was optional.

Subjective evaluation - Questionnaire

After the final session a questionnaire was distributed to the students. The questionnaire is partly based on SASSI (Subjective Assessment of Speech System Interfaces) (Hone and Graham 2000) and CCQ (The Communication Competence Questionnaire) (Monge et al. 1982). It consists of Likert items scaled from 1 (Strongly disagree) to 7 (Strongly agree), see Table 1. The questionnaire items were chosen to capture aspects of the agent's conversational abilities, e.g. that the agent understood user utterances and could give correct responses as well as the users' experience of conversing with the agent, e.g. naturalness and likeability.

Objective evaluation - Dialogue Coding Scheme

To objectively evaluate the agent's conversational abilities we analyzed the logs of the conversations. The coding scheme used is based on the coding schemes used by Robinson, Roque, and Traum (2010) to evaluate virtual humans. It has a set of codes characterizing the user's dialogue action and another set of codes that evaluates the agent's responses. For the investigations presented in this paper we only use a subset of the codes in the top layer used by Robinson, Roque, and Traum (2010) since our focus is on the quality of the agent's answers and we thus have no need to further differentiate the different utterances made by the users. See Table 2 for the categories and descriptions of how they are manifested in our corpus.

Each user utterance is coded with one of the categories in Table 2. The agent's responses are categorized as 3, 2, RR, or 1, Table 3, depending on the appropriateness of the response. It is a modified version of the coding scheme used by Robinson, Roque, and Traum (2010), since we do not have any instances of "No response" so we end up with four instead of five values for the agent's response.

The highest value 3, i.e. appropriate response, means that the agent understood the user and responded correctly. Request Repair, value RR, is when the system does not understand and asks for a clarification or request that the user changes topic. Partially appropriate, value 2, is typically used when the user's utterance is not understood by the agent, and the agent response is to initiate a new topic, see the dialogue in Figure 8. Inappropriate response, value 1, is when the system responds erroneously, typically because it has mis-interpreted the user's utterance.

For coding of the agent's utterances we also use a cate-

Table 2: Dialogue action codes.

Code	Description
D	General dialogue functions, e.g. Greeting, Closing, Politeness
H	Hazing, Testing, Flaming, e.g. Abusive comments and questions
Q	Information Request, e.g. Questions to the agent
R	Requests, e.g. Comments or questions that express that the user wants help or clarification
A	Answer to agent utterances
S	Statements
G	Incomprehensible,e.g. Random key strokes or empty utterances

Table 3: Agent response values.

Code	Value
3	Appropriate response
2	Partially appropriate
RR	Request Repair
1	Inappropriate response

Speaker, Utterance		Code
User	are you free on tuesday	Q-RR
Kim	I did not understand what you said, can we talk about something else	
User	hi	D-3
Kim	Hi	
User	are you good at school	Q-3
Kim	rather good	
User	neerd	H-3
Kim	maybe we can talk about hobbies instead	I
User	I like football	S-RR
Kim	what, how do you mean?	
User	I like kicking the balls	S-RR
Kim	hmm, I have difficulties understanding what you say, can we talk about something else	
User	do you have a problem with Swedish	Q-2
Kim	this weekend I am going to the movies. It is great to forget everything and just disappear into another world, at least if it is a good film, how about you?	I
User	may I join	Q-2
Kim	What are you doing at the weekends?	I

Figure 8: Example dialogue with taggings from the coding scheme.

Table 1: Questionnaire items and statistics from the evaluation. $n(7)$ denotes the number of highest score, $n(1)$ the number of lowest score, M denotes the average and SD the standard deviation. The scores on the negatively loaded items (13-17) were transformed so that a high score is positive for the dialogue system and a low score is negative for the system.

Questionnaire item	N	n(1)	n(7)	M	SD
1. Kim's answers often surprised me	19	5	3	4.05	2.27
2. Kim understood what I said	19	5	2	3.37	2.01
3. I could fix misunderstandings if I wanted to	19	4	7	4.79	2.39
4. Kim was a good listener	19	6	5	4.05	2.48
5. I would like to talk to Kim again	19	3	5	4.32	2.29
6. Kim expresses her ideas very clearly	19	4	5	4.47	2.27
7. Kim mostly says the right thing at the right time	19	4	4	4,05	2,32
8. Kim is easy to talk to	19	3	5	4.37	2.03
9. I liked to talk to Kim	19	3	5	4.37	2.22
10. I could control the interaction with Kim	19	3	5	4.42	2.17
11. It was easy to understand how to talk so that Kim should understand	19	3	3	4.00	2.13
12. It felt natural to talk to Kim	19	5	2	3.79	2.25
13. Sometimes I lost track of the conversation	19	2	4	4.37	1.92
14. It was frustrating to talk to Kim	17	1	7	5.12	2.06
15. It was hard to know what to talk about with Kim	19	3	4	4.12	2.23
16. Kim often repeated herself	19	12	1	2.05	1.78
17. Sometimes I wondered if I used the right word	19	1	7	4.37	2.22
18. I always knew what I could say to Kim	18	5	4	4.17	2.38

Table 4: Mapping of subjective and objective measures. N is the total number of agent utterances and $n(x)$ denotes the number of utterances tagged as category x, $|$ denotes *or*, and $X - Y$ denotes a turn-taking, e.g. $n(Q-3)$ denotes the number of user questions, Q, followed by a correct agent response, 3.

Description	Questionnaire	Dialogue rating
Correct interpretation	Q2	$\frac{n(3)}{N-n(G)}$
Correct response	Q7	$\frac{n(3)+n(2)+n(RR)}{N}$
Repetition	Q16	$\frac{N-n(REP)}{N}$
Control	Q10	$\frac{N-n(I)}{N}$
Coherence	Q13	$\frac{N-n(Q-1\|S-1\|RR)}{N}$
Habitability	Q11, Q18	$\frac{n(D-3\|Q-3\|S-3)+n(D-2\|Q-2\|S-2)}{n(D\|Q\|S)}$

gory for agent initiatives, I, and one for repeated agent utterances, REP. The category I is used *only* when the system deliberately takes control of the interaction from the user, for example, posing a question on a new topic after a repeated sequence of user utterances that the agent is unable to interpret, see Figure 6. For a sequence of abuse, see Figure 7.

Metrics

As one of our purposes of this study is to compare subjective and objective evaluation metrics, we need to have a way of mapping the subjective and objective measures used in our study. From the questionnaires six metrics where compiled: correct interpretation, correct response, repetition, control, coherence and habitability. Table 4 shows how these metrics were calculated for the dialogue logs.

Some mappings are rather straightforward, such as Correct interpretation, where Questionnaire item 2, Q2 *Kim understood what I said*, is mapped to the proportion of appropriate responses from the agent. However, the amount of nonsense, $n(G)$, i.e. random key strokes or empty utterances, is removed from the total, N, as such utterances never can be interpreted by the agent, nor a human. There is, thus, no correct interpretation for these and they are therefore excluded when calculating the proportion of correct interpretations.

Correct response is related to correct interpretation but more general since a correct response also includes when the agent responds with a request for repair or initiates a new topic when it fails to correctly interpret a user utterance. Thus, item Q7 *Kim mostly says the right thing at the right time* is mapped to the proportion of appropriate responses, $n(3)$, partially appropriate, $n(2)$, and request repairs, $n(RR)$.

For repetitions item Q16 *Kim often repeated herself* directly corresponds to the proportion of repetitions in the logs, REP. Since we want high values to correspond to pos-

Table 5: Subjective measures with mean (M), standard deviations (SD), and number of extreme values n(1) or n(7). M are also shown for the three groups: positive (Pos), slightly positive or neutral (Neut) and negative (Neg) attitude towards the conversational agent. t is calculated for the combined group Pos+Neut in contrast to Neg.

Questionnaire item	N	n(1)	n(7)	M	SD	M_{Pos}	M_{Neut}	M_{Neg}	t
Likeability (Q5, Q9)	19	2	5	4.34	2.23	7.0	5.2	1.6	<0.001
Naturalness (Q8, Q12)	19	3	1	4.08	2.04	5.9	4.9	2.0	<0.001
Correct interpretation (Q2)	19	5	2	3.37	2.01	4.4	4.4	1.6	<0.001
Correct Response (Q7)	19	4	4	4.05	2.32	5.8	5.6	2.2	<0.001
Repetition (Q16)	19	12	1	2.05	1.78	1.4	2	3.2	<0.1
Control (Q10)	19	3	5	4.42	2.17	6.2	6.0	2.2	<0.001
Coherence (Q13)	19	4	2	3.63	1.92	4.8	3.4	5.8	<0.05
Habitability (Q11, Q18)	19	3	2	4.03	2.18	5.3	5.6	1.7	<0.001

itive experiences of the conversations we deduct the number of repetitions, $n(REP)$, from the total number of utterances, N. This means that a conversation totally devoid of repetitions will have the value 1.

The user's sense of control, captured in item Q10 *I could control the interaction with Kim*, is not as straightforward to map to the dialogue coding. We use the proportion of initiatives the system takes, I-tags, since normally the user has control of the interaction and the system mainly takes the initiative when the user do not seem to want to control the interaction. Since a high proportion of system initiatives means a low value for control we turn the scale, by deducting the number of initiatives, $n(I)$, from the total number of utterances, N, in the same way as for repetitions.

The coherence of the dialogue, captured by questionnaire item Q13 *Sometimes I lost track of the conversation*, is mapped to the proportion of questions or statements that the system has misinterpreted and given faulty answers to, or utterances where the system responds that it has not understood. Such responses do not contribute to the flow of the conversation and is assumed to interrupt the users' track of conversation. Since this too is a negative value, the number of disruptive utterances $n(Q-1|S-1|RR)$, are deducted from the total number of utterances, N.

One important property of our system is habitability which is captured through the items Q11 *It was easy to understand how to talk so that Kim should understand* and Q18 *I always knew what I could say to Kim*. There is no obvious utterance type that directly correlates to habitability. We believe, however, that habitability can be correlated with the proportion of sequences of correct responses from the system to the users' questions, Q, statements, S, and greetings, closings and politeness, D, since this indicates that the user has been able to express such utterances in a way that the system can understand. Correct response does not necessarily mean that the system's interpretation is correct, a correct chat conversation also includes appropriate responses (tagged 2), see Figure 8. Such sequences depict conversations that flow naturally and as the user often has the initiative we believe that it is an indication of habitability. The reason for not dividing by the total number of utterances, N, is that N includes all Hazing/Flaming/Testing H and Non-interpretable G utterances, which varies between users, and these are not relevant since the user have not seriously tried to communicate with the agent in those turns of the dialogue.

Results

First we present the results from our two evaluations and then the correlations between the objective and subjective measures.

Subjective evaluation

Table 5 shows the results from the subjective evaluation, where items from the questionnaire has been reduced to a number of factors that capture various aspects of how the agent's conversational abilities and the dialogue with the agent is experienced. In Table 5 the scale has been adjusted so that high values always are positive for the system's performance. As can be seen the overall impression of the conversational agent is that it is neither very good nor bad as many measures have values around 4, for example likeability (M = 4.34) and naturalness (M = 4.08). The agent's conversational abilities are also neither good nor bad (correct interpretation M = 3.37, correct response M = 4.05), and it is neither hard nor easy to know how to interact with the agent (habitability M = 4.03).

However, there is a fairly large variation as indicated by standard deviations around 2 and in many cases high frequencies of both 1s and 7s. As observed during this and previous testings of the learning environment at the schools, there seem to be much bigger differences in the attitude towards the learning environment and the agent among the students in this age interval, than for younger students who tend to be more positive over all. Therefore we decided to further investigate subgroups of users. Looking in more detail at questionnaire item Q9, *I liked to talk to Kim* clearly revealed three groups of users, those with a negative attitude towards the agent (six persons of whom three responded with a 1 and three responded with a 2 in the questionnaire), those who like the agent (five persons who responded with a 7) and those who are slightly positive or neutral (seven persons where six have responded with a 5 and one person that responded with a 4). As seen in the right columns in Table 5, there are significant differences between the groups that like to chat (M_{Pos}, M_{Neut}) and those who do not like

Table 6: Mean of subjective measures over iterations between students that like and are neutral (M_{PN}), and dislike (M_{Neg}) the system. The difference between iterations is denoted Δ, e.g. ΔM_{Neg}=$M_{Neg}(2)$-$M_{Neg}(1)$ and t denotes the significance using the t-metrics. The t value is calculated using both positive and neutral students, just as for the calculations in Table 5.

Questionnaire item	N	$M_{PN}(2)$	$M_{Neg}(2)$	$M_{PN}(1)$	$M_{Neg}(1)$	ΔM_{PN}	t	ΔM_{Neg}	t
Likeability	19	5,83	1,79	5,22	1,86	0,61	<0.01	-0,07	-
Correct interpretation	19	4,33	1,71	3,11	1,86	1,22	<0.01	-0,14	-
Correct Response	19	5,25	2,00	4,54	3,00	0,71	0,06	-1,00	-
Repetition	19	6,42	5,14	5,67	5,71	0,75	-	-0,57	-
Control	19	5,75	2,14	5,11	3,86	0,64	<0.05	-1,71	-
Coherence	19	4,17	2,71	3,44	4,00	0,72	-	-1,29	-
Habitability	19	5,33	1,79	4,17	2,64	1,17	<0.01	-0,86	<0.01

to chat (M_{Neg}) for all factors, except repetition, concerning how they perceive the conversation with the agent.

We have also studied how students respond to the subjective metrics for earlier versions of the system to see if the students appreciate the new improved system, Table 6. Again we divide the students in groups that like respectively do not like the system and find that there is a significant increase for most metrics when the system's functionality is improved between iterations, but *only* for the group that like the system. Those students that do not like the system give the same low rating regardless of the system's actual functionality.

Objective measures

Tables 7 and Table 8 show the proportion of different types of user utterances and system responses in the logged conversations. As can be seen in Table 7 most user utterances are "appropriate" in that they are either Information requests (Q), Answers (A), General dialogue functions (D) or Statements (S), but a total of 22% are "inappropriate", i.e. Incomprehensible (G) or Abusive (H). As for the system's responses it seems that the system handles most utterances appropriately, see Table 8, although many of these are examples of RR, the agent very seldom (4%) responds inappropriate, 1.

Table 7: Proportion of different user utterances.

Code	Proportion (%)
D	14
Q	31
A	18
S	16
R	0
H	11
G	11

Table 9 shows the objective evaluation metrics. Since these are calculated as fractions, all values range from 0 to 1. While there are large variations between the max and min values for the objective measures, the objective measures differ from the subjective in that the standard deviations are much smaller. For some measures the mean falls in the middle, e.g. correct interpretation and habitability, but others are more on the extreme end of the scale, e.g. correct response. Looking at the subgroups based on whether they liked the chat or not, the significant differences are that there is more flaming/hazing and repetitions for the negative users (M_{Neg}).

Table 8: Proportion of different agent responses.

Code	Proportion (%)
3	51
2	15
RR	30
1	4

Comparison of subjective and objective measures

To compare the subjective and objective measures a correlation study was conducted where values for subjective and objective metrics for both the whole group as well as the subgroups were compared. No significant correlations between subjective and objective measures could be found, the correlation coefficients were approximately 0.2-0.3 for all aspects. Looking at the subgroups revealed only a single correlation between the subjective and objective measures for $Control$, which was 0.7, in the group that liked the agent.

The lack of correlations is not surprising given that although there are large individual differences in the subjective evaluation, especially between those that like the system and those that do not, (Table 5), there is no corresponding variance of the same magnitude in the actual dialogues (Table 9).

Discussion

Contrary to other investigations on subjective and objective measures, e.g. PARADISE (Walker et al. 1998) and the evaluation frameworks by Artstein et al. (2009), our study did not find any correlations between the subjective and objective evaluation metrics. We believe that the main reason for this can be attributed to the specific user group of young teenagers, but to a certain extent also to the design of the conversational agent and the design of the study itself.

Table 9: Objective measures with mean (M), minimum value (Min) and maximum value (Max), and standard deviations (SD). M are also shown for the three groups: positive (Pos), slightly positive or neutral (Neut) and negative (Neg) attitude towards the conversational agent. t is calculated for the combined group Pos+Neut in contrast to Neg.

Dialogue coding	N	$Min.$	$Max.$	M	SD	M_{Pos}	M_{Neut}	M_{Neg}	t
Correct interpretation	19	0.32	0.76	0.54	0.12	0.53	0.52	0.58	-
Correct Response	19	0.88	1	0.95	0.03	0.96	0.94	0.96	-
Repetition	19	0.84	1	0.91	0.04	0.88	0.90	0.94	$<$0.05
Control	19	0.62	0.88	0.70	0.06	0.69	0.69	0.74	-
Coherence	19	0.61	0.91	0.79	0.09	0.77	0.79	0.79	-
Habitability	19	0.16	0.75	0.49	0.15	0.43	0.56	0.45	-
Flaming/Hazing	19	0	0.55	0.11	0.14	0.08	0.06	0.19	$<$0.05

Our experience is that conducting studies with young teenagers in a school setting can be challenging as there are vast differences in how they approach the system and the study as such. Some students express enthusiasm and seriously engage with the system and also take their time to reflect over and answer questions in the questionnaire. Others have a very negative or uninterested attitude and do not put much effort in the interaction with the system nor answering the questionnaire.

Our analyses of the differentiated groups further support this as is shown in the analyses of the results from the subjective evaluations of previous versions of the system during the iterative development process, see Table 6, where we used the same items in the questionnaire as in this study. Students that have a positive attitude toward the system also appreciate the improved version whereas those that have a negative attitude do not, maybe because they do not take the survey seriously.

Since the conversational agent has a very robust approach for handling misunderstandings and flaming/hazing this leads to little variation in the objective measures. In the group of students that did not like the chat with the agent there were significantly more flaming and hazing (Table 6) but since the agent handles these and give appropriate responses the objective metric for correct responses remains very high. Similarly, uninterpretable utterances by users are not included in the analysis of correct interpretations and also contributes to high values for some users.

When calculating a Correct response, see Table 4, it may be a bit overoptimistic to weight the system responses Partially appropriate (2) and Request repair (RR) equally important as an Appropriate response (3). We have, however, experimented with various other weights for them, but that did not provide any significance either.

The number of subjects used in this study is admittedly small and the questionnaire was distributed after a rather long period (3 sessions). In a more recent study with more students a questionnaire was distributed after each session consisting of 30 minutes interactions. The results from this study are currently being analyzed.

To conclude, measures from our objective and subjective evaluation of a conversational agent for teenagers do not correlate. An implication of this is that data from subjective evaluations cannot be the only source of information to assess conversational agents, and neither can objective measures. However, objective measures are more homogenous and therefore probably better reflect a conversational system's capabilities. But as they do not correlate with the subjective measures they cannot be used to predict user satisfaction.

References

Artstein, R.; Gandhe, S.; Gerten, J.; Leuski, A.; and Traum, D. 2009. Semi-formal evaluation of conversational characters. *Languages: From Formal to Natural* 22–35.

Bickmore, T. 2003. *Relational Agents: Effecting Change through Human-Computer Relationships*. Ph.D. Dissertation, Media Arts & Sciences, Massachusetts Institute of Technology.

Biswas, G.; Katzlberger, T.; Brandford, J.; D., S.; and TAG-V. 2001. Extending intelligent learning environments with teachable agents to enhance learning. In Moore, J.; Redfield, C.; and Johnson, W., eds., *Artificial Intelligence in Education*. Amsterdam: IOS Press. 389–397.

Bunt, H.; Alexandersson, J.; Carletta, J.; Choe, J.-W.; Fang, A. C.; Hasida, K.; Lee, K.; Petukhova, V.; Popescu-Belis, A.; Romary, L.; Soria, C.; and Traum, D. R. 2010. Towards an ISO standard for dialogue act annotation. In Calzolari, N.; Choukri, K.; Maegaard, B.; Mariani, J.; Odijk, J.; Piperidis, S.; Rosner, M.; and Tapias, D., eds., *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.

Cooper, B., and Baynham, M. 2005. Rites of passage: embedding meaningful language, literacy and numeracy skills in skilled trades courses through significant and transforming relationships. Technical report, National Research and Development Centre for Adult Literacy and Numeracy.

Foster, M. E.; Giuliani, M.; and Knoll, A. 2009. Comparing objective and subjective measures of usability in a human-robot dialogue system. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, 879–887. The Association for Computer Linguistics.

Graesser, A.; Chipman, P.; Haynes, B.; and Olney, A.

2005. Autotutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions in Education* 48:612–618.

Gulz, A.; Haake, M.; and Silvervarg, A. 2011. Extending a teachable agent with a social conversation module – effects on student experiences and learning. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education. Auckland, New Zealand, 2011. Lecture Notes in Computer Science vol 6738*.

Hajdinjak, M., and Mihelič, F. 2006. The PARADISE evaluation framework: Issues and findings. *Computational Linguistics* 32(2):263–272.

Hamann, S. 2001. Cognitive and neural mechanisms of emotional memory. *Trends in Cognitive Sciences* 5(9):394–400.

Hone, K., and Graham, R. 2000. Towards a tool for the subjective assessment of speech system interfaces (SASSI). *Natural Language Engineering* 6(3/4):287–305.

Hung, V.; Elvir, M.; Gonzalez, A.; and DeMara, R. 2009. Towards a method for evaluating naturalness in conversational dialog systems. In *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA*, 1236–1241.

Kim, Y.; Wei, Q.; Xu, B.; Ko, Y.; and Ilieva, V. 2007. athgirls: Increasing girls' positive attitudes and self-efficacy through pedagogical agents. In *Proceedings of the 13th International Confernece on Artificial Intelligence in Education*.

Litman, D., and Forbes-Riley, K. 2006. Correlations between dialogue acts and learning in spoken tutoring dialogues. *Natural Language Engineering*.

Monge, P. R.; Bachman, S. G.; Dillard, J. P.; and Eisenberg, E. M. 1982. Communicator competence in the workplace: Model testing and scale development. In *Communication Yearbook, 5*. Beverly Hills, CA:Sage. 505–528.

Pareto, L. 2004. The squares family: A game and story based microworld for understanding arithmetic concepts designed to attract girls. In *In World Conference on Educational Multimedia, Hypermedia and Telecommunications*, volume 2004, 1567–1574.

Robinson, S.; Roque, A.; and Traum, D. R. 2010. Dialogues in context: An objective user-oriented evaluation approach for virtual human dialogue. In Calzolari, N.; Choukri, K.; Maegaard, B.; Mariani, J.; Odijk, J.; Piperidis, S.; Rosner, M.; and Tapias, D., eds., *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.

Shawar, B. A. A., and Atwell, E. S. 2007. Chatbots: are they really useful? *LDV-Forum* 22:31–50.

Silvervarg, A., and Jönsson, A. 2010. Towards a conversational pedagogical agent capable of affecting attitudes and self-efficacy. In *Proceedings of the Second Workshop on Natural Language Processing in Support of Learning: Metrics, Feedback and Connectivity, Bucharest, Romania*.

Silvervarg, A.; Gulz, A.; and Sjödén, B. 2010. Design for off-task interaction – rethinking pedagogy in technology enhanced learning. In *Proceedings ot the 10th IEEE Int. Conf. on Advanced Learning Technologies, Tunisia*.

Traum, D. R.; Swartout, W.; Gratch, J.; and Marsella, S. 2008. A virtual human dialogue model for non-team interaction. In Dybkjaer, L., and Minker, W., eds., *Recent Trends in Discourse and Dialogue*. NY: Springer. 45–67.

Walker, M. A.; Litman, D. J.; Kamm, C. A.; and Abella, A. 1998. Paradise: A framework for evaluating spoken dialogue agents. In Wahlster, M. M. . W., ed., *Readings in Intelligent User Interfaces*. Morgan Kaufmann.

Walker, M.; Boland, J.; and Kamm, C. 1999. The utility of elapsed time as a usability metric for spoken dialogue systems. In *Proceedings of ASRU*, 317–320. Citeseer.

Walker, M.; Kamm, C.; and Litman, D. 2000. Towards developing general models of usability with PARADISE. *Natural Language Engineering* 6(3):363–377.

Wallace, R. S. 2010. Artificial intelligence markup language. URL:http://www.alicebot.org/documentation/.

Speaking and Pointing – from Simulations to the Laboratory

Ingrid Zukerman[1], Arun Mani[1], Zhi Li[2] and Ray Jarvis[2]
[1]Faculty of Information Technology, [2]Intelligent Robotics Research Center
Monash University
Clayton, Victoria 3800, AUSTRALIA

Abstract

We describe our experiences in porting our probabilistic speech interpretation mechanism, in conjunction with a mechanism for interpreting pointing gestures, into a relatively unconstrained laboratory setting. Our results show that these conditions cause a drop in speech recognition and interpretation performance. While accurate pointing information is known to improve performance, gesture recognition in the lab was often inaccurate, yielding only small improvements under particular circumstances. This motivates our discussion of mechanisms for handling interpretation failures.

Introduction

DORIS (Dialogue Oriented Roaming Interactive System) is a spoken dialogue system designed for a household robot. *DORIS*'s spoken language interpretation module (called *Scusi?*) considers multiple sub-interpretations at different levels of the interpretation process, and estimates the probability of each sub-interpretation at each level (Zukerman et al. 2008). In (Kowadlo, Ye, and Zukerman 2010), we described an extension of *Scusi?* that incorporates information obtained from pointing gestures into interpretations of spoken utterances. This formalism was evaluated in a simulated setting where the pointing information was accurate, and the spoken language was delivered under optimal conditions (a person who trained the system spoke sanitized versions of real utterances into a microphone in a quiet room).

In this paper, we report our experiences in porting *Scusi?* from the simulated setting into a laboratory setting, where we employ a real gesture recognition system (Li and Jarvis 2010), and receive spoken and gestural input directly from a user. Figure 1 shows our trial subject pointing at an object on a table in our "sparse" setting (§ *Evaluation*). The main insight obtained from our experiments is that when objects are sparsely laid out, the combination of speech and gesture leads to better performance than each modality in isolation. However, overall interpretation performance still leaves something to be desired, owing to the inaccuracy of the speech and gesture recognition components. The adverse effect of this inaccuracy is exacerbated when objects are positioned close to each other or designated in a vague manner (e.g., by means of demonstrative pronouns). These insights motivate the need to address flawed interpretations due to ASR (and gesture recognition) errors.

Figure 1: Experimental setup: Trial subject in sparse setting

In the following section, we outline *Scusi?*'s interpretation process. Next, we describe our evaluation experiment, focusing on the interaction between speech and gesture. We then discuss related research, and consider approaches for handling poor ASR performance.

Interpreting Speech and Gestures

This section summarizes our previous work on the interpretation of single-sentence requests in conjunction with gestures (Makalic et al. 2008; Zukerman et al. 2008; Kowadlo, Ye, and Zukerman 2010). *Scusi?* processes spoken input in three stages: speech recognition, parsing and semantic interpretation. First, it runs an ASR (Microsoft Speech SDK 6.1 with dictation grammar) to generate candidate hypotheses (texts) from a speech signal, where each word in an output text is associated with a probability, and the texts are ranked in descending order of their overall probability (which is estimated by multiplying the individual word probabilities). In the second stage, Charniak's probabilistic parser (`ftp://ftp.cs.brown.edu/pub/nlparser/`) is applied to the texts in ranked order, associating each resultant parse tree with a probability.

During semantic interpretation, parse trees are succes-

Proceedings of the 7th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, pages 64–70,
Barcelona, July 2011

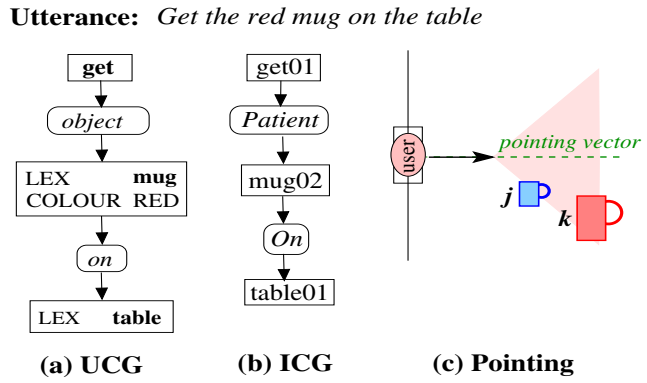

Figure 2: UCG, ICG and pointing for a sample utterance

sively mapped into two representations based on Concept Graphs (Sowa 1984). First *Uninstantiated Concept Graphs* (*UCGs*), and then *Instantiated Concept Graphs* (*ICGs*). UCGs, which represent syntactic information, are obtained from parse trees deterministically — one parse tree generates one UCG. Each UCG can generate many ICGs. This is done by nominating different instantiated concepts and relations from the system's knowledge base as potential realizations for each concept and relation in a UCG. Instantiated concepts are objects and actions in the domain (e.g., `mug01`, `mug02` and `cup01` are possible instantiations of the uninstantiated concept "mug"). Figures 2(a) and 2(b) respectively illustrate a UCG and an ICG for the request "get the red mug on the table". The *intrinsic* features of an object (lexical item and colour in this example) are stored in the UCG node for this object. *Structural* features, which involve two objects (e.g., "*mug* on the *table*"), are represented as sub-graphs of the UCG (and the ICG).

Information from a pointing gesture is incorporated into an interpretation by increasing the salience (probability) of the objects in the path of a *pointing vector*. Specifically, the probability that a user intended object k when pointing to a location in space and saying lexical item l is estimated using a conic spatial Gaussian density function and a temporal Gaussian density function. The spatial function is centered at the pointing vector, which extends from the speaker's face through to his/her hand into infinity, and increases in variance as the distance from the user's pointing hand increases (Figure 2(c)). This accounts for the increase in the area encompassed by a pointing gesture as a person points to objects that are farther from him/her. The probability of an object k is a function of its distance from the pointing vector and the variance of the Gaussian cone at the point where it intersects k. This probability is reduced in proportion to the area of the object that is occluded by objects between the user and k. For instance, both objects j and k in Figure 2(c) intersect the Gaussian cone around the pointing vector, with object j partially occluding k. The temporal Gaussian density function is centered at the time when pointing was performed. The closer the timing of lexical item l (which designates object k) is to the pointing time, the higher the probability of k.

In contrast to the simulated setting, our laboratory setting necessitates taking into account the confidence of the gesture interpretation mechanism in its pointing hypothesis. This is done by calculating a weighted average of the probability of an object due to pointing and its prior probability (in the absence of pointing), where the weighting is the confidence of the *Gesture Recognizer* (*GR*).

To respond in real time, the interpretation process continues until a preset number of sub-interpretations (including texts, parse trees, UCGs and ICGs) has been generated or all options have been exhausted. In addition, at each stage of the interpretation process, we employ an empirically determined *threshold* to discard interpretations whose probability is less than Pr(top-ranked interpretation) $\times$ *threshold*. For instance, the ASR threshold for textual outputs is currently set to 35%. As a result of these measures, many texts, parse trees and UCGs are not expanded.

Evaluation

Our evaluation aims to determine the relative contribution of our two input modalities (speech and gesture) to interpretation performance in close-to-realistic settings. We first present our experimental setup, followed by a description of our corpus, and the results of our experiments.

Evaluation Setup

Our evaluation was conducted in the laboratory which normally houses the *GR* (Figure 1). The setup consists of a space where several objects are placed (Figures 1 and 3).[1] Specific objects are then requested by a trial subject. The experiment was semi real-time, in the sense that there was no human intervention when processing the inputs, with the exception of the forwarding of the pointing vectors generated by the GR to *Scusi?*, and the selection of one of a pair of sentences (explained in § *ASR Performance*). The experimental conditions influenced the performance of the ASR and the GR as follows.

ASR Performance. The laboratory is relatively noisy, which adversely affected the performance of the ASR in preliminary trials. To minimize these effects, we recruited a Canadian speaker (a Canadian accent being the closest we could find to an American accent, for which the Microsoft ASR was developed). As a result, we had only one trial subject for our experiments.

Table 1 illustrates the differences between the simulated experiment reported in (Kowadlo, Ye, and Zukerman 2010) and the current experiment. In our previous experiment, utterances spoken by 19 people were re-spoken by one person who trained the ASR, after filtering the utterances who could not possibly be interpreted by *Scusi?*, and slightly sanitizing some of the remaining utterances in a systematic way. In contrast, in our current experiment, one person spoke to

[1] It is worth noting that in a deployed system, scene analysis software would identify the objects, and store them in *Scusi?*'s knowledge base. However, our GR only detects the presence of objects without identifying them. Hence, information pertaining to objects (type, position, dimensions and colour) is manually stored in *Scusi?*'s knowledge base.

Figure 3: Experimental setup: Cluttered setting

Table 1: Conditions and ASR performance in previous and current experiments

Conditions	Previous	Current
pointing vector	simulated	real
acoustics	quiet	noisy
# participants	19 (1 re-spoke the sentences)	1
filtered utterances	YES	NO
sanitized utterances	YES	NO
# of utterances	212	51
ASR performance		
% of top-ranked correct texts		
• without pointing	79.5%	47.6%
• with pointing	72.0%	36.7%
% of correct texts at any rank	90.6%	70.6%

DORIS after training the ASR for 10 minutes and receiving some instructions regarding the capabilities of the system. Specifically, the subject was advised not to refer to the composition and usage of objects, e.g., "rubber ball" or "water bottle" (which she sometimes did anyway), and she was asked to compose the sentences in her head prior to uttering them in order to avoid speech disfluencies, which are not handled by the system. Each sentence was spoken twice in order to improve the recognition chances of the ASR. For each pair of repeated sentences, we inspected the ASR output (all ranked texts), and used the best of the two outputs (the statistics in Table 1 are for these "best" outputs). No filtering or sanitizing of the participant's utterances was performed. The aim of this rather stringent setup was to provide a clear indication of the system's performance in realistic settings. Although the overall Word Error Rate (WER) was only 7.4% for the textual ASR output that was the closest to the spoken utterance (this was not the top-ranked text in 37% of the cases), the correct text was *not* returned (at any rank) for 29.4% of the requests. Further, in three of the remaining cases, the correct text had a very low probability, and it did not pass the ASR threshold (§ *Interpreting Speech and Gestures*). It is also worth noting that most of the erroneous textual outputs were single words in the noun position.

GR Performance. The GR is designed for realistic situations involving complex backgrounds, clothes of various colours and sleeve lengths, and different lighting conditions. However, the following subject-related aspects affected the performance of the GR: (1) arm position relative to the camera, (2) body stance, and (3) pointing steadiness.

- Arm position relative to the camera – when the subject pointed directly towards the camera, the detected pointing vector was less accurate than when the subject pointed sideways.
- Body stance – the subject sometimes tilted her head while pointing, which significantly affected the pointing vector generated by the GR (which is taken from the center of the user's face to the tip of the user's hand).
- Pointing steadiness – the steadiness of the pointing hand, which is normally reduced when the pointing finger is considered, affects the GR's confidence in the recognized gesture. Unfortunately, this confidence factor was not indicative of the accuracy of the pointing vector.

As a result, the pointing vectors returned by the GR were not as accurate as we had hoped. Specifically, they had an angular error of between $10 - 20^\circ$ to each side of the pointing arm. To cope with this level of inaccuracy, we significantly increased the increment made to the variance of the Gaussian cone for every meter of distance from the pointing hand (§ *Interpreting Speech and Gestures*). Specifically, in the simulated setting, the variance increment was 2.5 millimeters per meter of distance, while in the laboratory setting, the variance increment was 25 millimeters per meter. Such an increase was necessary in order to enable *Scusi?* to include at least one object within the cone. However, at the same time, this increment reduced the discriminating power of the pointing vector, especially when objects were relatively close to each other. For instance, in the cluttered setting the Gaussian cone typically encompassed about five objects.

A final limitation of this setup is that the ASR and the GR could not be temporally synchronized. We therefore had to disable *Scusi?*'s temporal component (§ *Interpreting Speech and Gestures*). However, from a practical point of view, this did not affect the results of our trial, as the participant pointed at most once for each utterance, and always at the intended object.

The corpus

As indicated above, we could recruit only one trial subject, who spoke directly to *DORIS*. Our subject was instructed to ask *DORIS* for several items (each item had a numerical label, and the subject was instructed to ask for the labeled objects in increasing numerical order).

To determine how the environment affects the contribution of the GR to interpretation performance, we considered two settings: *sparse* and *cluttered*. Under both settings, we generated three corpora: (1) *Speech only*, (2) *Speech + pointing*, and (3) *No-lex + pointing* (where the participant used demonstrative pronouns, rather than lexical items). The results for the third corpus are indicative of the performance

of pointing alone. As mentioned above, the trial subject was free to choose the wording to designate requested objects, modulo the instructions regarding *Scusi?*'s capabilities, which were sometimes disregarded.

- **Sparse setting –** In this setting, there were only six objects, all of which were placed on a table (Figure 1). The six objects were chosen so as to include distractors as follows: two similar plates, two white cups (one slightly larger), and two bowls (one red and one green). Our trial subject requested each of the six objects in the three ways described above, which yielded a total of 18 requests.
- **Cluttered setting –** In this setting, the space contained 18 objects, 11 of which were on a table, and the rest on chairs, boxes or on the floor. This setting contained additional distractors, e.g., an extra plate, two drink bottles, and two folders on a chair. Figure 3 shows a partial view of this setting (the red car and flower-patterned box under the table and the computer monitor and keyboard are not part of the experimental setup). The participant requested 15 objects for the *Speech only* condition, and all the 18 objects for the *Speech + pointing* condition. The *No-lex + pointing* condition was not considered for the cluttered setting owing to the poor interpretation performance for this condition in the sparse setting (§ *Results*).

Scusi? was set to generate at most 300 sub-interpretations in total (including texts, parse trees, UCGs and ICGs) for each spoken request. On average, *Scusi?* took about 28 seconds to interpret a spoken utterance. An interpretation was deemed successful if it correctly represented the speaker's intention, which was encoded in one or more *Gold ICG*s. These ICGs were manually constructed on the basis of the requested objects and the participants' utterances. Multiple Gold ICGs were allowed if there were several suitable actions in the knowledge base.

Results

Table 2 summarizes the results of our experiments, divided into our two environmental settings: sparse and cluttered. Column 1 displays the test condition: *Speech only*, *Speech + pointing*, and *No-lex + pointing*. For the *Speech + pointing* condition, we present the results obtained with two methods: (1) *pointing vector*, and (2) *no vector*. The second method considers the requests uttered by the user while pointing, e.g., "I want *that bowl over there*", but does not take into account the pointing vector. This method provides a baseline for determining the impact of pointing information.

Column 2 shows the fraction of ASR and GR failures for each condition (ASR failure appears *underlined* in the row corresponding to the speech test condition, and GR failure appears in the row corresponding to the pointing method). We considered the ASR to have failed if it did not return the correct text at a rank high enough to be considered by *Scusi?* (§ *Interpreting Speech and Gestures*). Although WER was good by current standards, ASR performance was quite mediocre in terms of understanding complete requests, failing approximately 50% of the time in three of the conditions (*Speech + pointing* in both the sparse and cluttered settings, and *Speech only* in the sparse setting). The GR was deemed to have failed if the returned vector did not point anywhere near the intended object (e.g., the floor instead of the red bottle). GR failure rates were extremely high for the *No-lex + pointing* condition, where it was needed the most. The GR exhibited consistent performance in the other settings (in terms of the angle from the intended object). However, owing to the increase in the variance of the Gaussian cone around the pointing vector (to cope with the inaccuracy of the vector returned by the GR), the cone included more objects in the cluttered setting than in the sparse setting, thus reducing its discriminating power.

Table 2: *Scusi?*'s interpretation performance

	# Fail <u>***ASR***</u> **GR**	**Avg. adj. rank**	**# Top rank**	**# Bad adj. rank**	**# Not found**
Sparse setting (average 4.67 words/sentence)					
Speech only	<u>*3/6*</u>	**1.17**	**4/6**	**2/6**	0
Speech + pointing	<u>*3/6*</u>				
pointing vector	1/6	**1.17**	**3/6**	**1/6**	0
no vector	–	1.58	2/6	3/6	0
No-lex + pointing	<u>*1/6*</u>				
pointing vector	5/6	4.13	1/6	3/6	2/6
Cluttered setting (average 6.06 words/sentence)					
Speech only	<u>*4/15*</u>	**1.25**	**8/15**	**2/15**	3/15
Speech + pointing	<u>*8/18*</u>				
pointing vector	3/18	1.80	7/18	2/18	8/18
no vector	–	**1.32**	**8/18**	**2/18**	7/18

The remaining columns display four measures of performance: *average adjusted rank* of the Gold ICG, *# Top rank*, *# Bad adjusted rank*, and *# Not found*. The *rank* of an ICG I is its position in a list sorted in descending order of probability (starting from position 0), but all equiprobable ICGs are deemed to have the same rank. The *adjusted rank* (*AR*) of an ICG I is the mean of the positions of all ICGs that have the same probability as I, e.g., if we have 4 equiprobable ICGs in positions 0-3, each has a rank of 0, but an adjusted rank of $\frac{r_{\text{best}}+r_{\text{worst}}}{2} = 1.5$. We use AR in addition to rank, as rank alone does not indicate whether *Scusi?*'s results are meaningful – in principle all options could be assigned the same probability, and hence have a rank of 0. In order to assess the quality of *Scusi?*'s results, we need to know how many equiprobable Gold ICGs there are, and where they are positioned in the overall ranking. The AR combines these quantities. The average AR – the mean of the AR of the Gold ICG for all the utterances – appears in Column 3. Column 4 shows *# Top rank* – the fraction of the utterances that yielded a Gold ICG with rank 0 (either as a singleton or as one of a pair). Column 5 shows *# Bad adjusted rank* – the fraction of the utterances that yielded a Gold ICG with a high (bad) AR. In the sparse setting, where there are 6 objects on the table, an AR ≥ 2 is considered bad, while in the cluttered setting, an AR ≥ 6 is considered bad. For instance, in the *Speech only* condition in the sparse setting, the Gold ICG for 2/6 requests has a bad AR. The last column displays *# Not found* – the fraction of the utterances that didn't yield a Gold ICG.

Sparse setting. The best performance was obtained when additional evidence was provided. Such evidence, in the form of pointing or a colour specification, enabled *Scusi?* to overcome the failure of individual components. For example, when the participant said "pass me the red bowl" (*Speech only*), the ASR returned "bull" for "bowl" (ASR failure), but the colour "red" enabled the identification of the correct bowl. Likewise, when the participant said "I want that red one" (*Speech + pointing*), the pointing action coupled with the colour enabled *Scusi?* to overcome a parsing error. However, certain ASR failures (e.g., "can you" heard as "tenure") could not be overcome.

When pointing was not accompanied by additional evidence (*No-lex + pointing*), the results were quite discouraging. In most cases, the inaccuracy of the pointing vector prevented *Scusi?* from generating useful interpretations, with all the objects being likely candidates. The GR yielded a single Gold ICG with a good rank (in fact, the top rank) only for one utterance. Note that the average AR for the Gold ICG in the *No-lex + pointing* condition is 4.13, which is quite high when there are only 6 objects on the table. Due to the inferior performance exhibited under the *No-lex + pointing* condition, this condition was not trialled for the cluttered setting.

Cluttered setting. As for the sparse setting, the provision of additional information enabled *Scusi?* to overcome some ASR failures. However, in this setting, pointing information had an overall detrimental effect on interpretation performance: when the pointing vector was inaccurate, it led the interpretation astray, and when the vector was fairly accurate, its information usually had no tangible effect owing to the vector's reduced discriminating power in this setting.

Most of the errors were caused by ASR failure, which occurred more frequently under the *Speech + pointing* condition, e.g., "bowl" heard as "bull", "saucer" as "sauce are", and "thread" as "red". As seen in Table 1, a drop in ASR performance from the *Speech only* condition to the *Speech + pointing* condition was also observed in (Kowadlo, Ye, and Zukerman 2010), but it was less pronounced than the drop observed here. This difference may be attributed to our more stringent experimental conditions.[2] Litman *et al.* (2000) argue that for a given speaker, longer sentences cause more failures in ASR performance than shorter sentences. Although our participant uttered longer sentences in the cluttered setting than in the sparse setting (6.06 words on average versus 4.67 respectively, Table 2), this did not translate to higher ASR failures. On the contrary, the longest sentences were uttered for the *Speech only* condition in the cluttered setting (6.4 words on average), which had one of the lowest ASR failure rates. We posit that most ASR failures may be attributed to particular terms the ASR found challenging (such as the above), which were not clearly enunciated under the pointing condition.

Overall performance. Although our evaluation was conducted with only one participant, it identifies limitations of current speech and gesture recognition technologies in realistic settings.[3] Firstly, our results confirm the findings of Stiefelhagen *et al.* (2004), whereby WER translates to a significantly higher *Sentence Error Rate* (*SER*). Further, we found that in most cases, the error is confined to one word, which is usually in the noun position (this remains to be verified for more users). These results highlight the need for mechanisms that handle ASR errors (López-Cózar and Callejas 2008; Stiefelhagen et al. 2004; Sugiura et al. 2009) and speech disfluencies (Germesin, Becker, and Poller 2008; Stiefelhagen et al. 2004), and for a procedure that generates clarifications questions for terms that cannot be otherwise elucidated.

In addition to ASR failure, six interpretation failures were due to *Scusi?*'s current inability to process references to sets of objects (e.g., "set of cups") and certain positional phrases, e.g., "way over there", "in the center" ("in" denotes containment for *Scusi?*), and "on top" or "closest" without a target referent. The most common of these cases (determined from other corpora) will be incorporated into *Scusi?* in the near future. However, more importantly, we intend to develop an approach to enable *Scusi?* to generate appropriate responses to such comprehension failures.

Related Research

Most of the research in gesture and speech integration employs speech as the main input modality, with gesture providing additional information. Different approaches are used for gesture detection, e.g., vision (Stiefelhagen et al. 2004; Brooks and Breazeal 2006) and sensor glove (Corradini, Wesson, and Cohen 2002); and for language interpretation, e.g., dedicated grammars (Stiefelhagen et al. 2004; Brooks and Breazeal 2006) and keywords (Einstein and Christoudias 2004). Fusion is variously implemented using heuristics based on temporal overlap (Bolt 1980; Johnston et al. 2002), querying a gesture-sensing module when ambiguous referents are identified (Fransen et al. 2007), or unification to determine which elements can be merged (Corradini, Wesson, and Cohen 2002; Stiefelhagen et al. 2004). These are sometimes combined with search techniques coupled with penalties (Einstein and Christoudias 2004; Brooks and Breazeal 2006). With the exception of Bolt's system, these systems were tested on utterances that were quite short and constrained.

Scusi? follows the above trend, with pointing information influencing the prior probability (salience) of the objects in the space. In addition, our use of a probabilistic parser (instead of one based on a hand-crafted grammar) enables us to handle more complex utterances than those considered by most Spoken Dialogue Systems (SDSs) (Jokinen and McTear 2010), and our consideration of multiple interpretations in conjunction with several sources of evidence allows us to recover from some errors due to the failure of individual input modalities. Nonetheless, our results show that in order to enable an SDS to function in realistic, reason-

[2] This drop was not observed in the sparse setting, but there were not enough utterances to draw conclusions.

[3] The "Let's go" challenge (Black and Eskenazi 2009) in the bus timetable domain is certainly realistic. However, the parameters of the interaction are more restricted than ours.

ably open-ended settings, significant improvements in component technologies are required, as well as the ability to identify and handle potential misunderstandings.

Discussion and Conclusion

We have described our experiences in porting our probabilistic speech interpretation mechanism into a realistic laboratory setting, where it receives input from a GR and an ASR in semi real-time. Our results show that pointing information which can discriminate between some candidate objects slightly improves interpretation performance. In our trials this happened in the sparse setting. Performance also improves when the user provides extra specifications about the desired object, e.g., lexical item, colour or position. That is, redundant information helps overcome some failures of individual input-sensing systems, viz ASR and GR.

Unfortunately, the accuracy of the GR is insufficient to handle less constrained conditions (i.e., the cluttered setting), leading to an overall detrimental effect on interpretation performance. The performance of the ASR is variable, ranging from 16% SER to 50%. Our limited trial, together with the insights from (Kowadlo, Ye, and Zukerman 2010), also indicates that ASR performance is sensitive to the conditions, in particular to whether the participant is concentrating on speaking clearly. This indicates that the contributing technologies (at least those we used) are not mature enough for deployment in realistic settings.

The poor performance of the ASR in terms of SER prompts us to consider two complementary approaches to improve *Scusi?*'s performance in light of ASR errors: *prevention* and *recovery*.

Prevention. This approach constrains the vocabulary and grammatical constructs understood by an ASR (Brooks and Breazeal 2006; Gorniak and Roy 2005; Matsui et al. 1999; Sugiura et al. 2009). The preventive approach enables ASRs to process expected utterances efficiently, and hence works well in restricted domains. However, it leads to situations were the system hears "what it wants to hear", and hence has difficulty processing unexpected utterances. We propose to employ some prevention, in the sense that we intend to use a vocabulary of a few thousand words (instead of an unrestricted vocabulary). However, we eschew severe vocabulary restrictions, such as those imposed by Sugiura *et al.* (2009) (23 words), and grammatical restrictions, such as those imposed by Brooks and Breazeal (2006).

Recovery. We consider two types of recovery: implicit and explicit. Implicit recovery employs different information sources (e.g., contextual, syntactic and lexical) to modify the output of a process, e.g., speech recognition, parsing or semantic interpretation. Researchers have investigated word-level and sentence level approaches to modify the output of an ASR. Word-based approaches involve replacing, inserting or deleting words in a textual ASR output (López-Cózar and Callejas 2008; Stiefelhagen et al. 2004; Sugiura et al. 2009), or modifying tenses of verbs and grammatical numbers to better match the grammatical expectations in the domain at hand (López-Cózar and Callejas 2008). Sentence-based approaches involve re-ranking the textual hypotheses produced by the ASR (Lemon and Konstas 2009), and identifying misunderstood sentences (Litman, Hirschberg, and Swerts 2000; Litman and Pan 2000).

López-Cózar and Callejas (2008) and Stiefelhagen *et al.* (2004) rely on semantic grammars and context-free grammars respectively to identify suspect words, while Sugiura *et al.* (2009) adopt a probabilistic approach, albeit in a very restricted setting. We propose to combine phonetic matching with contextual information to postulate words that have a higher prior probability in the current context than some of the words returned by the ASR. For instance, such a match would assign a higher probability to "bowl" and "ball" than to "plate" when the heard word is "bull". However, the problem of identifying promising word candidates for replacement or modification in real time is a challenge for open-ended settings.

At present, we rank the ASR outputs using the product of the probabilities of the individual words in a text (§ *Interpreting Speech and Gestures*). However, this disadvantages ASR textual outputs that have more words or that include a few low-probability words. To overcome these problems we propose to re-rank the ASR outputs according to their mean probability.

Explicit recovery involves asking clarification questions with respect to particular objects, attributes or actions in an interpretation, e.g., "Did you want the cup or the cap?" or "What did you want me to do with the mug?" (Oulasvirta et al. 2007). We are currently developing an explicit recovery mechanism that hinges on the identification of certain types of events which warrant clarification. For instance, when one or more words returned by the ASR yield a poor match with all the objects (or attributes or actions) known to the system, *Scusi?* generates multiple low-probability ICGs that differ only in one or two nodes. As another example, it is often the case that some words mis-heard by the ASR are plausible, but the entire request is problematic, e.g., the ASR hears "get me the green ball" in a room that contains a red ball and a green bowl.

Although these recovery measures will contribute towards improving ASR and SDS performance, it is unrealistic to expect *Scusi?* (or any open-ended dialogue system) to understand all possible user inputs. We therefore must develop a general formalism to enable *Scusi?* to diagnose the state of the understanding process (i.e., to what extent is the user being understood), and to handle unexpected events (e.g., out-of-grammar, out-of-vocabulary or out-of-capability utterances). To this end, we propose to identify problematic segments of an interpretation, and investigate the effect of ignoring different types of such segments on *Scusi?*'s overall understanding of an utterance and its ability to respond productively.

Acknowledgments

This research was supported in part by grants DP0878195 and DP110100500 from the Australian Research Council.

References

Black, A., and Eskenazi, M. 2009. The spoken dialogue challenge. In *Proceedings of the 10th SIGdial Conference on Discourse and Dialogue*, 337–339.

Bolt, R. 1980. "Put-that-there": Voice and gesture at the graphics interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*, 262–270.

Brooks, A., and Breazeal, C. 2006. Working with robots and objects: Revisiting deictic reference for achieving spatial common ground. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*, 297–304.

Corradini, A.; Wesson, R.; and Cohen, P. 2002. A Map-Based system using speech and 3D gestures for pervasive computing. In *ICMI'02 – Proceedings of the 4th International Conference on Multimodal Interfaces*, 191–196.

Einstein, J., and Christoudias, C. 2004. A salience-based approach to gesture-speech alignment. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 25–32.

Fransen, B.; Morariu, V.; Martinson, E.; Blisard, S.; Marge, M.; Thomas, S.; Schultz, A.; and Perzanowski, D. 2007. Using vision, acoustics, and natural language for disambiguation. In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, 73–80.

Germesin, S.; Becker, T.; and Poller, P. 2008. Domain-specific classification methods for disfluency detection. In *Proceedings of Interspeech 2008*, 2518–2521.

Gorniak, P., and Roy, D. 2005. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *ICMI'05: Proceedings of the 7th International Conference on Multimodal Interfaces*, 138–143.

Johnston, M.; Bangalore, S.; Vasireddy, G.; Stent, A.; Ehlen, P.; Walker, M.; Whittaker, S.; and Maloor, P. 2002. MATCH: an architecture for multimodal dialogue systems. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 376–383.

Jokinen, K., and McTear, M. 2010. *Spoken Dialogue Systems*. Morgan and Claypool.

Kowadlo, G.; Ye, P.; and Zukerman, I. 2010. Influence of gestural salience on the interpretation of spoken requests. In *Proceedings of Interspeech 2010*, 2034–2037.

Lemon, O., and Konstas, I. 2009. User simulations for context-sensitive speech recognition in spoken dialogue systems. In *EACL 2009 – Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, 505–513.

Li, Z., and Jarvis, R. 2010. Visual interpretation of natural pointing gestures in 3D space for human-robot interaction. In *ICARCV 2010 – Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision*, 2513–2518.

Litman, D., and Pan, S. 2000. Predicting and adapting to poor speech recognition in a spoken dialogue system. In *AAAI-00 – Proceedings of the 17th National Conference on Artificial Intelligence*, 645–651.

Litman, D.; Hirschberg, J.; and Swerts, M. 2000. Predicting automatic speech recognition performance using prosodic cues. In *NAACL-HLT 2000 Proceedings – Human Language Technologies: The 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, 218–225.

López-Cózar, R., and Callejas, Z. 2008. ASR post-correction for spoken dialogue systems based on semantic, syntactic, lexical and contextual information. *Journal of Speech Communication* 50(8-9):745–766.

Makalic, E.; Zukerman, I.; Niemann, M.; and Schmidt, D. 2008. A probabilistic model for understanding composite spoken descriptions. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, 750–759.

Matsui, T.; Asoh, H.; Fry, J.; Motomura, Y.; Asano, F.; Kurita, T.; Hara, I.; and Otsu, N. 1999. Integrated natural spoken dialogue system of Jijo-2 mobile robot for office services. In *AAAI99 – Proceedings of the 16th National Conference on Artificial Intelligence*, 621–627.

Oulasvirta, A.; Engelbrecht, K.; Jameson, A.; and Möller, S. 2007. Communication failures in the speech-based control of smart home systems. In *Proceedings of the 3rd International Conference on Intelligent Environments*, 135–143.

Sowa, J. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA: Addison-Wesley.

Stiefelhagen, R.; Fugen, C.; Gieselmann, R.; Holzapfel, H.; Nickel, K.; and Waibel, A. 2004. Natural human-robot interaction using speech, head pose and gestures. In *IROS 2004 – Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, 2422–2427.

Sugiura, K.; Iwahashi, N.; Kashioka, H.; and Nakamura, S. 2009. Bayesian learning of confidence measure function for generation of utterances and motions in object manipulation dialogue task. In *Proceedings of Interspeech 2009*, 2483–2486.

Zukerman, I.; Makalic, E.; Niemann, M.; and George, S. 2008. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, 581–592.