ISCA Archive
http://www.isca-speech.org/archive

ITRW on Multi-Modal Dialogue
in Mobile Environments
Kloster Irsee, Germany
June 17-19, 2002

THE PLANNING COMPONENT OF AN INTELLIGENT HUMAN MACHINE INTERFACE IN CHANGING ENVIRONMENTS

Sunna Torge, Stefan Rapp, Ralf Kompe
Sony International (Europe) GmbH
Advanced Technology Center Stuttgart, Germany
torge@sony.de, rapp@sony.de, kompe@sony.de

Abstract In a network environment, no matter if Internet or personal area network, many devices, applications, and services, which can be added and removed dynamically, are accessible by the user. An intelligent human interface to such an environment needs to adapt to changes in the network. This does not only concern input, output or understanding parts but also the knowledge about functionalities provided by the network. In this paper we describe, how to enhance a multi-modal dialogue system with a planning module, consisting of a reasoning component and abstract models describing the functionalities of the devices, applications, and services which are part of the network. This additional component allows to support plug&play and to serve complex user wishes according to the current available functionalities in the network. The work described in this paper was successfully integrated in the prototype system SmartKom.

Keywords: Planning, network environment, complex user wishes

Introduction

The fact that mobile environments require adaptive human machine interfaces is widely recognized. However, adaptivity is not only concerning input/output components or the understanding part of a system. Adaptivity also concerns the knowledge about functionalities of provided devices, applications, and services according to the current environment. Given e.g. a network environment many devices, applications, and services are accessible by the user. In many cases, they can be added and removed dynamically (plug&play), supported by appropriate hardware and protocols, e.g. bluetooth, IEEE 1394 or USB. An intelligent human machine interface to such an environment might be a mobile device which allows the user to ask for services provided by the network from everywhere or to control the local infrastructure. It needs to support the following aspects: First, it needs to adapt to newly added or removed devices, applications, or services (plug&play), i.e. it needs to know about the functionalities of each device, application, or service, which is currently part of the network. Secondly, the human machine interface should allow the user to ask for complex wishes involving several devices and/or applications and/or services in the network.

A problem with many of the user interfaces is that the user has to think in terms of devices, applications, and services (play, record, ...). Instead, the user should be able to interact with a system in a natural way, as he would do with a human assistant. To give an example, the user might simply say "please, record the film XYZ on Saturday". In consequence, the system should deal with the devices, applications, and services for him. It should use an EPG (electronic program guide) to find out the channel, starting time and duration of the film. With this knowledge the system then should program a video cassette recorder (VCR). The user in this case has a complex wish involving several devices and several actions per device. He wants simply his wish to be solved; he does not want to care about the individual devices or the action sequences to be carried out. Furthermore the user does not want to care about the available devices. Instead the system should realize the user's wish depending on the current network. Note, that this sort of tasks can only be solved with a central human machine interface operating several devices.

Given a home network or personal area network, the problem we address in this paper is to enable a flexible and intuitive control of an ensemble of devices, applications, and services with a single multi-modal dialogue system. We focus on

- how complex user wishes can be served, and
- how plug&play can be realized on the human machine interface side.

Traditional dialogue systems [v. Kuppevelt et al., 2000] used for the control of devices, applications, and services usually consist of an input recognition part, an input understanding part, a dialogue manager, and the devices, which are to be controlled. The simplest way to control the devices is to have a unique mapping of the user input to the appropriate control command. Given e.g. a speech input "CD play" it uniquely can be mapped to the "play"-*command* of a CD player. However, this approach does not allow to serve complex wishes like the above mentioned examples, since the user input for complex wishes cannot be uniquely mapped to a single control command, but rather to a sequence of actions possibly involving several devices.

In order to serve complex wishes in [Larsson et al., 2001] the use of pre-calculated plans for complex tasks is proposed. With this approach however, the user is limited to these tasks, where pre-calculated plans are available. In changing environments, where availability of devices, applications, and services might change very often this approach is not adequate. Every potential constellation of devices, applications, and services must be covered by pre-calculated plans, which might be a problem.

The realization of plug & play on the human machine interface side, i.e. to gain more flexibility w.r.t. adding new devices, applications, and services requires several features of a system. Firstly, the recognition, analysis, and rendering components must be dynamically configurable, see e.g. [Rapp et al., 2000]. Secondly, in order to keep the dialogue manager independent of applications it is necessary to separate domain dependent knowledge from domain independent knowledge [Flycht-Eriksson, 2000, Thompson and Bliss, 2000, Han and Wang, 2000, Lemon et al.,]. Furthermore, the system needs to know about the functionalities of newly added devices, applications, and services, and how to deal with them.

Separation of domain-dependent and -independent knowledge can be obtained by introducing an additional module serving as an interface between the dialogue manager and the devices to be controlled, including models of the devices [Pouteau and Arevalo, 1998]. However, to serve complex user wishes it is not enough to consider models of single devices. Instead, some sort of reasoning on the provided services is necessary.

In order to solve the problems concerning the separation of domain dependent and domain independent knowledge as well as the interplay between devices, applications, and services and the dialogue system, a multi-modal dialogue system is enhanced with a planning component. This planning component consists of a reasoning component and a set of abstract models describing the functionalities of devices, applications, and services available in the network. The abstract descriptions of the functionalities of the devices, applications, and services may be part of the devices, applications, and services themselves, thus will be added to the system by plugging in.

Henceforth, we use the term "devices" instead of "devices, applications, and services" for reasons of brevity. The paper is organized as follows: In the next three section the planning module is described in detail. A small section about a first prototype implementation follows. Some remarks on future work will conclude the paper.

1. The Planning Module: Overview

As described above our focus is on serving complex user requests which may involve several devices. In order to handle requests like “please, record the film XYZ on Saturday” the system needs to find out

- *which* devices are necessary to serve the request,
- *how* to control the devices.

For this purpose a planning component and appropriate data structures for planning are needed.

We propose to introduce a new module, the so called planning module, in order to solve the above described problems. The purpose of the planning module is to allow the integration of the functionalities of several different devices and to support plug&play. In a traditional dialogue system the devices which are to be controlled by the system, are controlled by the dialogue manager (see figure 1). In contrast the enhanced system (see figure 2) provides a sort of intelligent interface between the dialogue manager and the devices to be controlled, namely the planning module. The planning module consists of

- an abstract model of the functionalities of a device (so called functional model) for each device in the network,
- a reasoning component,

Furthermore, the planning module is always aware of the current state of each device in the network.

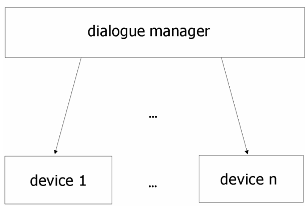

Figure 1. Traditional dialogue system

Instead of controlling the devices directly from the dialogue manager, this allows to formulate the requests given by the dialogue manager on an abstract level. Based on this abstract request, the planning module first calculates a plan to serve the request and then performs the plan.

The dialogue manager is therefore independent of the real devices, and robust against changes of them. The overall functionality of the

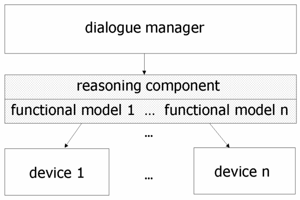

Figure 2. Dialogue system with planning module

given devices does not need to be known in the dialogue manager but it is deduced from the functional models of the given devices. This approach enables the system to serve complex user requests, and be flexible and robust against changes in the network of controlled devices, i.e. it supports plug&play.

In the following, the different parts of the planning module, i.e. the functional model and the reasoning component are described in detail.

2. The Functional Model

The functional model is separated into an external model and an internal model. This separation reduces the search space, since the external model describes the knowledge necessary to find appropriate devices whereas the internal model describes the single device in more detail.

2.1 External Model

The external model of a device describes the input data and the output data. As a very simple example let us consider an EPG database, that allows the extraction of certain EPG information by appropriate queries. From the viewpoint of a dialogue system an EPG database is a device where a request about the TV program can be sent to and a result will be returned (see figure 3). This knowledge is necessary but also sufficient to decide whether the device "EPG database" is appropriate to serve a given user request or not.

The external model also allows to draw conclusions on which devices are to be combined in order to serve one complex request. As a simple example let us consider the request "I'd like to see channel 4". This complex wish "I'd like to see channel 4" can be described as the need of a high frequency signal providing "channel 4" and the need of a device providing a viewable picture of "channel 4". Assume the given network consists of a tuner and a display. A tuner is a device which receives a high frequency signal and provides a video signal, whereas a display provides a viewable picture if it receives a video signal. In figure 4 the external models of a tuner and a display are depicted. Thus the knowledge given by the external models of the tuner and the display is sufficient to infer that these two devices need to be combined in order to serve the request. In the same way the system can infer that a tuner needs to be combined with a display in order to watch TV or that a camcorder can be combined with a VCR.

Additional problems occur, if there are several devices with the same functionalities in the network, e.g. several tuners, several displays, an internet EPG and a DVB EPG. The concept of the external model allows to draw conclusions about which devices are capable to serve a given request. This might be several different constellations of devices. However, the decision about which device is appropriate for the current request needs to be made separately and might be based on a user profile, costs or other heuristics. How to solve these sort of problems is part of the future work.

Note, that there might be several external models describing one device. As an example consider the external model of the tuner (see figure 4). Depending on the channel which is broadcasted the external model of the tuner differs from each other.

Figure 3. External model of an EPG database

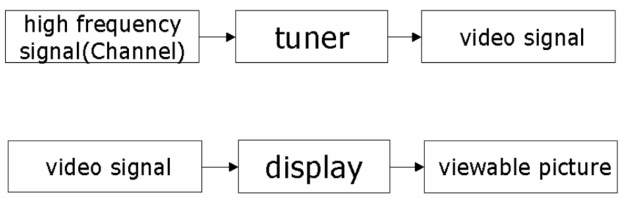

Figure 4. External model of a tuner and a display

2.2 Internal Model

The internal model of a device describes the possible states of a device and the actions which are necessary to bring the device from one state into another. Regarding the knowledge to be modeled, i.e. states and actions, finite state machines (FSM) ([McCarthy and Hayes, 1969, Moses and Tennenholtz, 1995, Henzinger, 1996]) are an appropriate means to built up an internal model. The states in the internal model are partly annotated with incoming or outgoing data respectively. This is an important point since these annotations ensure the connection between the external and the internal model of a device. In addition, the states also might be annotated with pre- or postconditions of each state. Pre- and postconditions of a state refer to states of other devices. They are also formulated as in- and outgoing data. Thus they refer to external models. They only refer to external models in order to allow for independence of the internal models. The internal model of the database is depicted in figure 5. A more complex example is depicted in figure 6, which shows the internal model of the tape drive of a VCR. Some of the states like e.g. *recording, playback* are annotated with in- and outgoing data. These annotations establish the connection between states and external models. The states *recording* and *playback* are additionally annotated with preconditions. The precondition of e.g. the state *recording* refers to the external model of, say, a tuner. Thus, the reasoning component can assure that before recording a film or the like with a tape drive of a VCR, a connected tuner must be set on the right channel. Note, that in this example the preconditions of the state *recording* directly refer to a tuner, i.e. a tuner is required to provide the video signal. This is a simplification, since a video signal also can be provided by e.g. a camcorder. Therefore the preconditions for the state *recording* can be reduced to $[Ingoingdata, videosignal]$, where $Ingoingdata$ is considered as a variable. Thus, from the modeling point of view the system is free to use a tuner, a camcorder or other devices, which provide the appropriate video signal.

3. The Reasoning Component

Each of the devices of the given network is modeled by its functional model. In order to use this knowledge, the planning module in addition consists of a reasoning component. This is necessary, since given a complex user request and given a network of devices, the following steps need to be executed in order to serve the user request:

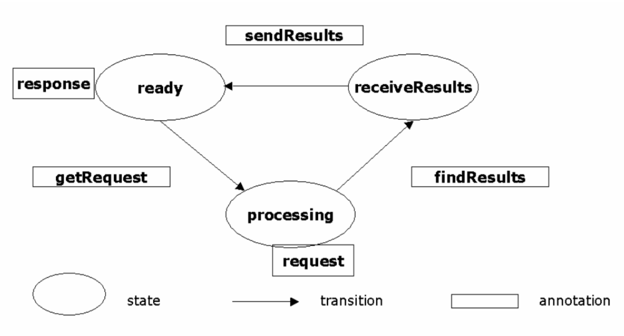

Figure 5. Internal model of a database

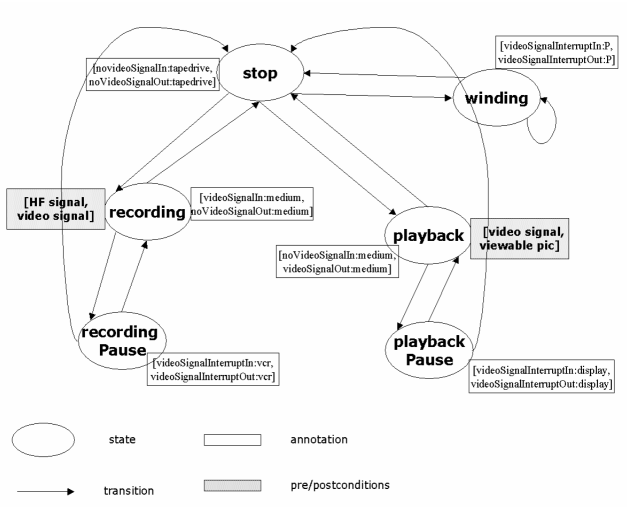

Figure 6. Internal model of a tape drive

1 Find out the device/devices which are necessary to perform the request.

2 Find out how to combine and to control the devices, i.e. generate an appropriate plan of actions to be executed.

3 Execute the necessary actions.

Using the functional model of each device in the given network the reasoning component first searches for the necessary devices and then generates a plan of actions for every device, which is involved. The plan finally will be executed.

The functional model and the reasoning component allow to model how devices need to be combined. This is done via the input and the output data of each device. Furthermore it is possible to describe a temporal order in which the devices are to be controlled. This is done by formulating pre- and postconditions of single states of a device.

Let us have a closer look to the algorithms. In a first step a complex task like e.g. "I'd like to see channel 4" is mapped to a tuple of in- and outgoing data. Given that, a device search algorithm either returns a single external model of a device or an ordered set of external models, where each external model uniquely corresponds to one device.

Given the information about the necessary device/devices a planning algorithm returns a sequence of actions, i.e. a plan. Recall, that the in- and outgoing data in the external model also occur in the internal model of the device. This link allows to identify an initial and a final state in the internal model of a device. Given these states and given the current state of a device, which is always known to the planning module, a path in the FSM is searched for. This path corresponds to a sequence of actions, since the transitions in the internal model are annotated with actions.

While searching for a path (i.e. a plan), pre- and postconditions of the states are checked. Informally spoken, the pre- and postconditions model the necessity of using further devices and how to use them. Since pre- and postconditions are formulated like the external models, the same algorithms are applied to fulfill them, which return additional plans. These additional plans, corresponding to other devices, which are necessary to fulfill the given complex task, finally are integrated in the overall plan.

This procedure is performed for every device which is returned by the device search algorithm. Since the device search algorithm returns an ordered sequence of external models the order of the execution of the generated plans is also fixed.

It should be pointed out that the concept described above allows two possibilities to model the necessity of using several devices. First, it can be modeled by mapping the complex task to in- and outgoing data, such that the device search algorithm returns an ordered set of external models (and also devices) as described above. Secondly, the concept of pre- and postconditions allows to model a more flexible temporal order.

4. Realization in a Prototype

The work described in this paper is applied in the SmartKom project [Wahlster et al., 2001]. In SmartKom, a multi-modal dialogue system is being developed. A first prototype implementation was shown at Eurospeech 2001 and at the international MTI Conference, Saarbruecken 2001. This successfully uses the reasoning component as well as the function models of tourist information database, navigation system, TV, VCR tape drive, tuner, electronic program guide (EPG), microphone, loudspeaker, address book, biometrics, e-mail service, telephone, and document camera. Some of the user wishes the system is able to handle, involve the usage of several devices. In particular some devices are used to serve different complex tasks at different times and therefore need to be configured according to the current usage. Making a telephone call includes e.g. to borough the loudspeaker and the microphone from the dialogue system and to use it temporarily for the telephone call. This configuration is triggered by the planning module. Furthermore the planning module is always aware of the current configuration and thus the current usage of the devices. Another example of a complex user wish is to send a scanned document as an attachment via email. In order to serve this complex wish, the planning module needs to trigger a format conversion of the scanned document according to the requirements of the email service.

5. Conclusion and Future Work

The work described in this paper shows how to enhance a multi-modal dialogue system with a planning module, consisting of a reasoning component and abstract models describing the functionalities of the devices, applications, and services which are part of a network. This enhancement is motivated by the fact, that a network environment, no matter if Internet or personal area network, where many devices, applications, and services, are accessible by the user and can be added and removed dynamically requires an intelligent human interface which adapts to changes in the network. This does not only concern input, output or understanding parts but also the knowledge about functionalities provided by the network. Furthermore, the user should be allowed to ask for complex wishes, which may involve the usage of several devices. Thus a central human machine interface is necessary.

The planning module, described in this paper, supports both, plug& play as well as serving complex user wishes. It includes a functional model of each device in the network and a reasoning component, which were described to some detail. The first prototype system of SmartKom, including the planning module, was also briefly mentioned.

Future work will include refinement of the reasoning component concerning efficiency and assessment of generated plans. Assessment will be based on costs, which are given through annotations of actions.

Other desirable features of the reasoning component are the generation of conditional plans and exception handling. Instead of choosing one single plan the planning module also could propose several alternative plans to the user or base the decision on heuristics, given a user profile.

In order to realize plug&play the functional model of a device needs to be integrated in the device itself or on an internet server. This is simulated right now in the SmartKom system.

Another issue is to consider complexity. In [Domshlak and Dinitz, 2001] complexity results are shown for a multi-agent system. The models used in this work are comparable to the functional model. Thus the complexity results may also apply to the work described in this paper.

6. Acknowledgment

This research was conducted within the SmartKom project and partly funded by the German Federal Ministry of Education and Research under grant 01IL905I7. The authors would like to thank Horst Rapp, Axel Horndasch, and Martin Emele for discussions about the ongoing work.

References

[Domshlak and Dinitz, 2001] Domshlak, C. and Dinitz, Y. (2001). Multi-agent off-line coordination: Structure and complexity. In *Proceedings of 6th European Conference on Planning, 2001.*

[Flycht-Eriksson, 2000] Flycht-Eriksson, A. (2000). A domain knowledge manager for dialogue systems. In *Proc. of ECAI 2000*, Berlin.

[Han and Wang, 2000] Han, J. and Wang, Y. (2000). Dialogue management based on a hierarchical task structure. In *Proc. of ICSLP 2000*, Bejing, China.

[Henzinger, 1996] Henzinger, T. (1996). The theory of hybrid automata. In *Proc. of the 11th Annual Symposium on Logic in Computer Science.*

[Larsson et al., 2001] Larsson, S., Cooper, R., and Ericsson, S. (2001). menu2dialog. In *Proc. of Workshop Knowledge and Reasoning in Practical Dialogue Systems at IJCAI 2001*, Seattle.

[Lemon et al.,] Lemon, O. et al. The witas multi-modal dialogue system 1. citeseer.nj.nec.com/lemon01witas.html.

[McCarthy and Hayes, 1969] McCarthy, J. and Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence.*

[Moses and Tennenholtz, 1995] Moses, Y. and Tennenholtz, M. (1995). Multi-entity models. *Machine Intelligence*, 14:63–88.

[Pouteau and Arevalo, 1998] Pouteau, X. and Arevalo, L. (1998). Robust spoken dialogue system for consumer products: A concrete application. In *Proc. of ICSLP 1998*, Sydney, Australia.

[Rapp et al., 2000] Rapp, S., Torge, S., Goronzy, S., and Kompe, R. (2000). Dynamic speech interfaces. In *Proc. of Workshop AIMS at ECAI 2000*, Berlin.

[Thompson and Bliss, 2000] Thompson, W. and Bliss, H. (2000). A declarative framework for building compositional dialog modules. In *Proc. of ICSLP 2000*, Bejing, China.

[v. Kuppevelt et al., 2000] v. Kuppevelt, J., Heid, U., and Kamp, H., editors (2000). *Best Practice in Spoken Language Dialogue Systems Engineering*, volume 6 of *Journal of Natural Language Engineering*. Cambridge University Press.

[Wahlster et al., 2001] Wahlster, W., Reithinger, N., and Blocher, A. (2001). Smartkom: Multimodal communication with a life-like character. In *Proc. of Eurospeech 2001*, Aalborg.