

Interoperability and Knowledge Representation in Distributed Health and Fitness Companion Dialogue System

Jaakko Hakulinen

Department of Computer Sciences
33014 University of Tampere, Finland
jaakko.hakulinen@cs.uta.fi

Markku Turunen

Department of Computer Sciences
33014 University of Tampere, Finland
markku.turunen@cs.uta.fi

Abstract

As spoken dialogue systems move beyond task oriented dialogues and become distributed in the pervasive computing environments, their growing complexity calls for more modular structures. When different aspects of a single system can be accessed with different interfaces, knowledge representation and separation of low level interaction modeling from high level reasoning on domain level becomes important. In this paper, a model utilizing a dialogue plan to communicate information from domain level planner to dialogue management and from there to a separate mobile interface is presented. The model enables each part of the system handle the same information from their own perspectives without containing overlapping logic.

1 Introduction

Most existing spoken dialogue systems provide a single interface to solve a well-defined task, such as booking tickets or providing timetable information. There are emerging areas that differ dramatically from task-oriented systems. In domain-oriented dialogues (Dybkaer et al, 2004) the interaction with the system, typically modeled as a conversation with a virtual human-like character, can be the main motivation for the interaction. These systems are often multimodal, and may take place in pervasive computing environments where various mobile, robotic, and other untraditional interface are used to communicate with the system. For example, in the EU-funded COMPANIONS-project (Wilks, 2007)

we are developing a conversational Health and Fitness Companion that develops long-lasting relationships with its users to support their healthy living and eating habits via mobile and physical agent interfaces. Such systems have different motivations for use compared to traditional task-based spoken dialogue systems. Instead of helping with a single, well defined task, the system aims at building a long-term relationship with its user and providing support on a daily basis.

1.1 Mobile and Physical Agent Interfaces

New kinds of interfaces are used increasingly often in conjunction with spoken dialogue technology. Speech suits mobile interfaces well because it can overcome the limited input and output modalities of the small devices and can also better support using during the moments when their hand or eyes are busy. Physical agent interfaces, on the other hand, have been used in systems, which try to make dialogue systems more part of people's life. In many cases, they include rich multimodal input and output while providing a physical outlook for the agent. While naturalistic human-like physical robots are under development, especially in Japan, there is room for a variety of different physical interface agents ranging from completely abstract (e.g., simple devices with lights and sound) to highly sophisticated anthropomorphic apparatus. For example, Marti and Schmandt (2005) used several toy animals, such as bunnies and squirrels, as physical embodied agents for a conversational system. Other example is an in-door guidance and receptionist application involving a physical interface agent that combines pointing gestures with conversational speech technology (Kainulainen et al., 2005). Some physical agent technology has also

been commercialized. For example, the wireless Nabaztag™/tag rabbits (<http://www.nabaztag.com/>) have been success

Both mobile use and physical agent interface can support the goal of making a spoken dialogue system part of users' everyday life and building a meaningful relationship between the system and the user. It has been found that mere existence of a physical interface changes users' attitude toward a system and having access to a system throughout the day via a mobile interface is likely to further support this relationship.

In this work, we have used the Nabaztag as a multimodal physical interface to create a conversational Health and Fitness Companion and a mobile version interface for outdoor usage has been implemented on Windows Mobile platform.

1.2 Inter-component Communication and Knowledge Representation Challenges

In systems, where multiple interfaces can be used to access parts of the same functionality and the system interacts with a user many times over a long time period, modeling the interaction and domain easily becomes complex. For example, the system should model interaction history on a longer timescale than a single session. With multiple interfaces, at least some such information could be useful if they can be shared between the interfaces. Furthermore, the system must include a model capable of reasoning about the domain, and learn from the user and his or her actions to provide meaningful interaction, such as to provide reasonable guidance on user's health and progress as the user's condition alters over time in our case with the Health and Fitness Companion. Such reasoning should be concentrated on one component, instead of duplicating the logic to keep the system maintainable. Still, the information must be communicated over different interfaces and the component inside them. Therefore, modularization of the system and appropriate knowledge representation become vital.

On dialogue management level, a common way to take some complexity away from the dialogue manager and limit its tasks more specifically to dialogue management is to separate domain specific processing, such as database queries, into a back-end component. Many researchers have worked with separating generic dialogue management processes from the domain specific processes. Example solutions include shells (Jönsson, 1991) and object oriented programming methods (Salonen, et al., 2004, O'Neill, et al., 2003). On the other hand, a simple back-end

and an active user community has emerged around it.

interface, e.g., SQL queries, can be included as configuration parameters (Pellon et al., 2000). Since dialogue management is usually based on state transition networks, form filling, or some other clearly defined model, separating domain specific processing to the back-end makes it possible to implement dialogue management purely with the selected model.

Health and Fitness Companion, as discussed in the following, is based on a model where the domain specific module is more than just a simple interface and includes active processing of domain information, reasoning, learning, or other complex processes. We call such a component the cognitive model. While the task of a dialogue manager is to maintain and update dialogue state, the cognitive model reasons using the domain level knowledge. In our case, we have two dialogue managers, one for the home system with a physical interface agent and one for mobile system (yet another is in development, but not considered here). The two handle somewhat separate tasks but each provides input to another and the cognitive model. Separation of the task between the different parts is not trivial. For example, managing dialogue level phenomena, such as error handling and basic input processing, are tasks clearly in the areas of respective dialogue managers. However, cognitive modeling can help in error handling by spotting input that seems suspicious based on domain level information and input parsing by providing information on potential discussion topics. The solution we have devised is to have the cognitive model produce a dialogue plan for the dialogue management in home system. The dialogue management in the home system provides parsed user inputs to the cognitive model and to the mobile system. The mobile system provides similar input back to the home system, which communicates it back to the cognitive model.

In the following we describe the Health and Fitness dialogue system in general. Then we discuss the mobile interface, the dialogue manager of the home system and the cognitive model, before going into details on how the components have been separated. The solution, which provides great flexibility for each, is discussed before conclusions.

2 Health and Fitness Companion

The Health and Fitness Companion (H&F) is a conversational interface for supporting healthy lifestyle. The companion plans each day together with its user at home, and suggests healthy activities, such as walking to work, when possible. During the day, a mobile interface to the Companion can be used to record various physical activities, such as those walks to work. Afterwards, the user is able to report back to the companion on the day, and get more advice and support. At this point information recorded by the mobile system is automatically used by the system.



- Good morning, anything interesting organized for today?
- I'm going for a walk.
- Is that walk before dinner?
- No, I think I'll walk after I've eaten.
- OK, so you are going for a walk after dinner, is that correct?
- Yes.
- Great, why don't you cycle to work?
- Okay, I can do that.

Figure 1: Health and Fitness Companion Scenario.

As seen in Figure 1, H&F home system uses a Nabaztag/tag WLAN rabbit as a physical interface. Nabaztag provides audio output and push-to-talk input, and is able to move its ears and operate four colored lights to signal, for example, emotions. The mobile interface, as seen in figure 2, runs on a Windows Mobile platform and uses push-to-talk speech input, speech output and a graphical interface with key and stylus input. The graphics include Nabaztag graphics and the same voice as in the home system is used for output to help users associate the two interfaces. The mobile Companion follows the user for physical activities, such as jogging, and collects data on

the exercises and feeds this back into the main system. While it includes a multimodal speech interface, the main input modality for the mobile Companion can be considered to be GPS positioning. It is used to collect information on user's exercise and provide feedback during the exercise. It is also used as the detection for the completion of the exercises, which information is then forwarded to the home system and the cognitive model.

From technical viewpoint, H&F is a multimodal spoken dialogue system containing components for speech recognition (ASR), natural language understanding (NLU), dialogue management (DM), natural language generation (NLG), and speech synthesis (TTS). Furthermore, it includes a separate cognitive model (CM), which works in close co-operation with DM of the home system, as presented in the following sections. The dialogue system in the home system is implemented using Java and Jaspis framework (Turunen et al., 2005) with jNabServer (<http://www.cs.uta.fi/hci/spi/jnabserver/>) for Nabaztag connectivity. The cognitive model is implemented in Lisp and integrated into the Jaspis framework. The mobile interface is implemented in Java with native C++ code for speech technology components. It uses PART (<http://part.sourceforge.net/>) for persistent storage and HECL for scripting in dialogue manager (<http://sourceforge.net/projects/hecl>).



Figure 2: Mobile Companion Interface.

For speech recognition and synthesis, H&F uses Loquendo ASR and TTS. Current recognition grammars for the home system, derived from a WOZ data and extended using user test data, have a vocabulary of 1090 words and a total of 436 grammar rules. Recognition grammars are dynamically selected for each user input, based on the dialogue state. The mobile interface use mobile versions of Loquendo technology. Due to the technological limitations, more challenging acoustic environment, potential physical exhaustion of users, and more restricted domain, the recognition grammars in the mobile interface will remain significantly smaller than those of the home system.

NLU is based on SISR semantic tags (<http://www.w3.org/TR/semantic-interpretation/>) embedded in the recognition grammars. In the home system, where mixed initiative interaction is possible, the tags provide parameters compatible with predicates used to represent information on the dialogue management level. Input parsing unifies these parameters into full predicates based on the current dialogue state. In mobile system, more strict state based dialogue modeling can results in unambiguous output straight from the SISR tags.

Natural language generation is a mixture of canned strings and, in the home system, tree adjoining grammar based generation. In addition, control messages for Nabaztag ears and lights can be generated.

As discussed previously, distribution and coordination of the different tasks between different components can become rather complex in systems such as H&F without proper modeling of interaction, domain, and reasoning components. Next, we present a model which allows flexible interaction between the cognitive model and the dialogue management.

3 Dialogue Management and Cognitive Modeling

There is great consensus that components of a dialogue system can be split into at least three parts: an input module, which receives user input and parses it into a logical form, dialogue management, which maintains and updates dialogue state based on user input and generates output requests, and an output module, which generates natural language output to user based on the requests. In the case of H&F, we have also separated a cognitive model (CM) from dialogue manager (DM), as seen in Figure 3. We call this

module the cognitive model, because it contains what can be considered higher level cognitive processes of the system. Next, we present DM of the home system, CM component, and the mobile interface, focusing on their interaction.

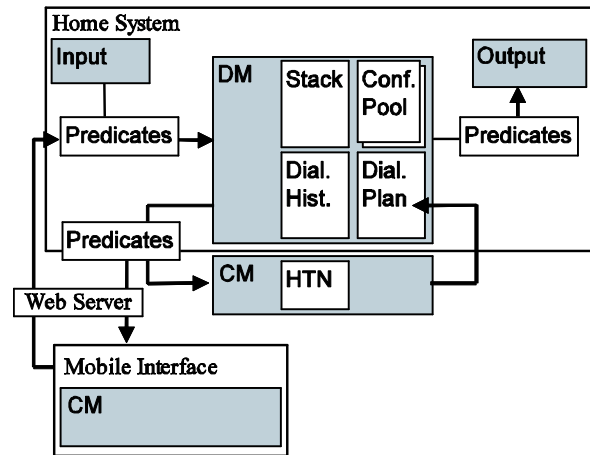


Figure 3: Information passed between the components.

3.1 Cognitive Model Responsibilities

The task of CM is to model the domain, i.e., know what to recommend to the user, what to ask from the user and what kind of feedback to provide. CM in H&F uses hierarchical task networks (HTNs) (Cavazza et al., 2008) as the method of planning healthy daily activity for the user. Part of a network can be seen in Figure 4. In the current H&F implementation, the planning domain included 16 axioms and 111 methods, 49 operators, 42 semantic tags, 113 evaluation rules and there are 17 different topics to be discussed with the user.

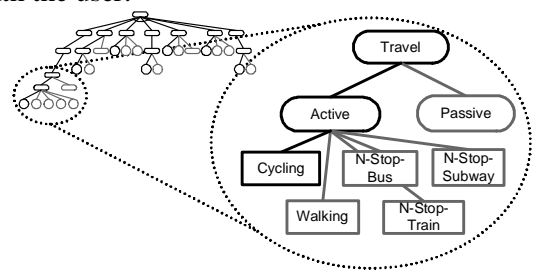


Figure 4: Hierarchical Task Network.

CM is aware of the meaning of the concepts inside the system on a domain specific level. It generates and updates a dialogue plan according to the information received from the user. The plan is forwarded to DM. Interaction level issues are not directly visible to CM.

3.2 Dialogue Management in the Home System

The task of DM is to maintain and update a dialogue state. In the H&F system, the dialogue state includes a dialogue history tree (currently linear), a stack of active dialogue topics, and current user input, including ASR confidence scores and N-best lists. In addition, two pools of items that need to be confirmed are stored; one for items to be confirmed individually and another for those that can be confirmed together in one question.

DM receives user inputs as predicates parsed by the NLU component. If an utterance is successfully parsed and matches the current dialogue plan (see Section 3.3), DM does not need to know what the meaning of the input actually is. It just takes care of confirmations and provides the information to CM. When generating output requests based on the plan, DM can also be unaware of the specific meaning of the plan items. Overall, DM does not need to have the deep domain understanding CM specializes in.

DM, however, is aware of the relations of the predicates on the topics level, i.e., it knows, which predicates belong to each topic. This information is used primarily for parsing input. DM also has understanding of the semantics of the predicates which relates to interaction. Namely, relations such as question – answer pairs (suggestion – agreement, confirmation – acceptance/rejection, etc.) are modeled.

On implementation level, dialogue management is implemented as a collection of separate small dialogue agents, following the principles of the underlying Jaspis architecture. These agents are small software components, each taking care of a specific task and in each dialogue turn one or more agents are selected by DM. In the current H&F prototype, there are over 30 dialogue agents. There is a separate agent for each topic that can occur in the plan. In practice, one topic maps to a single plan item. These agents are all instances of a single class with specific configurations. Each agent handles all situations related to its topic; when the topic is the first item of an active plan, they produce related output and when the user provides input matching to the topic they forward that information back to the cognitive model. In addition, topic specific agents handle explicit topic switch requests from the user (e.g., “let’s talk about lunch”) and also take turn if the topic is found on top of the dialogue topic stack. A topic ends up in the stack

when it has not been finished and a new topic is activated. The other agents found in the system include one that generates a confirmation if the ASR confidence score is too low, one that repeats the last system utterance when the user requests it (“please repeat the last one”), and an agent to handle ASR rejection errors.

3.3 Mobile System

Mobile system is designed mainly to support users’ on their physical exercises and collected data on them from the home system. The mobile system receives the day plan that the user has made with the home system and it is used as basis when users activate the system. This way, the user does not need to re-enter information such as the type of an exercise. This is possible, however, with simple spoken dialogue or by using the graphical user interface. During the exercise, GPS information is used by the system to provide feedback on pace to the user using speech output. For dialogue management, the mobile system uses a state based model, based on scripting. Since the mobile system focuses on the physical exercises, it is aware of the meaning of the predicates it receives on that level. It knows more about running and walking than any other component. At the same time, it ignores most of the day plan it receives. For example, eating related plan items are not relevant to the mobile system in any way and are ignored (however, in the future we could include the possibility to report on meals as well).

3.4 Dialogue Plan and Day Plan

The communication between the dialogue managers and CM is based on a dialogue plan and a day plan. Various kinds of dialogue plans (Larsson et al., 2000, Jullien and Marty, 1989) have been used inside dialogue managers in the past. A plan usually models what the system sees as the optimal route to task completion.

In H&F, CM provides a plan on how the current task (planning a day, reporting on a day) could proceed. The plan consists of items, which are basically expressions on domain specific propositional logic. Example 1 contains two items from the start of a plan for planning the day with the user in the morning. The first plan item (QUERY-PLANNED-ACTIVITY) can be realized as the question “Anything interesting planned for today?” by the system.

As new information becomes available (from the user), it forms a plan for the day or a report of the day. DM provides this information to CM,

piece by piece as it becomes available. At the same time, the information is uploaded into a web server, where the mobile interface can access it anytime.

As CM receives the information, it updates the dialogue plan as necessary. Query type items, whose information has been gathered, disappear from the plan and new items may appear.

The messages sent to CM can add new information (predicates) to CM state. DM can also remove information from CM if previously entered information is found to be untrue. Similarly, information uploaded to the web server for mobile use can be modified. The information includes statements on user's condition (tired), user's commitments to the system (will walk to work), user's preferences (does not like cafeterias) and user's reports on past activity (took a taxi to work), which can be accomplishments or failures of earlier commitments.

```

<plan>
  <plan-name>Generate-Task-
Model-Questions</plan-name>
  <plan-item>
    <action>QUERY-PLANNED-
ACTIVITY</action>
  </plan-item>
  <plan-item>
    <action>SUGGEST-TRAVEL-
METHOD</action>
    <param>CYCLING-
TRAVEL</param>
    <param>HOME</param>
    <param>WORK</param>
  </plan-item>
...

```

Example 1: Start of a plan.

DM in the home system can follow the dialogue plan produced by CM step by step. Each step usually maps to a single question, but can naturally result in a longer dialogue if the user's answer is ambiguous or error management is necessary, or if DM decides to split a single item to multiple questions. For example, the two dialogue turn pairs seen in example 2 are the result of a single plan item (QUERY-PLANNED-ACTIVITY). Since the first user utterance does not result in a complete, unambiguous predicate, DM asks a clarification question. A single user utterance can also result in multiple predicates (e.g., will not take bus, has preference to walking).

When the mobile interface is activated, it downloads the current day plan from the web server and uses it as a basis for the dialogue it has with the user. The exercise which will then take place can be linked to an item in the day plan, or it can be something new. As the exercise is completed (or aborted), information in this is uploaded to the web server. From there the DM of the home system can download it. This information is relevant to the DM when the user is reporting on a day. The home system downloads the information provided by the mobile system and reports it back to CM when the dialogue plan includes a related item. DM may also provide some feedback to the user based on the information. It is noteworthy, that CM does not need to differentiate in any way, whether the information on the exercise came from the mobile system or was gathered in a dialogue with the home system.

```

( <plan-item>
  <action>QUERY-PLANNED-
ACTIVITY</action>
</plan-item>
S: Good morning. Anything in-
teresting organized for today?
U: I'm going jogging.
(<pred>
  <action>PLANNED-
ACTIVITY</action>
  <param>ACTIVITY-
JOGGING</param>
  <param>unknownTime</param>
</pred> )
S: Is that jogging exercise
before dinner?
U: No, it's after.
( <pred>
  <action>PLANNED-
ACTIVITY</action>
  <param>ACTIVITY-
JOGGING</param>
  <param>AFTER-DINNER</param>
</pred> )

```

Example 2: A dialogue fragment and a corresponding plan item and predicates, latter of which is forwarded to the cognitive model and the mobile interface.

Similarly, clarifications and confirmations are not directly visible to CM. DM can confirm items immediately (for example, when low confidence is reported by the NLG component) or it can delay confirmations to generate a single con-

firmation for multiple items at an appropriate moment.

Most importantly, when presenting questions and suggestions to the user, DM is free to choose any item in the plan, or even do something not included in the plan at all. When information from the mobile system is available, it can direct where we start the dialogue from. DM could also decide to do some small-talk to introduce sensitive topics, which can be useful in managing the user-system relationship (Bickmore and Picard, 2005). In the future, we see DM to have various kinds of knowledge on the dialogue topics: it can know how personal these topics are and how topics are related to each other. It may also have some topics of its own. The communication that is not related to the domain does not reach CM at any point.

CM can include additional annotation in the plan. One such example is the importance of the information. If information is marked important, it is likely, but not certain, that DM will explicitly confirm it. It is also possible for CM to explicitly request a confirmation by generating a separate plan item. For example, if a user reports on having run much more than they are likely to be capable of in their condition, CM can generate a confirmation plan item. It is worth noting, that DM cannot do reasoning on such level and therefore CM must participate in error handling in such cases.

3.5 Benefits of the Model

The presented model for interoperability between the mobile system, the DM of the home system and CM has provided great flexibility for each component. While the dialogue plan generated by CM provides a base for dialogue management, which, in most cases, is followed, DM can deviate from it. DM can handle confirmations as it pleases, add small talk, and process the plan items in any order. The model also supports mixed-initiative dialogues; while DM may follow the plan, the user may discuss any topic. In our current implementation, user input is parsed first against the previous system output, next to the current topic, and finally to the entire space of known predicates. If needed, we can also make parsing more detailed by parsing against dialogue history and the current plan. This way, the information produced by CM is used in input parsing. The dialogue plan can be used in dynamic construction of recognition grammars to support this on ASR grammar level.

Most importantly, all this is possible without including domain specific knowledge. All such information is kept exclusive in CM. Similarly, CM does not need to know the interaction level properties of the topics, such as recognition grammars and natural language generation details. These are internal to their specific components. The mobile system uses the same knowledge representation as CM, but CM does not need to be aware of its existence at all. Similarly, the mobile system can use any part of the information it receives, but is not forced to do anything specific. DM just feed all the information to it and lets it decide what to do with it. When the mobile system provides information back to the home system, DM handles this and CM can ignore completely the fact that different parts of the information it receives were generated using different systems. Similarly, the mobile system does not see any of the internals of the home system.

On an implementation level, the model is independent of the mechanics of either DM or CM. DM can be implemented using state transition networks (a network per plan item), forms (form per item), agent based model, like in the case of mobile system, or any other suitable method. Similarly, the plan does not tie CM to any specific implementation.

4 Conclusions

When dialogue systems move beyond limited task based domains and implement multimodal interfaces in pervasive computing environment, their complexity increases rapidly. Dialogue management, which in most cases is handled with well understood methods such as form filling or state transition networks, tends to grow more complex. Therefore, a model to modularize dialogue management and domain reasoning is needed. At the same time, distributed systems required various kinds of information to be communicated with components and systems.

While traditional spoken dialogue systems have been task-based, the Health and Fitness Companions are part of the users' life for a long time, months, or even years. This requires that they are part of life physically, i.e., interactions can take place on mobile setting and in home environment outside of traditional, task-based computing devices. With the physical presence of the interface agent and spoken, conversational dialogue we aim at building social, emotional relationships between the users and the Compan-

ion. Such relationships should help us in motivating the users towards healthier lifestyle. The mobility of the interface integrates the system into the physical activities they aim at supporting users in.

We have presented a model, which separates cognitive modeling from dialogue management and enables flexible interoperability between the two and also enables sharing the gathered knowledge to the mobile part of the system and back. This division, while similar to separation of a back-end from dialogue management, draws the line deeper into the area of interaction management. The cognitive model processes domain level information and generates dialogue plans. The dialogue manager focuses only on interaction level phenomena, such as initiative and error management, and other meta-communication. In order to enable flexible interaction, the plan provides a potential structure for the dialogue, but the dialogue manager is free to handle things in different order, and even add new topics. It can also include input from a mobile interface of the system without making this explicit to the cognitive model. One example of flexibility is error management; while the actual error correction is the task of the dialogue manager, domain level knowledge can reveal errors. Using the dialogue plan, the cognitive model can provide such information to the dialogue manager without knowledge on details of error management. The model also enables user initiative topic shifts, management of user-system relationship and other novel issues relevant in domain-oriented dialogue systems.

Overall, the model presented has enabled a clear division and interoperability of the different components handling separate parts of the interaction. The presented model has been implemented in the Health and Fitness Companion prototype, and it has enabled the cognitive model, the dialogue manager, and the mobile interface to be developed in parallel by different groups using various programming languages an integrated system.

5 Acknowledgements

This work is part of the EU-funded COMPANIONS-project (IST-34434). The Cognitive Model has been developed by University of Teesside, UK, while the mobile interface has been implemented in Swedish Institute of Computer Science.

References

- Dybkjaer, L., Bernsen, N. O., Minker, W., Evaluation and usability of multimodal spoken language dialogue systems, *Speech Communication*, 43, 1-2, June 2004, pp. 33-54.
- Wilks, Y., Is There Progress on Talking Sensibly to Machines?, *Science*, 9 Nov 2007.
- Marti, S. and Schmandt, C. Physical embodiments for mobile communication agents. *Proceedings of the 18th annual ACM symposium on User interface software and technology*: 231 – 240, 2005.
- Kainulainen, A., Turunen, M., Hakulinen, J., Salonen, E.-P., Prusi, P., and Helin, L. A Speech-based and Auditory Ubiquitous Office Environment. *Proceedings of 10th International Conference on Speech and Computer (SPECOM 2005)*: 231-234, 2005.
- Jönsson, A. A Natural Language Shell and Tools for Customizing the Dialogue in Natural Language Interfaces, *Research Report, LiTH-IDA-R-91-10*, 1991.
- Salonen, E.-P., Hartikainen, M., Turunen, M., Hakulinen J., Funk, J. A. Flexible Dialogue Management Using Distributed and Dynamic Dialogue Control. *Proceedings of ICSLP 2004*. pp. 197-200.
- O'Neill, I. Hanna, P. Liu, X., McTear, M., The Queen's Communicator: An Object-Oriented Dialogue Manager, *Eurospeech 2003, Geneva, Switzerland (2003)*, pp. 593–596.
- Pellom, B. Ward, W. Pradhan, S., The CU Communicator: An Architecture for Dialogue Systems, *Proceedings of ICSLP 2000, Beijing China, November 2000*.
- Turunen, M., Hakulinen, J., Räihä, K.-J., Salonen, E.-P., Kainulainen, A., and Prusi, P. An architecture and applications for speech-based accessibility systems. *IBM Systems Journal*, Vol. 44, No 3, 2005, pp. 485-504.
- Cavazza, M., Smith, C., Charlton, D., Zhang, L., Turunen, M. and Hakulinen, J., A 'Companion' ECA with Planning and Activity Modelling, *Proceedings of AAMAS08, 2008 (to appear)*.
- Larsson, S. Ljunglöf, P. Cooper, R. Engdahl, E., Ericsson. S. GoDiS - an accommodating dialogue system. *ANLP / NAACL '00 Workshop on Conversational Systems*, May 2000.
- Jullien C., Marty, J.-C. Plan revision in person-machine dialogue, *Proceedings of ACL'89, Manchester, England, April 1989*, pp.153-160.
- Bickmore, T. W., Picard, R. W. Establishing and maintaining long-term human-computer relationships. *ACM Trans. Computer-Human Interaction* Vol. 12, No. 2. (June 2005), pp. 293-327.