Prosodic marking of contrasts in information structure

Markus Guhe, Mark Steedman, Ellen Gurman Bard
Human Communication Research Centre
University of Edinburgh
{m.guhe; m.steedman; e.bard} @ed.ac.uk

Max M. Louwerse
Department of Psychology / Institute for Intelligent Systems
University of Memphis
mlouwers@memphis.edu

1 Prosodic marking of contrast

Successful dialogue requires cultivation of common ground (Clark, 1996), shared information, which changes as the conversation proceeds. Dialogue partners can maintain common ground by using different modalities like eye gaze, facial expressions, gesture, content information or intonation. Here, we focus on intonation and investigate how contrast in information structure is prosodically marked in spontaneous speech.

Combinatory Categorial Grammar (CCG, Steedman 2000) distinguishes *theme* and *rheme* as elements of information structure. In some cases they can be distinguished by the pitch accent with which the corresponding words are realised. We experimentally evoke instances of contrasting themes and rhemes to establish the circumstances under which the pitch accents occur in unrestricted spoken dialogue. ‘Contrast’ means ‘alternatives are available’, not ‘contrastive accent’. It is difficult to manipulate context or outcome in quasi-natural engaging situations. Even if contrasting themes and rhemes are available, speakers choose from among a wider set of contrastable elements when framing utterances. Their choice may be difficult to predict: contrasts not apparently critical to the local context may be as important to speakers as ones usually thought to define the situation under discussion.

Unscripted dialogue with pressing communicative motivation is difficult to control for genre, topic, and goals. We use a modified map task (Anderson et al. 1991), a restricted-domain route-communication task, which establishes what each participant knows at any time. Without sight of each other’s maps, an Instruction Giver (IG) and Follower (IF) collaborate to reproduce on IF’s map a route printed on IG’s. The route can be adequately described by route-critical landmarks. As Fig. 1 illustrates, map pairs differ in the features of landmarks and in ‘ink damage’ that obscures the colours of some landmarks on IF’s map. Participants know that maps can differ but must learn where and how.

The discrepancies between maps do not fully define the alternatives sets speakers may wish to contrast. Instead, speakers define that alternatives set by their intonation. Provided that it is consistent with the context, the hearer will accommodate that set. Take:

(1) IF: *Do you see the two brown trees and the and the four black trees?*
IG: *You mean* THREE *black trees right?* (1:1–2:T:700.7; 1–1)

By deaccenting ‘black’ and ‘trees’ IG presupposes that the alternatives are confined to sets of black trees; specifically to IG’s set of three and IF’s set of four. Both can then adjust common ground incrementally.

As there is intense debate about whether the involved pitch accents (L+H* and H*) are actually categorically distinct (Ladd & Schepman 2003, Calhoun 2004), we simply seek to establish that contrasts in the information structure are indeed marked overtly by some form of prominence. We therefore use an undifferentiated notion of perceptual prominence to determine whether contrasts are marked by phonetic means.

Our prediction is the following: Only words whose denotation contributes to distinguishing the entity referred to from the other entities in the alternatives set are marked by prominence.

2 Experiment

Key-objects (here: trees) provide the route-critical landmarks for a map. They differ among a single map’s landmarks by colour and by one other feature (here: number). We report findings for two dialogues for the maps in Fig. 1 in order to identify episodes containing the predicted contrasts. (We superficially looked at others, which corroborated our findings.) The results are consistent within and between participant dyads. Landmarks differ in colour of tree groups; group size (1 to 5), presence of the group on IG’s /IF’s map, whether ink obscures the colour on IF’s.

brandial 2006: Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue, pages 179–180.

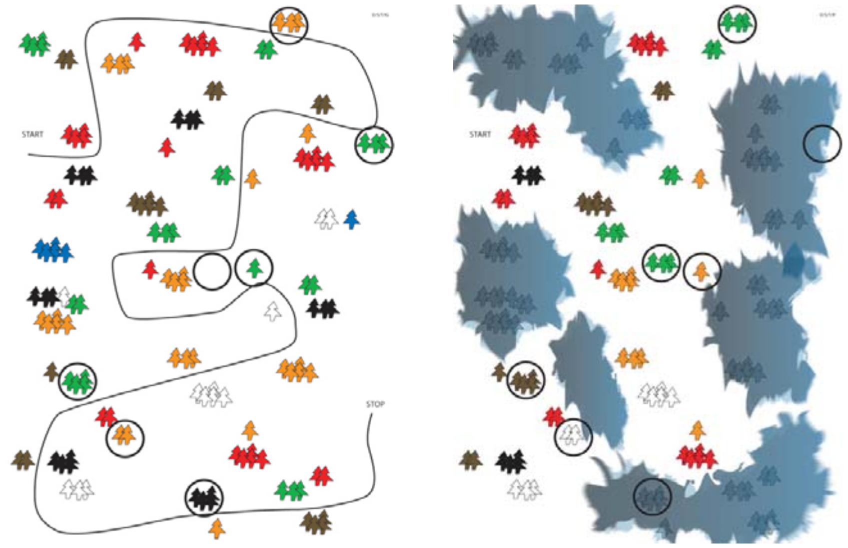

Figure 1: Maps for the analysed dialogues; IG's map (left) contains a route and a START and STOP mark; IF's map contains 'ink blots' that obscure the colour of some objects; circles (added here for expository purposes) indicate the differences between the maps

We assessed perceptually whether the mentioned items are prominent. For landmarks differing between maps (except those inked out) we also established the most prominent item of the intonation phrase – the contrasted element.

The material contains 146 intonational phrases that mention one or two landmarks in the form [number] [colour] ['tree'/'one'] and where at least one of [number] or [colour] is present. There are 334 mentions of features (e.g. 'red', 'two') in these phrases. In only 6 mentions is the feature term *non*-prominent, but not all prominences are realised by pitch movement. Seven differences between the maps are unrelated to ink-blots: 4 colour differences, 1 number difference, 1 landmark present only on one map, respectively. They are the prime place for eliciting contrasting intonation that correct the dialogue partner's knowledge representation, cf (1). Of the 146 phrases, 9 refer to differences between maps.

The phrases include 210 mentions of landmarks, of which 124 mention both features. There is no clear preference for assigning prominence to features (86 use equal prominence; 21 make the number term more prominent, 17 the colour term). Number mentions predominate in single-feature mentions (65 number vs 21 colour). This appears to be a response to the fact that number is the more reliable feature. 137 phrases describe landmarks on a single map, of which 131 instances mention landmarks within the 'magic circle', an imaginary circle around the current position that contains the landmarks identifying the next leg. Of the other 6, 4 are close to the circle and 2 are only in the discourse history.

The two dialogues mention 9 of the 14 possible differences between maps; in 8 cases a pitch accent marks the contrast. In 2 instances the participants are off-route. So, the speakers could have chosen to mention 12 differences between the maps. The ratio of 9(8)/12 is very satisfactory.

3 Discussion

In this exploratory evaluation we looked at places in the maps that are prone to prompt intonation patterns marking a contrast in the information structure. Differences *within* one map do not seem to elicit prosodic structures that mark contrasts between landmarks. These mentions are only informing or describing. Differences *between* maps require to correct the dialogue partner's knowledge representation and to introduce new information into the common ground. These contrasting items receive the most prominent pitch accent. With the exception of Ito et al (2004) we are not aware of experimental settings that can elicit 9 of 12 possible contrasts in unrestricted dialogue. In contrast to reading sentence lists this will provide deeper insight into actual dialogue.

Acknowledgements

This research was supported by the National Science Foundation (IIS-0416128).

References

Anderson, A., et al (1991). The HCRC Map Task Corpus. *Language and Speech*, 34: 351–366.

Calhoun, S. (2004). Phonetic dimensions of intonational categories: The case of L+H* and H*. In: *Proceedings of Prosody 2004*, Nara, Japan.

Clark, H. H. (1996). *Using Language*. Cambridge University Press, Cambridge.

Ito, K., Speer, S. R., & Beckman, M. E. (2004). Informational status and pitch accent distribution in spontaneous dialogues in English. In: *Proceedings of Speech Prosody 2004*, 279–282.

Ladd, D. R. & Schepman, A. (2003). 'Sagging Transitions' between high pitch accents in English. *Journal of Phonetics*, 31(1): 81–112.

Steedman, M. (2000). *The syntactic process*. MIT Press, Cambridge, MA.

Evaluation of an Information State-Based Dialogue Manager

Antonio Roque, Hua Ai[†], and David Traum

USC Institute for Creative Technologies
13274 Fiji Way
Marina Del Rey, CA 90292
{roque,traum}@ict.usc.edu

[†]Intelligent Systems Program
University of Pittsburgh
210 S. Bouquet, Pittsburgh PA 15260
hua@cs.pitt.edu

Abstract

We describe an evaluation of an information state-based dialogue manager by measuring its accuracy in information state component updating.

1 Introduction

Evaluation of dialogue managers is essential for the development of dialogue systems. However, it can be difficult to separate the performance of a dialogue manager from the performance of the system as a whole. Here we describe an approach towards evaluating the performance of an Information State-based dialogue manager separately from the other components of the dialogue system and the system as a whole.

Our testbed system, Radiobot-CFF (Roque et al., 2006), is a military virtual reality environment designed to train soldiers in artillery strike requests. The trainees hold a radio dialogue with Radiobot-CFF during which an enemy target is located and attacked. Radiobot-CFF includes a speech recognition component, a dialogue move interpreter, and an information state-based dialogue manager (Roque and Traum, 2006). We ran an evaluation of the system from which we calculated task completion rates and time-to-task measures for the system as a whole, as well as error rates for the speech recognition and interpreter components (Robinson et al., 2006). However, we lacked an analysis of the dialogue manager component's performance.

2 Evaluation

Radiobot-CFF uses an information state-based (Traum and Larsson, 2003) dialogue manager, and therefore works by firing update rules which are dependent on and which change information state components. For example, Radiobot-CFF uses information state components to track whether it has received a target's location and what that target location is, as well as whether it has enough information to send a fire. To evaluate the performance of our dialogue manager, we studied how well it updated its information state components.

2.1 Approach

Our approach is to use human coders to decide how the information state components should be updated, given a sequence of utterances, and to compare that to how the system actually does update its information state components.

We develop a coding manual of guidelines for updating the information state components based on the kind of input received. We then use a sequence of trainee utterances (produced by hand-transcribing audio logs and hand-correcting system dialogue move interpretations of those utterances) to produce a sequence of hand-coded information state components. That sequence is our gold standard, and represents the output of the dialogue manager if the speech recognition, interpreter, and dialogue manager components are all performing to the level of a human.

We compare our system's performance to this gold standard corpus in two conditions. First, we run the dialogue manager on perfect input by feeding it the hand-corrected interpreter output, recording the information state components after every utterance, and comparing that to our gold standard. This allows us to evaluate the dialogue manager separately from the rest of the system, so that errors in the speech recognition and interpreter components do not affect its performance. Secondly, we compare the gold standard to the system's information state components when updated by the system on actual speech recognition and interpreter input. This allows us to evaluate the dialogue manager's performance given noisy input.

brandial 2006: Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue, pages 181–182.

IS Component	*Accuracy, corrected input*	*Accuracy, noisy input*
has warning order	0.76	0.67
has target location	0.98	0.90
has grid location ‡	0.99	0.96
has polar direction	0.83	0.80
has polar distance	0.99	0.91
has target descript.	0.93	0.76
has enough to fire	0.99	0.52
method of control	0.71	0.71
method of fire †	0.38	0.44
grid value ‡	0.98	0.96
direction value	0.83	0.79
distance value	0.99	0.91
adjust fire	0.88	0.65
repeat FFE *	0.89	0.97
LR adjustment	0.99	0.92
AD adjustment	1.00	0.97
end of mission	0.93	0.91
disposition	0.93	0.78
number of casualties	0.95	0.83
mission is polar	0.99	0.85
last method of fire †	0.90	0.61
missions active	0.81	0.67

† Kappa was less than 0.8 and greater than 0.67
‡ Kappa was less than 0.67
* Kappa could not be calculated, as its value never changed in the data over which kappa was measured.

Table 1: Accuracy per IS Component

2.2 Results

We worked with a corpus of 17 sessions consisting of 407 utterances, representing a total of 8954 information state components to be updated. A pair of human coders coded several sessions by consensus to develop a set of guidelines, then individually coded the rest of the corpus. Several sessions were held out for concurrent coding by both coders, from which a kappa score was calculated per information state component. Components had kappa values above 0.8 except as noted in Table 1.

We then fed the corrected utterance interpretations into the dialogue manager to get sequences of IS component updates for corrected interpretations, and processed log files from the full system evaluation to get sequences of IS component updates for noisy interpretations. Accuracy results (measured by number of times the dialogue manager agreed with the human coder) for both are shown in Table 1.

3 Future Work

Because the input used in the corrected input condition is not reacting to the dialogue manager's responses, the dialogue may take an unnatural direction; for example, in which the dialogue manager is repeatedly prompting or correcting the trainee, but the trainee is proceeding as if there is no problem.

Also, a component's value may be more important at certain parts of a dialogue than at others. For example, as shown in Table 1, the "method of fire" component's accuracy is low, but the dialogue manager and humans disagree on its value most often at a phase of the dialogue in which the "method of fire" value is never used in decisions or output.

We hope to quantify and address these problems in future work.

Acknowledgments

This work 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

Susan Robinson, Antonio Roque, Ashish Vaswani, Charles Hernandez, Bill Millspaugh, and David Traum, "Evaluation of a Spoken Dialogue System for Virtual Reality Call For Fire Training," Submitted, 2006.

Antonio Roque, Anton Leuski, Vivek Rangarajan, Susan Robinson, Ashish Vaswani, Shri Narayanan, David Traum, "Radiobot-CFF: A Spoken Dialogue System for Military Training," 9th International Conference on Spoken Language Processing (Interspeech 2006 - ICSLP), Pittsburgh, PA, September 17-21, 2006.

Antonio Roque and David Traum, "An Information State-Based Dialogue Manager for Call for Fire Dialogues," 7th SIGdial Workshop on Discourse and Dialogue, Sydney, Australia, July 15-16, 2006.

David Traum and Staffan Larsson, 2003. The Information State Approach to Dialogue Management. In R. Smith & J. van Kuppevelt (eds.) Current and New Directions in Discourse and Dialogue. Dordrecht: Kluwer, 325-353.

Dialogue management for cooperative, symmetrical human-robot interaction

Mary Ellen Foster
Robotics and Embedded Systems Group
Department of Informatics, Technical University of Munich
Boltzmannstraße 3, 85748 Garching, Germany
foster@in.tum.de

Abstract

We describe the JAST human-robot dialogue system, which supports fully symmetrical collaboration between a human and a robot on a joint construction task. We concentrate on the dialogue manager, which is based on Blaylock and Allen's (2005) collaborative problem-solving model of dialogue and which supports joint action between the dialogue participants at both the planning and the execution levels.

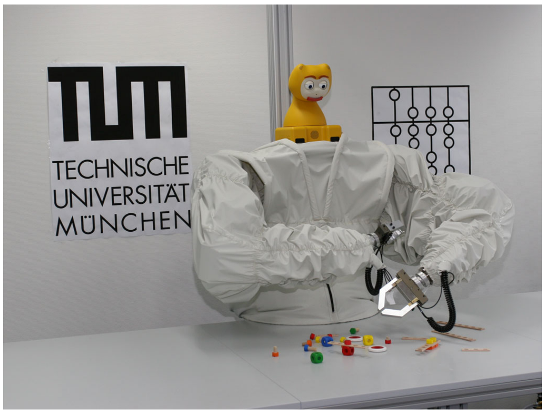

Figure 1: The JAST human-robot dialogue system

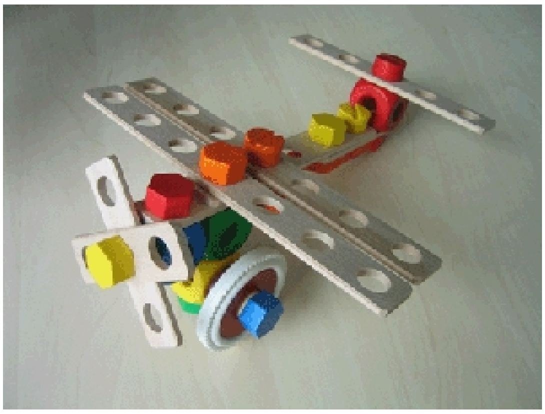

Figure 2: Assembled Baufix airplane

1 Human-robot dialogue in JAST

The overall goal of the JAST project ("**J**oint **A**ction **S**cience and **T**echnology"; http://www.euprojects-jast.net/) is to investigate the cognitive and communicative aspects of jointly-acting agents, both human and artificial. The JAST human-robot dialogue system (Foster et al., 2006) is designed as a platform for integrating the project's empirical findings on cognition and dialogue with its work on autonomous robots, by supporting symmetrical human-robot collaboration on a joint construction task.

The robot (Figure 1) consists of a pair of mechanical arms, mounted to resemble human arms, and an animatronic talking head capable of producing facial expressions, rigid head motion, and lip-synchronised synthesised speech. The system input channels are speech recognition, object recognition, and face tracking; the outputs include synthesised speech, facial expressions and rigid head motion, and robot actions. The human user and the robot work jointly to assemble a Baufix wooden construction toy (Figure 2), coordinating their actions through speech, gestures, and facial motions. Joint action may take several forms in the course of an interaction: for example, the robot may ask the user to provide assistance by holding one part of a larger assembly, or may delegate entire sub-tasks to be done independently. In the current version of the system, the robot is able to manipulate objects in the workspace (e.g., picking them up, putting them down, or giving them to the user) and to perform simple assembly tasks.

brandial 2006: Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue, pages 183–184.

2 Dialogue management in JAST

The JAST human-robot dialogue system has several features that distinguish it from many existing dialogue systems. First, the roles of the user and the robot are, in principle, completely symmetrical at all levels: either agent may propose a goal or a strategy for addressing one, and either—or both—may perform any of the actions necessary to achieve it. Also, the interaction must deal with both the selection of the actions to take in the execution of those actions, and may switch between the two tasks at any point. Finally, *joint action* is central to the dialogue at all levels: the participants work together to create domain plans, and also jointly execute the selected plans.

The distinctive requirements of the JAST dialogue system are most similar to those addressed by Blaylock and Allen (2005) in their collaborative problem-solving (CPS) model of dialogue. In collaborative problem solving, multiple agents jointly select and pursue goals, in three interleaved phases: selecting the goals to address, choosing procedures for achieving the goals, and executing the selected procedures. The central process in the CPS model is the selection of values (or sets of values) to fill roles, such as the goal to pursue or the allocation of sub-tasks among the participants. Slot-filler negotiations of this sort make up a large part of collaborative communication.

Dialogue management in the JAST system is based on this CPS model. As in COLLAGEN (Rich et al., 2001), the JAST dialogue state consists of three parts: the active set of goals and procedures, a set of open issues, and the interaction history. An *open issue* corresponds to any request, proposal or action that has occurred during the course of the dialogue and that has not yet been fully addressed; these are essentially the same objects as Ginzburg's (1996) *questions under discussion* (QUD). As an interaction proceeds, two parallel processes are active: the participants must complete domain goals such as locating and assembling objects, and must also address open issues that arise during the conversation. These two processes are tightly linked; for example, if an agent proposes a procedure for a particular subgoal and the other agrees (and closes the open issue), the next step in the interaction is likely to be executing the agreed-upon sequence of actions. Similarly, when an sub-goal is completed, the participants must address the open issue of how to proceed. The dialogue manager therefore maintains explicit links between the open issues and the current state of the domain plan to enable information to flow in both directions.

3 Current status and future work

At the moment, an initial dialogue-manager prototype based on the CPS model has been implemented in Java. This prototype supports a limited range of simple interactions with a cooperative user, using template expansion to create the domain plans. We are currently developing a more full-featured interaction manager, using a hierarchical planner to create the action sequences. As the system develops, we aim to expand its coverage to support phenomena such as failed actions and incorrect beliefs about the world, and to increase its robustness on incomplete or ill-formed messages from the input-processing modules.

Once a full working dialogue system has been developed, we intend to use it to implement and test the findings from the human-human joint-action dialogues that are currently being recorded and analysed by other participants in the JAST project; for example, we hope to derive strategies for confirmation, grounding, role assignment, and error handling. We will then perform a range of user studies to compare the success of the different strategies, as well as to measure the impact of factors such as feedback from the talking head, using both objective task-success measures and subjective measures of satisfaction and engagement.

Acknowledgements

This work is supported by the EU FP6 IST Cognitive Systems Integrated Project "JAST" (FP6-003747-IP). Thanks to Candy Sidner, Ron Petrick, Oliver Lemon, Geert-Jan Kruijff, Colin Matheson, Jon Oberlander, and the BranDial 2006 reviewers for useful comments and advice.

References

N. Blaylock and J. Allen. 2005. A collaborative problem-solving model of dialogue. In L. Dybkjær and W. Minker, editors, *Proceedings, 6th SIGdial Workshop on Discourse and Dialogue*, pages 200–211.

M. E. Foster, M. Rickert, and A. Knoll. 2006. Human-robot dialogue for joint-action construction tasks. In *Proceedings, 8th International Conference on Multimodal Interfaces (ICMI 2006)*. To appear.

J. Ginzburg. 1996. Dynamics and the semantics of dialogue. In J. Seligman and D. Westerstahl, editors, *Language, Logic and Computation, Volume 1*, CSLI Lecture Notes.

C. Rich, C. L. Sidner, and N. Lesh. 2001. COLLAGEN: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, 22(4):15–25.

Hierarchical Reinforcement Learning of Dialogue Policies in a development environment for dialogue systems: REALL-DUDE

Oliver Lemon and Xingkun Liu
School of Informatics
Edinburgh University
olemon,xliu4@inf.ed.ac.uk

Daniel Shapiro and Carl Tollander
CSLI/Applied Reactivity
Stanford University
dgs,carl@appliedreactivity.com

Abstract

We demonstrate the REALL-DUDE system[1], which is a combination of REALL, an environment for Hierarchical Reinforcement Learning, and DUDE, a development environment for "Information State Update" dialogue systems (Lemon and Liu, 2006) which allows non-expert developers to produce complete spoken dialogue systems based only on a Business Process Model (BPM) and SQL database describing their application (e.g. banking, cinema booking, shopping, restaurant information, ...). The combined system allows rapid development and automatic optimization of spoken dialogue systems. Hierarchical Reinforcement Learning (RL) has not been applied to the problem of dialogue management before. It provides a way of dramatically reducing the size of the state space to be considered in RL problems. REALL-DUDE thus allows iterative development of dialogue policies through Hierarchical RL to be combined with a development environment for complete dialogue systems, encompassing parsing, speech recognition, synthesis, and dialogue management.

1 Introduction

It has been shown in previous work (Singh et al., 2002) that dialogue policies obtained by Reinforcement Learning (RL) can improve over hand-coded dialogue managers. However, a key problem in RL applied to dialogue management is the very large policy spaces generated by the dialogue management problem. REALL 's key source of power is its ability to constrain learning with background knowledge, within a principled framework. It has been shown (Shapiro and Langley, 2002) that this approach generates three order of magnitude reductions in problem size, and two order of magnitude improvements in learning rate, relative to the common formulation of RL tasks which offers all feasible options in all possible situations.

We demonstrate a development environment for dialogue systems which allows iterative development and refinement of dialogue policies through Hierarchical RL. We present the concepts behind REALL and DUDE , and show how to use DUDE to generate complete spoken dialogue systems (Lemon and Liu, 2006). We then demonstrate learning experiments that explore dialogue policies in the presence of different reward signals and channel noise characteristics, and show how the learner acquires different optimized policies.

2 REALL – Reactive Planning and Hierarchial RL

REALL is a language for defining extremely reactive agent behavior. It consists of a representation for expressing hierarchical, goal-oriented plans, together with an interpreter for evaluating those plans that operates in a repetitive loop. This iteration supplies reactivity: even if the world changes radically between two execution cycles, REALL will find a goal-relevant action to employ.

REALL is also a learning system. Because its interpreter contains a model-free reinforcement learning algorithm, every REALL agent has the ability to acquire an action policy from delayed reward. Programmers can access this capability

[1]This research is supported by Scottish Enterprise under the Edinburgh-Stanford Link programme.

by writing plans with disjunctive elements, and by embedding those choice points in hierarchical plans. As a result, REALL offers a means of invoking learning in the context of background knowledge, and this constrains the learning task.

Because REALL is a learning system, it supports a novel development metaphor called programming by reward. Here, the programmer may encode a dialogue strategy with options, and specify reward functions that serve as the targets of optimization. Via a training period, the reward functions select one of the many policies implicitly contained in the REALL plan, and developers can obtain distinct behaviors by making small changes to the reward functions (Shapiro et al., 2001).

REALL learns a policy by finding the best action to take in every state. It learns the value of a given state-action pair by sampling its future trajectory, and it represents this value using a linear function of currently observable features. REALL bootstraps: it updates the estimate for a state-action pair using its current value, the current reward, and the estimate associated with the next state-action pair. Over time, these estimates converge to their appropriate values.

3 The DUDE development environment

The contribution of DUDE (Lemon and Liu, 2006) is to allow non-expert developers to build ISU dialogue systems using only the Business Process Models (BPMs) and databases that they are already familiar with, as shown in figure 1.

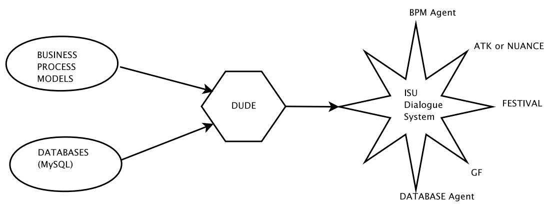

Figure 1: The DUDE development process

The environment includes a development GUI, automatic generation of Grammatical Framework (GF) grammars for robust interpretation of spontaneous speech, and uses application databases to generate lexical entries and grammar rules. The GF grammar is automatically compiled to an ATK or Nuance language model for speech recognition. See (Lemon and Liu, 2006) for details.

The power of REALL-DUDE is to embed Hierarchical Reinforcement policy learning and optimization from REALL within the rich development environment supplied by DUDE .

4 Demonstrating learning

We will present a REALL program, `Slotfiller`, embedded in the DUDE environment, which contains a scaffolding of required dialogue behavior (e.g., confirmations, clarifications, mixed-initiative questions). The demonstration presents a variety of learning experiments that explore these decisions in the presence of different reward signals and channel noise characteristics. We will show how the learner acquires and optimizes distinct dialogue policies in each case.

5 Conclusion

Hierarchical RL has not been applied to the problem of dialogue management before. It provides a principled way of dramatically reducing the size of the state space to be considered in RL of dialogue management. Here we demonstrate a development environment, REALL-DUDE , which combines RL for optimization of dialogue policies with a full development environment for automatic generation of spoken dialogue systems. We will demonstrate how to develop complete spoken dialogue systems using DUDE and then we will demonstrate strategy learning for those systems using REALL, which optimizes policies for different noise and reward conditions in dialogue.

References

Oliver Lemon and Xingkun Liu. 2006. DUDE: a Dialogue and Understanding Development Environment, mapping Business Process Models to Information State Update dialogue systems. In *Proceedings of EACL (demonstration systems)*.

D. Shapiro and P. Langley. 2002. Separating skills from preference: using learning to program by reward. In *Nineteenth International Conference on Machine Learning*.

Dan Shapiro, Pat Langley, and Ross Shachter. 2001. Using background knowledge to speed reinforcement learning. In *Fifth International Conference on Autonomous Agents*.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of Artificial Intelligence Research (JAIR)*.