Using Multimodal Resources for Explanation Approaches in Technical Systems

Florian Nothdurft, Wolfgang Minker

Ulm University, Institute of Communications Engineering
89081 Ulm, Germany
florian.nothdurft@uni-ulm.de, wolfgang.minker@uni-ulm.de

Abstract

In this work we show that there is a need of using multimodal resources during human-computer interaction (HCI) in intelligent systems. We propose that not only creating multimodal output for the user is important, but to take multimodal input resources into account for the decision when and how to interact. Especially the use of multimodal input resources for the decision when and how to provide assistance in HCI is important. The use of assistive functionalities like providing adaptive explanations to keep the user motivated and cooperative is more than a side-effect and demands a closer look. In this paper we introduce our approach on how to use multimodal input ressources in an adaptive and generic explanation pipeline. We do not only concentrate on using explanations as a way to manage user knowledge, but to maintain the cooperativeness, trust and motivation of the user to continue a healthy and well-structured HCI.

Keywords: Knowledge based systems, Cooperative systems, Adaptive systems

1. Introduction

Assisting the user during HCI is a mandatory functionality in state-of-the-art technical systems. Pre-defined help texts or illustrations and step-by-step instructions are the backbone of assistive functionalities in user-friendly technical systems. In most cases help can be requested by the user in the like of pressing a help-button or changing to a help-dedicated part of the technical system (e.g., frequently asked questions (FAQ) or a tutorial section). Additionally help functionalities are often embedded implicitly in the interaction by describing interaction elements or providing exemplary use-cases. The goals of these functionalities are imparting declarative or procedural knowledge to the user.

However, intelligent technical systems should not be limited to manage user knowledge, but be able to anticipate the users' needs and adapt the running HCI to the user's current situation, requirements or capabilities in an appropriate way. When looking closer at the area of help functionalities one can see that they are all a sort of explanation. Explanations are a typical intervention in human-human interaction (HHI) to clarify concepts, tasks or instructions. Douglas Walton defined successful explanation as a transfer of understanding in a dialogue system in which a questioner and a respondent take part (Walton, 2004). However, explanations can on the one hand influence more than only the knowledge model of the user and on the other hand are not limited to providing conceptual or tutorial explanations as mentioned before.

The first thing that comes into mind when thinking about explanations are pure declarative explanations. For example, explaining the concept, purpose and the looks of an HDMI cable. However, apart from these obvious explanations an intelligent system has to be able to provide explanations on its own system behaviour in order to justify decisions and keep them transparent for the user. In order to derive required explanation capabilities for intelligent systems, we need to examine which problems a user will typically face during usage and which explanations will be necessary to overcome these problems. These problems and matching explanations for intelligent technical systems are inherited from expected questions that might occur in human discourse. Analogously to human discourse, providing reasons for certain system decisions may convince the user of proposed actions and solutions or already made changes. In HHI the person which provides the explanations has usually specific intentions and implicit goals in mind which should be achieved. This is the same for any explanation provided by an intelligent technical system. Sørmo and Cassens (2004) list the different goals of explanations (see table 1. for more details) which correspond in parts to the goals pursued in HHI when providing explanations. Intelligent technical systems should be able to pursue all of these goals of explanation.

Goals	Details
Transparency	How was the answer reached?
Justification	Why is the answer a good answer?
Relevance	Why is it a relevant answer?
Conceptualisation	Clarify the meaning of concepts
Learning	Learn something about the domain

Table 1: The goals of explanation

Applied for the use in intelligent technical systems this means that the system should on the one hand be able to recognize situations in which explanation capabalities are needed and on the other hand chose the appropriate goal of explanation in the appropriate point of time. In order to make these decisions we think that there is more to consider than only the current task and the related user knowledge. Several factors can influence the relationship between user and technical system and therefore make the difference between interaction success and failure.

One important information resource during the interaction in our developed system architecture is the trust model of the user (see fig. 1). Trust has shown to be a crucial point in keeping the user motivated and cooperative. The users'

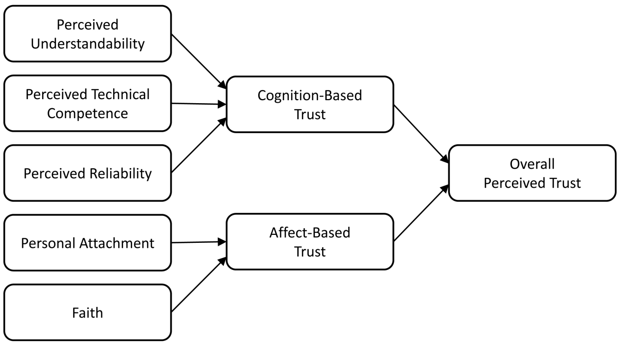

Figure 1: In this for human-computer trust constructed model, personal attachment and faith build the bases for affect-based trust and perceived understandability, perceived technical competence and perceived reliability for cognition-based trust.

trust in the system will be decreased if he does not understand system actions or instructions (Muir, 1992). This can lead to a change in the willingness to interact or in the worst case scenario to an abort in interaction and use (Parasuraman and Riley, 1997). However, providing explanations can help to prevent the decrease of trust (Glass et al., 2008).

The knowledge base contains several information resources, which can be helpful to decide when the integration of an explanation is appropriate. For example, the user's knowledge model serves as an indicator whether to augment the HCI with system-initiated help functionalities. This means that it can indicate when an explanation is necessary in order to qualify the user for an upcoming task execution. Or the knowledge model can help to adapt instructions given by the system to the profile of the user. However, the decision on the point of time when to interfere or adapt the interaction cannot solely be based on one information resource. Information input resources can be typical user input via speech, text, touch or system input sources cumulated in a knowledge base. This knowledge base can contain resources like the user model (e.g., users' knowledge, trust, emotion model) or the environmental model of the sourroundings.

2. Related Work

There are several fields of research involved in developing assistive support for the user in human-computer interaction. For example intelligent tutoring systems (Anderson et al., 1985), which try to impart knowledge on a specific topic to a learning person in the best possible way. However, intelligent tutoring systems are limited to a specific topic and do not explain their behaviour. Expert systems (Jackson, 1990) provide solutions for a given problem or analyze situations based on expert knowledge in order to support and justify user decisions. These mentioned systems concentrate on imparting knowledge to the user. In our work we propose a more multifunctional view on using explanations to not only fulfil the user's needs, but to focus on maintaining the willingness of the user to cooperate.
In the following section we describe the architecture of our explanation pipeline and how we interfere in the right point of time with a system-initiated interaction based on different input resources.

3. Architecture

The underlying architecture depictured in figure 2 shows the basic components which are necessary to receive multimodal input resources in a technical system. Sensors are necessary to receive input data like the user's facial expression or gestures to infer user emotions. Devices are used for user-input as well as system-output. The knowledge base manages the provided sensed or recorded data and reasons about them to provide coherent information.

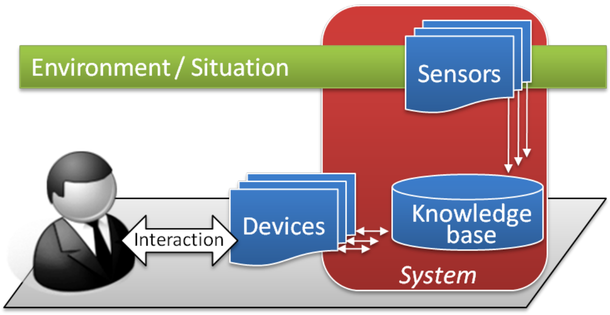

Figure 2: Overview of the basic components of the underlying technical system

Figure 3 gives an overview of the necessary modules for processing the different input ressources and providing appropriate explanations. In our system exist three possibilities to initiate an explanation. The first one is that the user can explicitly request an explanation (e.g., via speech or touch interaction). The second possibility are by the system initiated explanations. The decision of the system to interfere in the HCI is based on information contained in the knowledge base (e.g., user knowledge and trust model). The last one is HCI-intervention by pre-defined points in the dialogue where explanations are initiated automatically by the system. However, these explanations are implemented by the designer and are therefore not really system-initiated explanations. If we take a look at the differences between these three options regarding the goal of the emerging explanation, we can notice that for option one and three the explanation goal is already inherent. This is due to the fact that for user-initiated explanations the user pursues a specific goal. In the case of spoken dialogues this goal has to be inferred by a semantic analysis. For pre-defined points in dialogue the goal of the explanation is already included as well and the goal of the explanation was determined by the designer. However, for by the system-initiated explanations the decision when and what to explain is far more complex. Therefore, in the remainder of this paper, we will concentrate on system-initiated explanations.

4. System-Initated Explanations

Basically there are three decision to make in order to initiate and generate a system-initiated explanation (see figure 4). Mapped to our explanation pipeline, when to explain is matter of the knowledge base or the module monitoring it.

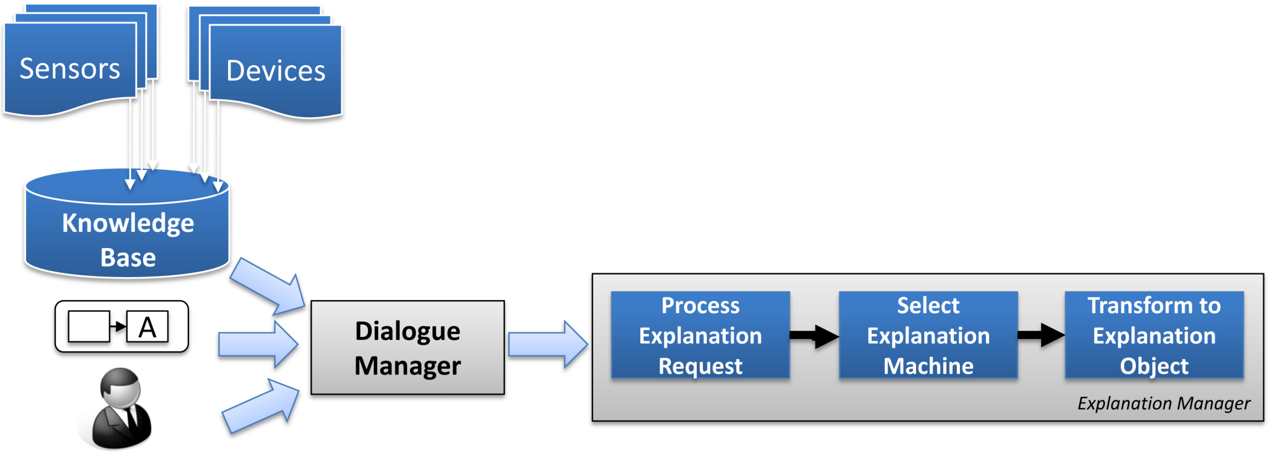
Sensors
Devices
Knowledge Base
A
Dialogue Manager
Process Explanation Request
Select Explanation Machine
Transform to Explanation Object
Explanation Manager

Figure 3: Explanation Architecture: The Dialogue Manager processes the user, system or pre-defined explanation requests. These are forwarded to the Explanation Manager. The Knowledge Base provides all necessary information resources for an appropriate explanation adaptation.

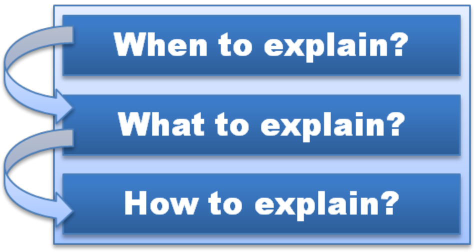

Figure 4: The three basic decisions to make in a system-initiated explanation pipeline

The decision what to explain is the generation of the explanation request and the decision how to explain it, is the selection of an appropriate explanation machine. These steps will be described in more detail in the next section. As mentioned before one of the basic input resources is the knowledge model of the user. However, compared to typically used knowledge models based on knowledge levels (e.g., novice, intermediate, expert) we use a more fine-grained knowledge model infered from past interaction. For this we are keeping track of the HCI in a so-called dialogue history. This history records the decisions and actions of user and technical system. The history is used to note when for example, something was explained to the user, when the user executed a task or requested help from the system. This enables the knowledge base to infer a fine-grained knowledge model of the user. Contained is for example not only which task the user did execute, but which entities were used for this task and if the task completion was successful. In this knowledge model we distinguish between declarative knowledge and procedural knowledge. Declarative knowledge can be used to describe the being of things (i.e., appearance, purpose). Possessing declarative knowledge about something does not necessarily mean to be able to use this knowledge for a task or action. In comparison to that procedural knowledge can be applied to a task. Procedural knowledge provides the knowledge on how to execute a task or on how to solve a problem.

During runtime, the system can check if the knowledge of an upcoming task or action is sufficient and tailor the system prompt appropriatly. For example, if the user does posses declarative knowledge about the concept of HDMI cabels, but lacks procedural knowledge about connecting devices using a HDMI cabel, the explanation will contain a tutorial explanation about how to connect devices, but no additional conceptual explanation about HDMI.

However, the user's knowledge is not the only input resource to decide wheather to explain. As unexpected or not understandable system actions or decisions may influence the trust of the system negatively and indirectly the use of the system, we need to avoid such situations. Therefore we are using the dialogue history to monitor the interaction not only knowledge-wise, but to record system actions and explanations to determine if specific situations have occured before. If this is not the case, system decisions can be augmented by transparency and justification explanations to describe and provide reasons for the system's behavior to the user. After the decision on if to explain, the next step is to select an appropriate type of explanation.

5. Explanation Generation Pipeline

Due to several information resources and contained uncertainties the Explanation Manager has to choose the type of explanation goal (e.g., justification, transparency, relevance) which is most likely the best in the current situation. However, any type of explanation can be a combination of several explanation goals as well. The goals of explanation depend on the results of the request to the knowledge base. For example, if the decision never occured before, the goal of transparency will be added. And if a concept included in this decision is not known to the user, the goal of conceptualisation has to be considered as well.

In the next step, a so called explanation machine defines which parts the explanation should consist of, in order to construct the explanation for the user (see listing 1).

Explanation machines are part of content planning or determination because they select the appropriate ingredients for the explanation. Additionally the explanation machines are modality-independent. The fission decides in the last moment in which modality and on which device (e.g., PDA, Smartphone, LCD-Monitor . . .) the information should be presented. This means that explanation machines should not possess any modality-dependent information. Listing 1 shows a typical explanation machine which then has to be processed further in order to send it back to the dialogue manager to be included into the running dialogue.

```
<?xml version=”1.0” encoding=”UTF–8”?>
<dialoguemodel:DialogueModel xmi:version=”
    2.0” xmlns:dialoguemodel=”dialoguemodel
    ”>
  <explanationMachine initiative=”system”
      inputChannel= knowledge base >
  <content type=”task”>connect_via_HDMI</
      content>
    <type>Justification</type>
    <ingredients>
    <part>user knowledge model</part>
    <part>reasoning trace</part>
    </ingredients>
  </explanationMachine>
</dialoguemodel:DialogueModel>
```

Listing 1: An exemplary explanation machine

5.1. Explanation machine selection

In order to select an appropriate explanation machine, we need to select the most appropriate one for a given event. However, information resources from the knowledge base and the decisions resulting from it have always a certain degree of uncertainty to it. This could be due to several contained data changing at the same moment (e.g., knowledge and trust) or due to the initiative itself containing uncertainty. In the first case the uncertainty is in the decision, which defined explanation machine is the best for knowledge and trust loss together and in the second case it is due to the uncertainty in the input channel itself (e.g., in speech recognition the misrecognition of a word).

However, the most common case is the change in only one aspect of contained data. For example, when it comes to selecting the appropriate explanation machine for specific trust issues, we need to determine which bases of trust need improvement. The idea is that some extended explanation machines perform better than others for selected bases of trust. For example, when coping with a possibly low perceived understandability, which relates to the mental model of the user and predicting the future system behaviour, an explanation machine imparting transparency is appropriate. Understanding how the system reached an answer will help the user predict similar behaviour in future events.

An explanation machine is generated for the current running dialogue and meant to be arranged before or after the next action in the dialogue flow. For example, a dialogue action “take the bus and not the taxi” is extended by an appropriate explanation (e.g., “You cannot take a taxi, because you do not have enough money right now. And my information shows that the next ATM is 3 miles away”.). However, the abstract information contained in the explanation machine needs to be converted to real text or an image or a multimodal explanation for a specific interaction like this first.

5.2. Explanation machine transformation

The first step to a detailed and well phrased explanation is the transformation of the explanation machine into an explanation dialogue goal, which can be integrated into the running dialogue by the Dialogue Manager.

```
<?xml version=”1.0” encoding=”UTF–8”?>
<dialoguemodel:DialogueModel xmi:version=”
    2.0” xmlns:dialoguemodel=”dialoguemodel
    ”>
  <explanationObject>
    <content type=”task”>connect_via_HDMI
    </content>
    <type>Justification</type>
    <ingredients>
    <part name=”user knowledge”>
     intermediate(connect_via_HDMI), expert(
        HDMI)
    </part>
    <part name=”reasoning trace”>
     setup_HomeTheater(TV, Receiver,
        Speakerset), connect_TV_Receiver(TV
        , Receiver)
    </part>
    </ingredients>
  </explanationObject>
</dialoguemodel:DialogueModel>
```

Listing 2: An exemplary explanation object after the second step

For this task basically two steps are required. The ingredients of the explanation machine have to be refined to concrete content fragments (see figure 2). In this listing we can see that the user is amongst other things an expert in connecting devices, but a novice in the concept of HDMI. The explanation object notes that the system knowledge about how to connect TV and receiver should be included into the explanation. Additionally his disposition, meaning if he is in a positive or negative mood, decreased by 0.5 points since the last check (on a scale from -1 to +1). During the second step these fragments have to be selected, deleted or adapted to the current user and his current situation. For example, if some minutes ago the system explained to the user that it will not present information by speech because there are too many persons inside the room, the current explanation does not need to mention that fact again and the part about the number of persons in the room can be deleted. This generated explanation dialogue has to be processed by the fission component which decides on the modalities and complexity of the explanation. This depends on the available devices and the situation the user currently is in.

5.3. Beautifications

In our architecture the final modality-dependent explanations are generated by so-called beautifications. Beautifications are for example natural language generation pipelines for generating real text, which can be presented in a graphical or speech user interface.

Text generation is done by using a typical natural language generation (NLG) pipeline (Reiter and Dale, 2000). The first step in a NLG-pipeline, namely the content determination is realized by the explanation machine. The selected content needs to be arranged and structured in the next step, followed by micro planning, consisting of lexicalisation, aggregation and referring expressions generation. Lexicalisation decides which specific words should be used to express the given abstract content. Aggregation decides which content should be put together in sentences or paragraphs and the referring expressions generation decides how the given entities should be referred to. The last step in the natural language generation is the linguistic realisation of the structured content, which can be included as explanation utterance in the dialogue. However, the generation of text from a given explanation machine is only an example of how the final explanation may be realised.

6. Conclusion

In this paper we introduced our ideas and architectural designs on why multimodal input resources are important for a successful interaction of user and system. Our focus to maintain a healthy HCI is on providing user-adaptive explanations. The system has to be able to take the initiative in adaptation of dialogues and assist the user in a way that keeps him cooperative. Multimodal input resources can be used in several stages of an adaptive explanation pipeline. They can help to decide upon the point of time to augment the interaction with an explanation and help with the decision which goal of explanation is the most appropriate one. We explained how the concept of human-computer trust can help to improve HCI. Providing explanations not solely based on the user knowledge model can help to maintain the user cooperative and perceive the system as trustworthy.

7. Acknowledgements

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

8. References

JR Anderson, CF Boyle, and BJ Reiser. 1985. Intelligent tutoring systems. *Science*, 228:456–462.

Alyssa Glass, Deborah L. McGuinness, and Michael Wolverton. 2008. Toward establishing trust in adaptive agents. In *IUI '08: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 227–236, New York, NY, USA. ACM.

Peter Jackson. 1990. *Introduction to Expert Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.

B M Muir. 1992. Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems. In *Ergonomics*, pages 1905–1922.

Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 39(2):230–253, June.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge University Press, New York, NY, USA.

F. Sørmo and J. Cassens. 2004. Explanation goals in case-based reasoning. In *Proceedings of the ECCBR 2004 Workshops*.

Douglas Walton. 2004. A new dialectical theory of explanation. In *Philosophical Explorations*, pages 71–89.