Companion-Technology: Towards User- and Situation-Adaptive Functionality of Technical Systems

Frank Honold*, Pascal Bercher†, Felix Richter†, Florian Nothdurft‡, Thomas Geier†, Roland Barth*, Thilo Hörnle†, Felix Schüssel*, Stephan Reuter§, Matthias Rau†, Gregor Bertrand‡, Bastian Seegebarth†, Peter Kurzok†, Bernd Schattenberg†, Wolfgang Minker‡, Michael Weber*, Susanne Biundo†

**Institute of Media Informatics*
†Institute of Artificial Intelligence
‡Institute of Communications Engineering
§Institute of Measurement, Control and Microtechnology

Ulm University, Germany
{firstname.lastname}@uni-ulm.de

***Abstract*—The properties of multimodality, individuality, adaptability, availability, cooperativeness and trustworthiness are at the focus of the investigation of Companion Systems. In this article, we describe the involved key components of such a system and the way they interact with each other. Along with the article comes a video, in which we demonstrate a fully functional prototypical implementation and explain the involved scientific contributions in a simplified manner. The realized technology considers the entire situation of the user and the environment in current and past states. The gained knowledge reflects the context of use and serves as basis for decision-making in the presented adaptive system.**

***Keywords*-Adaptive HCI; AI Planning; Dialog Management; Interaction Management; Companion Technology**

I. Introduction

In the future, many technical systems will be Companion Systems – competent assistants, which provide their functionality in a completely individualized way: they adapt to the user's capabilities, preferences, requirements, and current needs and take into account both the emotional state and the situation of the individual user [1]. For instance, the user does not have to learn how to operate a system, instead, the Companion Technology enables the system to adapt itself to distinct users.

To illustrate our technology, we present a movie[1] in which we demonstrate a prototypical Companion System. The referenced movie shows two users, who have the task to install their home theater. They have varied levels of knowledge and thus, the Companion System behaves accordingly. In Fig. 2, a scene from the scenario is shown, where the user stands in front of the home theater system that has to be wired up. The user interacts with the Companion System via the user interface and a set of sensors. A Companion System is a knowledge-based system and comprises planning and plan execution, the dialog management, and the interaction management.

II. Architecture

We present a distributed interactive system, which components are able to interact using a common middleware concept, based on the SEMAINE API [2]. The system's main components (cf. Fig. 1) communicate with each other using XML.

Our model of human computer interaction – and thus the realized architecture – distinguishes between application, tasks, and the user interface itself. We employ the Cameleon Reference Framework, which specifies four levels of abstraction [3]. The different abstraction levels can be directly mapped to dedicated components in the presented architecture. The first level of "Tasks and Concepts" is handled by the depicted component for *Planning and Plan Execution* [4] (cf. Fig. 1). The subsequent "Abstract User Interface" (AUI) level is individually generated by the *Dialog Management* (DM). The AUI acts as the modality-independent model of the UI. Based on that, the *Interaction Management* (IM) transfers the "Concrete User Interface" (CUI) on the next level. The interplay of DM and IM is implemented as motivated in [5]. The IM's output is realized by diverse *User Interface* components, which are able to use different widgets to realize the so-called "Final User Interface" (FUI).

Inspired by [6], the presented system is able to utilize multiple modalities for input and output. Furthermore, the knowledge-based system is able to use AI-task planning to operate in an adaptive manner, even on the abstract task level. Different sensors allow to characterize the context of use, which encompasses user, task, and environment.

III. Knowledge Base

To provide individual user support, a Companion System must maintain possible uncertain knowledge of different kinds. We use Dynamic Markov Logic Networks to represent the current belief of the system as it evolves over time [8].

Within the knowledge base, different models are combined with each other, such as the user model (including

[1] Link to the video: `http://companion.informatik.uni-ulm.de/ie2014/companion-system.mp4` [2014-04-15]

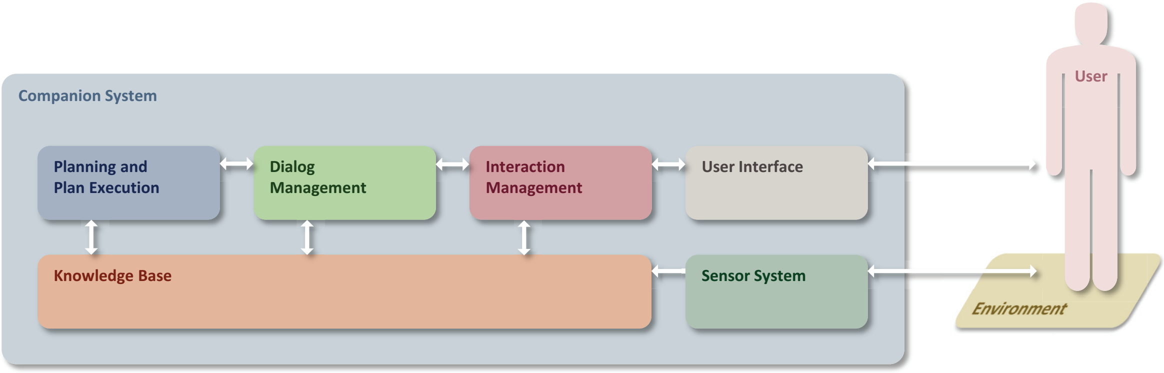

Figure 1. Architectural overview of the presented knowledge-based system according to [7]. The system is able to react and adapt to implicit and explicit actions of the user as well as to environmental changes in the context of use.

aspects like preferences, handicaps, knowledge, and emotional state), the planning model (with a description of the available actions, as well as the user's goal), and a model of the environment (including models of the available user interface components). Due to this integration, it is possible to transparently use information across different modules [9]. An example is the exploitation of planning information to improve user localization [10], which is then used to improve the selection of output devices [7].

IV. AI Planning - Plan Generation

The Companion System provides user support on the basis of automatically generated plans of action. We employ a hybrid planning approach that combines causality-based planning and hierarchical planning [11], [12]. Hybrid plans map a complex – or *abstract* – task to a sequence of *primitive tasks*, which – executed in their order by the user – will solve the abstract task. In our scenario, the goal of connecting the home theater will be transformed into a sequence of more differentiated tasks, like connecting two devices.

Hybrid planning relies on broad background knowledge, which includes the user's goal and a formal description of the user's environment. The planning process starts by generating an initial plan, which consists of the current (initial) state and an initial abstract task representing the user's goal. This initial plan is referred to as planning problem.

In order to come up with a solution, the initial plan needs to be refined. This means to decompose abstract tasks into successions of more specific tasks, which are predefined solution schemata. In general, for the same abstract task there might exist several solution schemata. In our example, an audio connection may be realized either via analog or digital interfaces. The planning system always chooses the most suitable solution scheme by taking into account the user's personal preferences. The decomposition of abstract tasks is repeated until a plan is obtained that merely consists of primitive tasks, which can be directly executed by the user. They are called *actions*.

In our example, putting a specific plug into a specific socket would be such an action. At this point, the system needs to ensure that the derived plan is actually executable. To this end, the planning component analyzes causal dependencies between the actions, which are represented by so-called *causal links*.

Where the planning component determines that the conditions for executing an action are not met, additional actions will be inserted to ensure a consistent causal structure. The resulting plan is a solution to the initial planning problem. Actions in the solution plan are only partially ordered, so, a most suitable total order is generated and passed to the plan execution component. It is then checked whether the pending action can be executed. Then, a valid plan for the initial planning problem is available and the first action of the sequence of tasks can be passed on to the dialog management.

V. Dialog Management

The dialog management [13]–[15] controls content and flow of the interaction between user and system. This means, it translates each action into a user-adaptive dialog. For each possible action, (or according to [3]: its task description), there is a dialog model available consisting of multiple dialog steps which each achieve parts of the desired effects of the action. Though several dialog steps may realize the same desired effects, they might require specific user characteristics to be tackled.

The sequence of dialog goals to achieve all desired action effects is therefore adapted to each individual user. This means, in order to improve each user's individual interaction experience, the dialog management decides, which of the next dialog steps is most suitable. For that, information coming from the knowledge base will be taken into account, namely user emotion, or user knowledge.

In other words, the dialog content and flow may change, if two different users interact with the system. For example, if the first user has expert knowledge on a topic, the sequence of dialog steps would include simplified instructions. Contrary to that, a novice user would receive a more detailed and longer sequence of dialog steps with additional extent of assistance.

Before each dialog step is passed on to the interaction management, the explanation manager [16] additionally checks whether the user has the required knowledge. If necessary, an additional dialog step explaining the missing knowledge is included in the course of the dialog. Each dialog step's abstract user interface (AUI) description is then passed on to the interaction management for further processing.

In the next section we explain how the passed AUI is presented in a situation- and user-adaptive way by the interaction management and the linked user interface components.

VI. Interaction Management

The interaction management (IM) controls which user interface components are offered for interaction. It determines an adequate combination of diverse user interface components, adapted to the identified context of use. In addition, it is able to combine user inputs from different modalities, such as pointing gestures in conjunction with verbal cues.

As a first step, the dialog management's modality-independent output (the AUI) is split up into its single abstract information fragments. As the user situation may change, it must be analyzed, which user interface components are currently available. For each information fragment that can be communicated, all possible variants of representation have to be identified. Each such variant of representation is judged by diverse evaluation functions, taking into account the knowledge base's current user- and environment model. The IM's evaluation process identifies the most adequate variant of representation for rendering via a so-called fission process [7]. These single output fragments are re-assembled and form the concrete user interface (CUI). This CUI description is passed on to the involved user interface components for rendering, as well as to the knowledge base. The knowledge base can use this data to optimize the user localization. Each addressed user interface component renders an individually assigned part of the output as final user interface (see Fig. 2).

The evaluation process judges the output after any change in the context-of-use to ensure a proper user interface adaptation. In the given scenario, the context-of-use is updated about every 500 ms. These permanent updates enable the interaction management to react instantly to changes of the user's position by changing the output device components accordingly.

As user input, in addition to uni-modal input via speech or touch, the user can also perform cross-modal inputs via pointing gestures in conjunction with speech. The system fuses any incoming input fragments to derive the user's intended meaning [17]. This interpretation is then passed on to the dialog management as the user's input (e. g. a selection). A more detailed description is given in [18].

VII. Adaptation and Explanation

As described earlier, adaptation to changes in the context of use can be applied on the levels of the abstract, concrete and final user interfaces. The CUI as well as the FUI are instantly adapted in cause of the fission's modality arbitration process within the interaction management. For instance, the UI is rendered on another display. The AUI can be adapted by the adaptive dialog management with the effect of an altered dialog behavior.

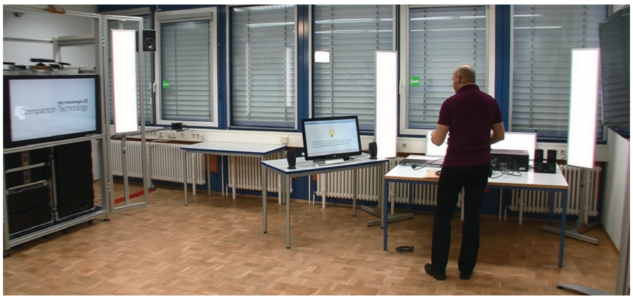

Figure 2. Based on the user's position, the user interface is rendered on the screen in front of the user.

Next, we will describe how a Companion System is able to adapt on the level of tasks and concepts using plan repair, since unforeseen changes in the context of use may also affect the pre-planned sequence of tasks.

At the end of this section we describe how the presented Companion System responds to possible user demands for explanation. In the presented example the user addresses such a question to the level of tasks and concepts.

A. Plan Repair

If interaction results do not match the system's expectations, the current plan must be repaired by the planning component. It does so by searching for an alternative solution, taking into account previously executed actions [12]. If the execution of an action fails, the plan execution component identifies that action and annotates it in the currently executed plan. While the original initial plan also serves as new initial plan, the planning process also uses the already executed plan to find a new solution.

Any repaired solution must be obtained by refining the initial plan as it is done with ordinary plan generation. In case of repairing, we additionally need to ensure that the actions which are already executed are also part of any new solution.

Plan repair starts by decomposing the initial plan. Because of the execution failure, choosing different solution schemata or inserting other primitive tasks might be required. To obtain an executable plan, additional actions might have to be inserted, even if they were already executed. The repaired plan is then linearized, taking into account also the previously executed actions, and passed on to the plan execution component for further processing, as described earlier.

B. Plan Explanation

If the user wants to get justifications for why a particular action is needed, the plan explanation component becomes active. The plan execution component passes on the necessary knowledge, such as the plan being currently executed and the action of interest, to the plan explanation component. It generates a formal explanation for why that action is necessary to achieve the user’s goal, based on the decomposition hierarchy and the causal links of the plan [19]. The explanation in its raw form is a proof in an axiomatic system resulting from a formal representation of the given plan in terms of first-order formulas. The explanation proof is then adapted by taking user knowledge into account. The adapted explanation is passed on to the dialog management, which converts it into text using natural language generation. This abstract output is forwarded to the interaction management, which subsequently causes specific user interface components to present the explanation to the user. With the user’s information needs satisfied, he can now confidently proceed in his intended way.

VIII. Conclusion

Our example has shown, how a Companion System can individually support multiple users with different skills, and adapt to their situation continuously. By the use of Companion Technology, similar systems can be developed for a variety of applications and fields. Companion Systems will fundamentally change the way we deal with technical equipment in the future.

IX. Acknowledgments

This work was supported by the Transregional Collaborative Research Center SFB/TRR 62 “Companion-Technology for Cognitive Technical Systems”, which is funded by the German Research Foundation (DFG).

References

[1] A. Wendemuth and S. Biundo, “A companion technology for cognitive technical systems,” in *Proceedings of the EUCogII-SSPNET-COST2102 International Conference (2011)*, ser. LNCS Proceedings on Cognitive Behavioural Systems, A. Esposito, A. Vinciarelli, R. Hoffman, and V. C. Müller, Eds., Dresden, 2012.

[2] M. Schröder, “The semaine api towards a standards-based framework for building emotion-oriented systems,” *Adv. in Hum.-Comp. Int.*, vol. 2010, p. 21, January 2010.

[3] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt, “A unifying reference framework for multi-target user interfaces,” *Interacting with Computers*, vol. 15, no. 3, pp. 289–308, 2003.

[4] P. Bercher, S. Biundo, T. Geier, T. Hoernle, F. Nothdurft, F. Richter, and B. Schattenberg, “Plan, repair, execute, explain – how planning helps to assemble your home theater,” in *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS 2014)*, 2014.

[5] B. Dumas, D. Lalanne, and S. Oviatt, “Multimodal interfaces: A survey of principles, models and frameworks,” in *Human Machine Interaction – Research Results of the MMI Program*, ser. LNCS. Berlin: Springer, March 2009, vol. 5440/2009, ch. 1, pp. 3–26.

[6] W. Wahlster, *SmartKom: Foundations of Multimodal Dialogue Systems*, ser. Cognitive Technologies Series, J. te Vrugt, V. Zeissler, and W. Wahlster, Eds. Heidelberg: Springer, 2006.

[7] F. Honold, F. Schüssel, and M. Weber, “Adaptive probabilistic fission for multimodal systems,” in *Proceedings of the 24th Australian Computer-Human Interaction Conference*, ser. OzCHI ’12. New York, NY, USA: ACM, 2012, pp. 222–231.

[8] T. Geier and S. Biundo, “Approximate online inference for dynamic markov logic networks,” in *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, nov. 2011, pp. 764–768.

[9] F. Honold, F. Schüssel, M. Weber, F. Nothdurft, G. Bertrand, and W. Minker, “Context models for adaptive dialogs and multimodal interaction,” in *Proc. of the 9th Int. Conf. on Intelligent Environments – IE’13*. Athens, Greece: IEEE, July 2013, pp. 57–64.

[10] T. Geier, S. Reuter, K. Dietmayer, and S. Biundo, “Goal-based person tracking using a first-order probabilistic model,” in *Proceedings of the Ninth UAI Bayesian Modeling Applications Workshop (UAI-AW 2012)*, 8 2012, pp. 18–25.

[11] S. Biundo and B. Schattenberg, “From abstract crisis to concrete relief (a preliminary report on combining state abstraction and HTN planning),” in *Proc. of the 6th European Conf. on Planning (ECP 2001)*, 2001, pp. 157–168.

[12] S. Biundo, P. Bercher, T. Geier, F. Müller, and B. Schattenberg, “Advanced user assistance based on AI planning,” *Cognitive Systems Research*, vol. 12, no. 3-4, pp. 219–236, 2011, special Issue on Complex Cognition.

[13] G. Bertrand, F. Nothdurft, F. Honold, and F. Schüssel, “Caligraphi – creation of adaptive dialogues using a graphical interface,” in *35th Annual IEEE Computer Software and Applications Conference (COMPSAC 2011)*, ser. COMPSAC, Munic, June 2011, pp. 393–400.

[14] M. Horchani, L. Nigay, and F. Panaget, “A platform for output dialogic strategies in natural multimodal dialogue systems,” in *IUI ’07: Proc. of the 12th Int. Conf. on Intelligent User Interfaces*. New York, USA: ACM, 2007, pp. 206–215.

[15] F. Nothdurft, G. Bertrand, T. Heinroth, and W. Minker, “GEEDI - Guards for Emotional and Explanatory DIalogues,” in *6th Int. Conf. on Intelligent Environments (IE’10)*. IEEE Computer Society, 2010, pp. 90–95.

[16] F. Nothdurft, G. Bertrand, H. Lang, and W. Minker, “Adaptive explanation architecture for maintaining human-computer trust,” in *36th Annual IEEE Computer Software and Applications Conference*, ser. COMPSAC, Jun. 2012, pp. 176 – 184.

[17] F. Schüssel, F. Honold, and M. Weber, “Using the transferable belief model for multimodal input fusion in companion systems,” in *Multimodal Pattern Recognition of Social Signals in HCI*, ser. LNCS, vol. 7742. Springer, 2013, pp. 100–115.

[18] F. Honold, F. Schüssel, and M. Weber, “The Automated Interplay of Multimodal Fission and Fusion in Adaptive HCI,” in *IE’14: Proceedings of the 10th International Conference on Intelligent Environments*. Shanghai, China: IEEE, July 2014.

[19] B. Seegebarth, F. Müller, B. Schattenberg, and S. Biundo, “Making hybrid plans more clear to human users – a formal approach for generating sound explanations,” in *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS 2012)*. AAAI Press, 6 2012, pp. 225–233.