View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Publications at Bielefeld University

Towards Closed Feedback Loops in HRI: Integrating InproTK and PaMini

Birte Carlmeyer
Applied Informatics Group
Dialogue Systems Group
Bielefeld University
Germany
bcarlmey@techfak.uni-bielefeld.de

David Schlangen
Dialogue Systems Group
Bielefeld University
Germany
david.schlangen@uni-bielefeld.de

Britta Wrede
Applied Informatics Group
Bielefeld University
Germany
bwrede@techfak.uni-bielefeld.de

ABSTRACT

In this paper, we present a first step towards incremental processing for modeling asynchronous human-robot interactions, to allow closed feedback loops in HRI. We achieve this by combining the incremental natural language processing framework InproTK with the human-robot dialog manager PaMini, which is based on generic interaction patterns. This enables the robot to provide incremental feedback during interaction and allows the user to give online feedback and corrections. We provide a first realization scenario as a proof of concept for our approach.

Categories and Subject Descriptors

I.2.9 [**Artificial intelligence**]: Robotics—*Operator interfaces*; H.5.2 [**Information interfaces and presentation**]: User Interfaces—*natural language*; I.2.7 [**Artificial intelligence**]: Natural Language Processing

Keywords

Dialog management, HRI, incremental processing, multimodal systems, spoken dialog

1. INTRODUCTION

According to current research trends, robots will become part of our daily private and occupational lives. While very simple robots, such as vacuum cleaning robots[1] or reactive social robot animals such as the seal robot Paro[2], already belong to the daily lives of a range of households or nursery homes, more functional and interactively controllable robots are still lacking. This is because both the functional as well as the interactive capabilities are still not sufficiently robust. In our contribution we propose a new approach to pave the way for more robust interaction capabilities.

One important feature of interactive systems is their "closed loop" character, which means that the system has to be capable of reacting to an input given by the user and of providing a feedback that again affects the user's next actions [8]. Such closed loop interactions allow, among others, the system to learn or to adapt to the user. In active learning approaches – where the learning system itself chooses the questions to be asked to a tutor in order to maximize its performance – it has been shown that it is beneficial if the segments of information that are requested by the system are small, leading to a system that is more frequently updated by smaller chunks and thus achieving earlier a better performance [19]. Similarly, one can observe in human interactions that feedback signals are not issued in long information blocks. Rather, feedback signals are provided in parallel and via a range of different channels or modalities.

Consider, for example, a simple command like "Can you give me the blue bottle, please?". In current systems this utterance would first have to be completely recognized and parsed before it will be forwarded to the robot system components. Once it is in the processing pipeline, a correction is no longer possible. The user has to wait, because the next utterance can not be given before the execution of the action is ended. This example provides several points of interruption, though. On the one hand, the user could correct or detail out her/his utterance, e,g. by adding "that one next to you" (specification) or "I mean the orange one" (correction). On the other hand, the user could still start to correct the robot directly after the robot has initiated the action.

For both kinds of interruptions, incremental processing is required. In the first case, the subsequent parsing component would have to be able to relate the subsequent utterance to the first one and to classify it as a specification or correction of the referent of the first utterance. In that case the following processing modules would have to be able to change the targeted reference and possibly to inform the user of this change. In the second case, the action module would need to be able to interrupt its current action, e.g. by retracting the arm or simply by stopping it.

Because of this, incremental processing in such a complex system is an entirely unknown problem so far. In the following we will discuss in more detail the requirements that such a behavior would impose on the dialog management components and how these can be adapted in order to addresses

[1]http://www.irobot.de/shop/shop
[2]http://www.parorobots.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$15.00.

the raised issues.

2. RELATED WORK

In human-robot interaction (HRI) verbal interaction has been strongly neglected and dialog modeling is only an emerging research topic. Where dialog modeling is done, the dialog is typically in the style of command and control. This means that the user issues a command to the robot which is then either carried out or not understood. Clarification does often not take place. Yet, experience shows that such interactions leave it to the user to figure out on her/his own what kinds of commands can be given to the robot and how they are interpreted. Alternatively, the user has to be briefed generally by specific command examples, leading to a very stereotyped interaction that does not allow for subtle corrections which makes human interaction so efficient. This leads to a brittle and hesitant interaction with short and well-considered utterances, such as stereotypized commands or brief confirmations or negations. In contrast, in order to allow for a more natural dialog style the system has to deal with a range of linguistic phenomena, such as speech disfluencies [14] or parallelized interaction signals. For instance, pauses, repetitions or corrections in human speech are still an unsolved challenge in dialog modeling. Parallelized, non-verbal interaction signals such as facial expressions (e.g. sudden surprise or annoyance) or verbal cues such as an interrupting abortion command are also difficult to handle. These challenges do not only affect the speech recognition module but also the sub-sequent processing components, especially the dialog modeling component which has to take decisions based on the results of previous components.

Compared to other application areas, dialog modeling on robots is a highly complex and yet mostly unsolved task as this component has to manage the interface between a range of asynchronous processing events resulting from the internal robot components on the one side with a range of asynchronous processing events resulting from the interaction with the user on the other side. So far, no systematic solution to this challenge has been developed. Due to this high complexity the vast majority of HRI dialog approaches are based on finite state machines or statecharts [2, 17] with the major advantage of simplicity and intuitiveness for the developer. However, in order to model less restricted interactions, which introduces more states, the dialog graph has to be enriched leading to a fast growing population of states which become difficult to handle.

More sophisticated approaches to deal with this challenge are based on Bayesian networks or partially observable Markov decision processes (POMDP) [7, 9, 11] which focus on modeling the highly dynamic and uncertain environment that robots have to deal with. However, the probability parameters need to be learned or else be handcrafted which tends to be quite expensive for more complex interactions.

Traum and Larson [21] propose the information state approach for dialog management in order to account for updates of information through the ongoing interaction. However, this approach does not provide a systematic solution for the asynchronous nature of the underlying processes in HRI. Peltason and Wrede [13] propose an approach to dialog management that is based on generic interaction patterns. These provide a generalized description of interaction structure, which can be configured easily for a concrete scenario and have been targeted to deal with asynchronous events of the robot internal processes which can evoke specific dialog actions. However, this approach does not facilitate incremental processing which is required to achieve more natural interactions in HRI [5]. In spoken dialog systems, incremental speech processing has been explored e.g. by Aist et al. [1], Schuler et al. [16] or Skantze and Hjalmarsson [18]. Schlangen and Skanze propose a general abstract model of incremental processing [15], which is partially implemented by Baumann and Schlangen [3]. However, this approach does not take into account the asynchronous nature of the underlying robot processing modules which need to be coordinated in order to allow for incremental interaction processing.

In the application field of virtual agents incremental processing is often operationalized in terms of parallelized feedback [20]. The feedback of these virtual agents can have different modalities, for example verbal utterances (paraverbals e.g. "mhm" or short statements such as "I see") or non-verbal signals such as head nods. These backchannels permit a more natural interaction with these conversational agents. In contrast to humans, where these cognition responses are results from incremental processing of the communication signals, here they are modeled by independent processing modules which focus on surface features such as prosody [4] or gaze, but are agnostic to the semantic or pragmatic content of the utterance. This can lead to contradicting signals, with the feedback module producing continuous positive feedback while the semantic speech processing module cannot make sense of the input and finally produces a request for clarification or does not react at all.

In short, in order to enable a robot to provide meaningful incremental feedback during interaction with a user we have identified two necessary abilities: on the one hand, it needs to be able to process the verbal and non-verbal user input in an incremental way, on the other hand, the asynchronous robot system internal events need to be managed and interleaved with the interaction relevant events.

To achieve this, we propose to combine (1) the "Incremental Processing Toolkit" (InproTK) [3] for the front-end incremental processing chain with (2) PaMini, a framework for human-robot interaction management based on generic interaction patterns that handles asynchronous internal robot system processing events and thus provides an interface to the back-end processing modules.

3. INCREMENTAL SPEECH PROCESSING AND ASYNCHRONOUS DIALOG MANAGEMENT

In this section, we will introduce in more detail the existing speech processing InproTK and robot dialog component PaMini as a basis for the incremental robot dialog management system described in the following section.

3.1 InproTK

The incremental speech processing toolkit InproTK [3] is capable of dealing with incremental output from a speech recognition system and to insert or change existing processing results when updates (i.e. corrections) are sent by the previous processing modules. This is achieved by the definition of incremental units (*IUs*), organized in buffer sequences [15]. InproTK realizes this *IU*-model of incremental processing, where incremental systems consist of a

network of processing modules (see fig. 1) that work on incremental units. These serve as the basic units which can be subject to (post-hoc) changes during processing, e.g. affecting incrementally produced ASR results or subsequent syntactic or semantic parsing results. The underlying idea is based on a sequence of processing modules, which have a *left buffer* and a *right buffer*. These modules take input from their *left buffer*, perform some kind of processing and provide output on the *right buffer*, which can be the input for the next processing module. The modules exchange data in form of *incremental units* (*IUs*). Incremental units are the

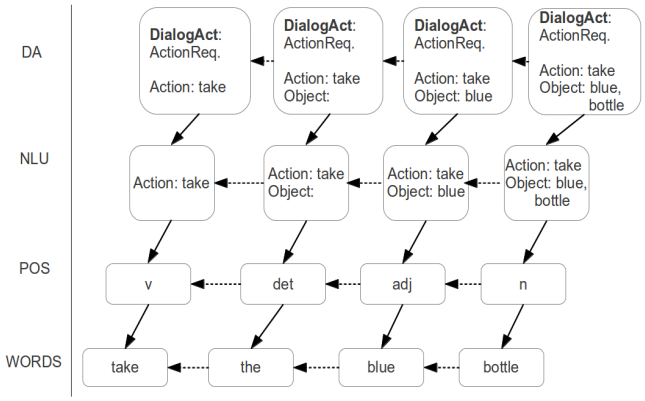

Figure 1: Example of an *IU* network: It describes the processing pipeline of *IU*s from the ASR results *wordIU* through to *DialogActIU*. Dashed arrows represent *grounded-in links*, solid arrows are *same level links*.

smallest 'chunks' of information that can be passed between connected modules and trigger actions. The length of such chunks thus determine the level of incrementality of the system. Note that modules have to be able to react on three different states of the *IUs*: *added* - indicating that a new unit has entered the processing module, *revoked* - indicating that a previously added unit has been updated or revoked and *committed* – indicating that an incremental unit has been finally committed and will not be changed any more. After an *IU* is added to a buffer it is still possible that the previous module changes its hypothesis. In such a case the *IU* is revoked and a new may be generated. If an incremental unit is marked as committed, it is considered unalterable and can not be revoked. Importantly, *IUs* can be part of a larger unit, e.g. words that can be combined to a phrase. The *IUs* have two different kinds of connections. *Same level links* connect *IUs*, which are produced by the same module and reflect their temporal order. *Grounded-in links* represent on which *IUs* they depend, e.g. a phrase depends on individual words, thus representing the possibility to build a hierarchical structure.

Additionally to the processing modules, INPROTK provides *listener* and *informer* modules [10] that connect the speech processing module with other system components, which send events over the network (generally via a specific middleware such as the *Robotics Service Bus* [22]). *Listeners* receive information from the network, combine this data into an *IU* and put them onto their *right buffer*. *Informers* otherwise publish the information of all *IUs* from their *left buffer*.

3.2 Dialog Manager PaMini

PaMini [13] is a framework for assembling mixed-initiative human-robot interaction from generic *interaction patterns* and provides, through the mechanism of *interaction patterns*, a generic interface to the back-end, i.e. the robot system. Each *interaction pattern* describes recurring dialog structures and is a configurable building block of the interaction. It can be understood as a basic unit for grounding information (also in terms of action) within different kinds of clarification subroutines, thus avoiding the over-generalizing concept of presentation-acceptance pairs (as suggested by Clark [6]) and specifying more specific recurring grounding patterns. The Interaction Patterns of PaMini can be formalized as an extended form of a finite state machine augmented with internal state actions. A schematic graphical representation of simple action request by the human user *interaction pattern* is sown in figure 2. The finite state machine takes human dialog acts and task events as input and produces robot dialog acts as output. These dialog acts can have different modalities, like verbalization, mimic or deictic gestures. In this example an action request is initiated by the human. The dialog manager initiates the corresponding system task, e.g. a grasp action and can notify about the task state update and acknowledge task execution.

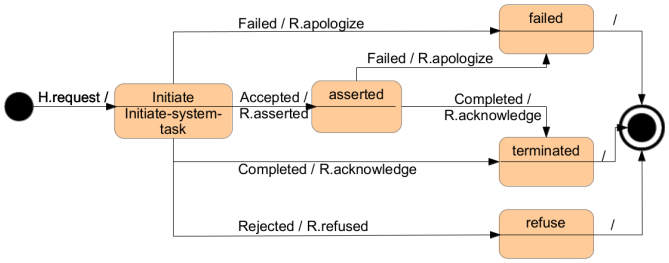

Figure 2: *Human Simple Action Request pattern*: The finite state machine describes an action request initiated by the human. If the state machine enters the *initiate* state, the dialog manager initiates the corresponding system task. Depending on the feedback of the respective system component, different responses from the robot are possible.

For a concrete scenario the *interaction patterns* have to be configured. The human dialog acts, robot dialog acts and the tasks have to specified. During the interaction the different *interaction patterns* may be interleaved, which provides more flexibility. For instance while the robot executes a grasp action, the human can ask interposed question.

4. CURRENT STATE

The incremental processing toolkit INPROTK provides opportunities for incremental speech recognition and incremental speech synthesis. The dialog manager PaMini provides a generalized description of human-robot interaction structure and needs events, e.g. speech recognition results, to decide

how the robot should (re-)act. To combine InproTK and PaMini, we have to integrate some extensions in both systems.

4.1 InproTK Extensions

InproTK and PaMini can both communicate via the *Robotics Service Bus* (RSB), a message-oriented, event-driven communication middleware [22]. InproTK has the concept of *informers*, which allows to publish informations e.g. via RSB. For the communication with the dialog manger PaMini some kind of dialog acts needed to be sent. As explained in [10] only strings can be sent by these informers. We created a new custom data type *dialog act* for the communication via RSB. A *dialog act* can be in state *added*, *revoked*, *updated* or *committed*, corresponding to the states of *IUs* in InproTK. We implemented a new informer for sending this *dialog act* via RSB. Also a new module for the dialog act generation was needed. For the first step we implemented a simple dialog act generation based on keyword-spotting.

Another extension is a new *listener*, which receives the verbalization output from PaMini. It splits incoming verbalizations into small phrases and passes its outcome to the incremental speech synthesis (iSS) module. Furthermore the *listener* can stop the speech synthesis by revoking the *IUs* from the right buffer of its following processing module.

4.2 PaMini Extensions

Normally, PaMini processes human input as following: When human input is received, the dialog manager checks whether the given human input matches one of the configured *interaction patterns*. If no pattern matches with the given input, PaMini starts a clarification strategy, e.g. triggers a pattern that asks the human to repeat the utterance. PaMini in the original incarnation cannot handle incremental input, for instance input which is not based on a complete utterance and may be updated or revoked.

In our first step towards integrating incremental speech processing in the dialog process we have addressed the following two issues: (1) For the communication with InproTK, PaMini has to distinguish between the different states of the incoming dialog acts (i.e. added, revoked, updated, committed). (2) In order to allow the user to interrupt the robot, system components (pertaining to both, verbal as well as non-verbal actions) need to be interruptable during execution.

We addressed the first issue by extending PaMini to react to different states of dialog acts. In the first implementation, PaMini only reacts to input whose state is committed. For this, we created a new input source in PaMini that reacts on the dialog acts of InproTK. If the incoming dialog act is committed, then the input can be processed as usual. Otherwise a more careful processing strategy is needed. Depending on the input we can possibly produce some output, but we have to take into account that the dialog act may be revoked, when the dialog act generation module changes its hypothesis based on new input.

In order to achieve interruptability, we integrated a new *interaction pattern*, which makes it possible to stop a current verbalization (and is extendable to other system actions). If this pattern is triggered, it sends an appropriate command to the listener module of InproTK. This clears the left buffer of the *incremental speech synthesis* module, leading to a interrupted synthesis.

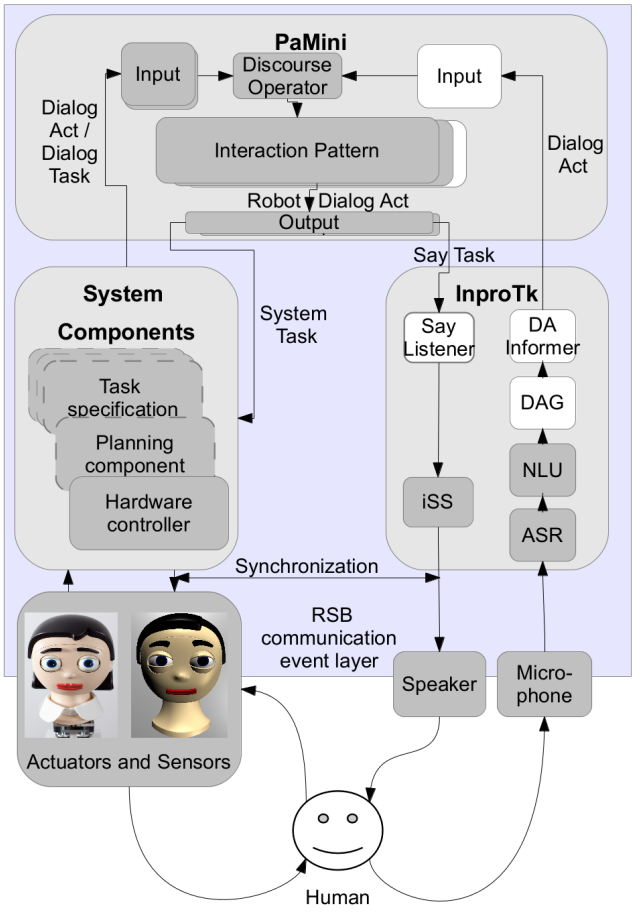

Figure 3: Processing pipeline and communication between InproTK, PaMini, other system components and the human. It is explain in more detail in section 4.3. The white modules are extensions explained in sections 4.1 and 4.2. Dashed modules are not yet integrated.

4.3 Architecture

Figure 3 gives an overview of the architecture, starting with the microphone input through to the speaker output (bottom right corner). The microphone input is used by the *incremental speech recognition* module (ASR) of InproTK, which produces incremental units. These results are processed by the *natural language understanding* module (NLU), which provides its outcomes to the *dialog act generation* module (DAG). These dialog acts are sent by the *dialog act informer* via RSB and interpreted by PaMini. Other sensors such as the cameras of the robot can also perceive the human and send additional informations to the dialog manger. The *discourse operator* of PaMini receives this input and processes it in sequence. It activates a suitable *interaction pattern* and depending on the configuration of this pattern a robot dialog act is produced. If the robot dialog act contains a verbalization output, a say task is sent via RSB and further processed by the *incremental speech synthesis* module (iSS) of InproTK. Other outputs such as facial expressions or other system tasks (e.g. a grasp action) are sent to the

H:	Hello Flobi. *(When the human starts speaking, the robot immediately starts to raise its head.)*
R:	Hello. Nice to meet you.
H:	Tell me something about yourself.
R:	My name is Flobi and I'm an anthropomorphic robot had and I'm able to to show ...
H:	Stop, that is enough. *(Interaction pattern is interrupted and reset)*
R:	Ok.

Table 1: Interaction example between the human (H) and the robot (R).

corresponding system component. To move the robots lips synchronously with the speech synthesis, a first synchronization between the hardware control and the sound synthesis can easily be applied at this level.

4.4 First Scenario

We implemented a first very basic scenario for testing the processing pipeline and as proof of concept for our approach. An interaction example between the human and the robot is shown in table 1. In this first scenario the robot is rather reactive. The user can greet or say good bye and can give some instructions (e.g. "tell me something about yourself"). With the current architecture the robots behavior is now more interactive. It is able to give a first basic feedback based on the speech processing and furthermore the user is able to give online feedback. The robot platform "Flobi" [12], an anthropomorphic robot head which is able to show facial expressions, is used. This makes it possible to produce multimodal output. As can been seen in figure 3 we can use the robot head (left) or its simulation (right).

New interaction opportunities emerge from the integration of the *incremental speech synthesis* (iSS) module of InproTK. Previously it was impossible to interrupt an ongoing robot verbalization. A dialog act can have any number of utterances, but the dialog manager had no opportunity to interrupt its own current robot dialog act. The new *interaction pattern* allows such an interruption by the user saying something like "stop, that's enough".

The incremental results of the speech recognition allow the system to react faster. As a first approach the robot gives nonverbal feedback to the human. An obvious use case is to empathize attention. If the incoming dialog act has the state *added* the robot can react even though the dialog act may be revoked. In this case the robot shows attention, e.g. looks at the person to demonstrate responsiveness.

5. OUTLOOK AND CHALLENGES

PaMini provides us with generalized interaction modeling capabilities and InproTK allows us to use incremental processing. On the basis of this first combination and extension of PaMini and InproTK new possibilities in modeling human-robot interaction are achievable. These allow more natural interactions, but also lead to new challenges in dialog modeling and overall system architecture. In conclusion this means that every component needs to handle incremental information to achieve closed loop behavior in its area of responsibility.

5.1 Benefits

An incremental dialog manager allows multiple improvements. First of all, the system has the possibility to give quicker feedback. The human input is processed earlier, which makes it possible to react while the person is still speaking. Listening behavior (e.g. nodding) can be triggered based on the speech understanding and not only focused on low level features (like voice pitch), which previously could lead to contradictory situations with positive, concurrent feedback and later negative feedback.

Furthermore other feedback is conceivable. Resuming the human-robot interaction from the beginning of this paper, different robot actions become possible before the complete task is completely verbalized. If the human starts to give the robot an assignment, the accomplishment may be initiated although not all information for completion is available. For instance the robot can prepare a grasp task and bring its gripper arm into the right position during the instruction. Even if the task is ambiguous and multiple different objects could be grasped, the robot can start to prepare grasping the one most likely to be the one that's intended. During this preparation a clarification procedure can be performed. For example the robot can ask the human directly or use other modalities. Watching the person during the grasp may prompt the interaction partner to give online corrections or resolve uncertainties. While the robot prepares its grasp, the human can give additional information or correct the robots behavior. Furthermore new possibilities of action planning emerge. Potentially planning components could evaluate possibilities even though the specification of the task is still incomplete.

5.2 Challenges

This new interaction possibilities give rise to new challenges in system modeling. First of all the dialog management has to deal with adaption or retraction of dialog acts. It is still possible, that the dialog manager only reacts to unalterable results, but for closed feedback loops it is necessary to also react to incomplete incremental results in some reasonable way. When such an incremental result was wrong and needs to be revoked, the system should be able to revert unnecessary actions and possible comment on them.

For the dialog management to handle such situations, the concept of *interaction pattern* needs to be extended. In some cases it can be sufficient to take a step backwards in the process flow of an *interaction pattern*, e.g. if the corresponding output is not yet produced. However besides that, it could be necessary to "repair" an already performed dialog output. This can happen in various ways. For verbal self-repairs, there are different strategies, like saying "Excuse me, I mean...". But at the latest when other actions were triggered, for instance the systems grasp planning component was instructed to start an action, verbal self-repairs are not sufficient. In this case the action has to be stopped and the old system status has to be restored. The grasp planning component in this example has to move the gripper back into a safe position. In order to realize this, feedback and interruption capabilities are mandatory for all system components, which are involved in the incremental process.

When planning with incomplete task specifications, a balance must be found between the benefit of fast task completion and the costs of repairing overhasty actions. Resuming the grasp action, the planning component has to determine

how far the grasp can be performed while achieving this balance.

Another challenge is the synchronization of the dialog output. Because of the faster feedback it is even more important to synchronize the different output modalities. PaMini itself does not provide capabilities for the simultaneous execution of different modalities of the same robot dialog act. The execution system components have to take the responsibility for it. Therefor the actuators of InproTK (e.g. the synthesis module) and the other actuators of the system (e.g. head movement control) have to be closely connected.

6. CONCLUSIONS

We have presented a proof of concept for an approach to combine two different dialog systems to realize incremental processing and modeling of asynchronous human-robot interactions. Both systems have been presented separately and the necessary extensions for their first functional interaction have been propound. Furthermore we described an initial interaction scenario using the new interaction and modeling possibilities. At last we gave an outline and discussed various benefits and challenges, which are implicated by this new possibilities of interaction.

7. REFERENCES

[1] G. Aist, J. Allen, E. Campana, L. Galescu, C. A. G. Gallo, S. Stoness, M. Swift, and M. Tanenhaus. Software architectures for incremental understanding of human speech. interspeech 2006. In *In Proceedings of Interspeech/ICSLP*, 2006.

[2] A. Bauer, D. Wollherr, and M. Buss. Information retrieval system for human-robot communication-asking for directions. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4150–4155. IEEE, 2009.

[3] T. Baumann and D. Schlangen. The inprotk 2012 release. Proceedings of the NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012), pages 29–32. ACL, 2012.

[4] E. Bevacqua, E. Sevin, S. J. Hyniewska, and C. Pelachaud. A listener model: introducing personality traits. *Journal on Multimodal User Interfaces*, 6(1-2):27–38, Apr. 2012.

[5] T. Brick and M. Scheutz. Incremental natural language processing for hri. In *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*, pages 263–270, March 2007.

[6] H. H. Clark. *Using language*, volume 1996. Cambridge University Press Cambridge, 1996.

[7] F. Doshi and N. Roy. Efficient model learning for dialog management. In *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*, pages 65–72, March 2007.

[8] H. Dubberly, U. Haque, and P. Pangaro. What is interaction? Are there different types? *Interactions Magazine*, 2009.

[9] J.-H. Hong, Y.-S. Song, and S.-B. Cho. A hierarchical bayesian network for mixed-initiative human-robot interaction. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3808–3813, April 2005.

[10] C. Kennington, S. Kousidis, and D. Schlangen. Inprotks: A toolkit for incremental situated processing. Proceedings of SIGdial 2014: Short Papers, pages 84–88, 2014.

[11] P. Lison. *Structured Probabilistic Modelling for Dialogue Management*. PhD thesis, University of Oslo, 2013.

[12] I. Lütkebohle, F. Hegel, S. Schulz, M. Hackel, B. Wrede, S. Wachsmuth, and G. Sagerer. The bielefeld anthropomorphic robot head "flobi". In *2010 IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 03/05/2010 2010. IEEE, IEEE. accepted.

[13] J. Peltason and B. Wrede. Pamini: A framework for assembling mixed-initiative human-robot interaction from generic interaction patterns. SIGDIAL 2010 Conference. Association for Computational Linguistics, 2010.

[14] D. Schlangen. Modelling dialogue: Challenges and approaches. *Künstliche Intelligenz*, 3/05:23–28, 2005.

[15] D. Schlangen and G. Skantze. A general, abstract model of incremental dialogue processing. *Dialogue and Discourse*, 2(1):83–111, 2011.

[16] W. Schuler, S. Wu, and L. Schwartz. A framework for fast incremental interpretation during speech decoding. *Comput. Linguist.*, 35(3):313–343, Sept. 2009.

[17] G. Skantze and S. Al Moubayed. Iristk: A statechart-based toolkit for multi-party face-to-face interaction. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, pages 69–76, New York, NY, USA, 2012. ACM.

[18] G. Skantze and A. Hjalmarsson. Towards incremental speech generation in dialogue systems. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–8. Association for Computational Linguistics, 2010.

[19] K. Tomanek, J. Wermter, and U. Hahn. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *Proc. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 486–496. ACL, 2007.

[20] D. Traum, D. DeVault, J. Lee, Z. Wang, and S. C. Marsella. Incremental dialogue understanding and feedback for multi-party, multimodal conversation. In *The 12th International Conference on Intelligent Virtual Agents (IVA)*, Santa Cruz, CA, Sept. 2012.

[21] D. Traum and S. Larsson. The information state approach to dialogue management. In J. van Kuppevelt and R. Smith, editors, *Current and New Directions in Discourse and Dialogue*, volume 22 of *Text, Speech and Language Technology*, pages 325–353. Springer Netherlands, 2003.

[22] J. Wienke and S. Wrede. A middleware for collaborative research in experimental robotics. IEEE/SICE International Symposium on System Integration (SII2011), pages 1183–1190. IEEE, 2011.