International Journal of Engineering Sciences Paradigms and Researches:
(Vol. 27, Issue 02) and (Publishing Date: January 2016)

Adaptive HCI Multi-Modal Fission and Fusion

Jyostnarani Tripathy[1], Sudhansu Bisoyi[2], Jagannath Ray[3]

[1,3]Associate Professor, Department of Computer Science Engineering,
Gandhi Institute For Technology (GIFT), Bhubaneswar
[2]Assistant Professor, Department of Computer Science Engineering,
Gandhi Engineering College, Bhubaneswar

Publishing Date: January 16, 2016

Abstract

Present context-aware systems gather a lot of information to maximize their functionality but they predominantly use rather static ways to communicate. This paper motivates two components that serve as mediators between arbitrary components for multimodal fission and fusion, aiming to improve communication skills. Along with an exemplary selection scenario we describe the architecture for an automatic cooperation of fusion and fission in a model driven realization. We describe how the approach supports user-initiative dialog requests as well as user-nominated UI configuration. Despite that, we show how multimodal input conflicts can be solved using a shortcut in the commonly used human-computer interaction loop (HCI loop).

Keywords: *HCI; multimodal; multi-modal; fission; fusion; interaction management; companion technology.*

I. Introduction

Interaction and communication are important aspects of human life. Humans manage to use different modalities and are able to adapt their way of communication to different contexts of use. Looking at current trends in HCI research we find different approaches aiming to provide user interfaces for multimodal interaction. Some of them focus on adaptive output generation, others focus on recognition, combination, and semantic understanding of user inputs from different modalities. So it takes both, an adaptive output mechanism as well as a flexible concept of user input understanding to realize technical systems, which are able to meet the high standard of human communication and interaction skills.

Present architectures for multimodal systems contain specialized fission and fusion components to process system outputs and user inputs. Within the HCI loop these two components realize the *interaction management*. Their linking component is commonly realized as a dialog management component (cf. [1]). Until now, these two interaction-specific components (fusion and fission) act rather isolated. This article demonstrate how an interplay of these two components can help to increase interaction opportunities.

If an interactive multimodal system is aware of its syntactic output, a more general fusion approach can be used to automatically handle diverse multimodal inputs. Without such knowledge, fusion has to be realized in a more or less hard-coded way. The other way around, the fusion can deduce user demands concerning the system's way of output rendering. For example, a user might say: "*[Show|Tell|Give]* me more information about *that*", while performing a pointing gesture on an object displayed on the screen. Such individual and user-initiated nominations for different channels or devices affect the fission's reasoning on modality arbitration for information representation at runtime.

The architectural interplay in combination with a model driven realization offers a third possibility: a shortcut in the HCI loop, from the fusion directly to the fission, bypassing the dialog manager (DM). If the fusion detects ambiguous inputs, the fusion component can use the fission's capabilities to resolve possible input conflicts. So UI- and interaction-specific conflict resolution can be performed within the interaction management without depending on the DM's functionality.

II. Background

In recent years, there have been a number of approaches considering both, fission and fusion [1]–[5]. While [1], [4] focus on the fusion aspects and treat fission as a relatively simple response planning, others explicitly regard the complexity introduced by a fission component [2], [3], [5]. Their presented architectures focus on the adaption of the interface where fission and fusion are realized as two isolated components. The interplay of fission and fusion seems to be application-specific and mainly hard-coded. The idea of "No Presentation without Representation" [5] motivates the use of a common internal representation, which is accessible for different components. We are convinced that such a concept can also be used in a fully model-driven approach, where outputs based on an abstract, and modality-independent dialog model (cf. [6]) are refined by the interaction management to form an individual, and user-specific output at runtime. However, little is known about the possibilities that arise using a direct interplay of fission and fusion in a comprehensive, application-independent, and model-driven approach.

Multimodal Fission: According to [7], systems, which combine different output modalities like text and speech evolved since the early nineties. The allocation of output

modalities of these early multimodal systems was rather hard-coded than based on intelligent algorithms. To summarize the findings of [8] and [9], the main tasks in fission are concerned with the following four questions: (1) What is the information to present? (2) Which modalities should be used to present this information? (3) How to present the information using these modalities? (4) and Then, how to handle the evolution of the resulting presentation? An important survey on multimodal interfaces, principles, models, and frameworks is provided by [1]. Beyond that, [1] mentions the idea of machine learning approaches for multimodal interaction. The given example focuses on machine learning in multimodal fusion on the feature level; but such techniques may also be appropriate for fission approaches. Another interesting approach is presented in [10]. The authors present a multi-agent system, where past interactions are taken into account to reason about the new output. They recommend a machine learning approach for case based reasoning.

To reason about the best UI configuration in a certain Context of Use (CoU) is a challenging task. Some approaches provide meta UIs where the user can specify a certain UI configuration, e.g. via an additional touch device (cf. [11]). Based on that, the system is able to respect the user's demands and can distribute the UI via the referenced device components. In our current approach for modality arbitration [12] we use real-world data, afflicted with uncertainty, to perform a continuous UI adaptation that respects the ongoing changes in the CoU. Based on our investigations rule-based approaches can be seen as established practice. Recent work goes together with model-driven UI generation to realize adaptive UIs. Additional user input can be used to support the system's decision process.

Multimodal Fusion: Fusion of sensory information can be performed at different abstraction levels, namely feature, decision, and hybrid level fusion [1], [13]. In the domain of HCI, approaches usually apply decision or hybrid level fusion on incoming events caused by sensors for different modalities. Each approach can be realized in a frame-based, unification-based, or in a statistical manner, as described by [4]. From early frame-based approaches [14], [15], today's systems have evolved to unification approaches based on typed feature structures like [16], [17] or on rules like [18], [19]. These systems are particularly good at handling complex multimodal utterances and specific time synchronicity of events. Statistical approaches that use statistical processing techniques to exploit recognition probabilities [20] promise an increase in robustness. Our current approach [21] applies evidential reasoning as a generalization of probabilities to provide robust fusion results. Most of these mentioned approaches use a fixed set of possible interactions specific for the type of application the approach was intended for. In [21] a very flexible, but fixed abstraction of interaction events on which the reasoning is performed was used. The latest revision can utilize arbitrary interaction models that suit the domain at hand using an abstract graph notation based on GraphML [22]. This allows specifying the semantics of fusion of arbitrary interactions via XML.

We can identify three aspects, where a direct interplay of fission and fusion can be advantageous or even essential. (1) When using a fission component that realizes an adaptive UI in a model driven way, the fusion component should be provided with all relevant information about the resulting generic UI. As a result, input understanding can be tailored to the currently possible inputs and variable properties (e.g. positions of objects on the screen) can be considered. (2) User-initiative demands for specific ways of information presentation (e.g. something should be presented in a particular modality), which are identified by the fusion component should directly influence the fission's reasoning process. This way, such demands can directly be considered, whenever possible. (3) Ambiguities or other conflicting user inputs can be detected by the fusion component. In such situations the interaction management shall directly respond to the user and ask for clarification. Fusion and fission can handle the disambiguation at runtime in cooperation in a clarifying interaction with the user. This process can be handled completely within the *interaction management*, without depending on the dialog management's functionality.

III. Contributions

The main contribution of this paper is an exploration into the possible connections of multimodal fission and fusion and how this can lead to an increased usability. We find that (1) an architecture with interfaces linking fission and fusion allows both components to benefit from each other. We demonstrate, how fission and fusion can act together to form an adaptive, and model driven interactive multimodal system, which allows functionality beyond the state of the art. As part of the architecture, we introduce and emphasize two components, (2) the content manager (CM) and (3) the nomination manager (NM). We explain, how the CM generates an abstract interaction model (AIM) at runtime to provide the fusion with all relevant information to tailor input understanding to the currently realized UI. The NM assists the fission and allows respecting user-initiative UI specifications at runtime. Beyond that (4) a bypass to the dialog manager is presented, which can help to resolve ambiguous inputs in an automated manner within the interaction management. The prototypical implementation of these concepts as well as cutting the HCI loop short shows that the presence of such functionality brings useful addition to existing behavior. The three concepts CM, NM, and bypass realize the interplay of fusion and fission.

IV. Concept and Realization

This section starts with the description of a short scenario that serves as an elucidating example throughout the rest of the article. Next we explain the architecture of our prototypical implementation that realizes a common HCI loop. After that, we focus on the interplay of fission and fusion, and describe how the aforementioned aspects are realized using two new components as well as a shortcut in the HCI loop.

Scenario: Imagine the situation in which a random person shall setup a home cinema system. The user's challenging task is to wire-up all the different devices. The system's task is to advise and explain alternative connections between different devices and to support the user in the decision on how to connect them. In the scenario we look at a situation in which the system has to realize a selection dialog that offers the user a choice between three different cables. The examples used in the following sections deal with different user reactions. One time the user just selects a cable. Another time additional information about a specific cable is requested together with a demand on how it should be presented. Yet another time, the user performs an unclear input because he mixes up the cable types. At this point we want to emphasize the pure model-driven approach, and that the presented approach is not limited to the given scenario.

A. Architecture

Based on the scenario, we describe our architecture (see Fig. 1) and the major components that are involved in the HCI loop. Our architecture is based on findings from [1]. We extended their architecture with the *nomination manager*, the *content manager*, and the possibility to realize a *dialog management bypass*. The components are connected via different topics using a message oriented middleware [23].

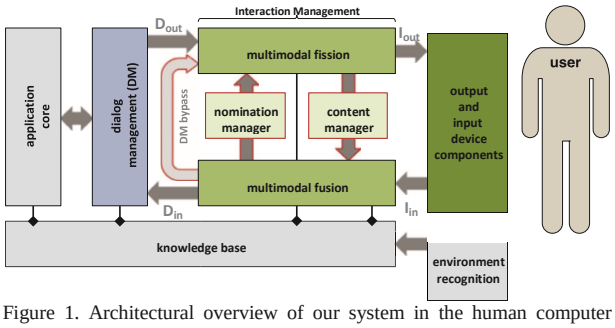

Figure 1. Architectural overview of our system in the human computer interaction loop including three new aspects (nomination manager, content manager, and the dialog management bypass).

Our system's dialog management initiates the HCI loop as described in the scenario with a dialog output D_{out}. This abstract output consists of a modality-independent selection offer of three cables (see Listing 1). The interaction management's fission component is in charge to infer a modality-specific user interface description based on the abstract description of the selection offer in the received D_{out}. The fission's resulting interaction output I_{out} (see Listing 2) is then passed to the involved device components that render the user interface. The system can utilize displays of different sizes at different locations in this scene as well as use speakers for text-to-speech synthesis (TTS). Now the UI is ready to accept diverse user inputs (cf. Fig. 5).

In the scenario the fission reasons a UI where the user is free to use speech or pointing gestures and touch interaction as explicit user inputs. Beyond that, the system's prototypical implementation is also able process implicit inputs (cf. [24]) like recognized user disposition and user location shifts to adapt the UI.

Once the input device components have sensed and interpreted any explicit user inputs up to a certain decision level of abstraction, they provide the fusion component with their modality-specific interaction input I_{in}. So far

```
<?xml version="1.0" encoding="utf-8"?>
<dialogOutput dialogID="cable_selection">
  <topic>
    <abstractInformation objectID="topic" informationID="cable_selection_topic"/>
  </topic>
  <dialogAct>
    <selection objectID="cable_selection_container"
               informationID="cable_selection_prompt">
      <abstractInformation objectID="cinch" informationID="cinch_information"/>
<abstractInformation objectID="scart" informationID="scart_information"/>
      <abstractInformation objectID="hdmi" informationID="hdmi_information"/>
    </selection>
  </dialogAct>
</dialogOutput>
```

Listing 1. An exemplary abstract and modality-independent dialog output. The output contains a topic and a selection (one of three items) including a selection prompt. The dialog's control flow is influenced by the object IDs. The interaction's information flow is defined by the information IDs.

the diverse input components work independently and isolated from each other. After combining these diverse inputs by the fusion component, the most probable event is passed as an abstract and modality-independent dialog input D_{in} to the dialog management. This marks the end of the current HCI cycle. The HCI loop can continue with another dialog output.

In the application scenario for example, the user can perform a pointing gesture on the HDMI cable visualized on the screen, while saying "this one" as a deictic reference. In this case, two input device components would raise I_{in} events. One event is raised by the gesture recognition as reference to the HDMI cable. The other input event is raised from the Automatic Speech Recognition (ASR) component because of the verbal "this one" trigger. The fusion of these events then results in a dialog input D_{in} with the selection of the HDMI cable, which is sent to the dialog management.

Detached from the given scenario the presented approach is able to meet the different requirements of a dynamic working domain in an intelligent environment as there are: changing user models, a dynamic surroundings model, different and fluctuating device models[1] which allow or require different interaction concepts.

The following two paragraphs within this sub-section provide insights into the work of the dialog management component and into one of the possible input device components – the component for Automatic Speech Recognition (ASR).

Dialog Management: The dialog management component serves as link between a system's application logic and the user interface. In our scenario we use a hierarchical dialog model, where complex tasks can be communicated via sequences of individual dialog acts [25]. The acts result from user-specific decompositions of the hierarchical dialog model based on the course of the dialog.

[1] Devices may be present or become non-available in a certain period of time. In addition the user-to-device-distance may vary.

Within the hierarchy they are structured using guards and effects as pre- and post-conditions.

In the remainder of this article we describe a situation where a user requests additional information on the HDMI cable: "*Show me* more information about the HDMI cable." In this situation a modality-independent information request **request_hdmi_explanation** is sent as part of a dialog input D_{in} to the dialog manager. The guard for the HDMI explanation gets activated and the dialog management component responds with the liked modalityindependent dialog output. The demand for the visual channel is handled by the fission using the nomination concept, which is presented in section IV-B.

Input Device Component for ASR: The device component for ASR recognizes the user's verbal utterances and tries to convert it into a corresponding speech interaction input. The ASR component's recognizer is able to work in four different ways. (1) Recognition can be done based on certain given parameters. In this case the ASR component builds up the grammar on a given interaction input, e.g. based on the choices in the second half of Listing 2. Secondly, (2) nomination templates are added to the grammar, that works like a matching algorithm for regular expressions. This allows the recognition of utterances like those starting with: "(do not)[show|tell|give] me ([more|additional]) information about ...". This allows the system to respect user-initiated UI specifications and dialog requests. In addition to that, (3) the ASR component enhances the grammar with items to support deictic references (e.g. "this one", "that", ...). This allows the fusion to resolve cross-modal interactions like pointed references in combination with verbal triggers. Finally the ASR component is able to (4) analyze inputs or input fragments in dictation mode.

Using these techniques the ASR component is able to analyze the sentence "Show me more information about the HDMI cable." Based on the sentence's starting sequence the component can infer (i) an eventual nomination for a given channel as well as (ii) an explanation request. Subsequently the ASR recognizes (iii) the HDMI cable (either by identifying a parameter via a grammar (1) or via dictation mode (4)). In combination the ASR device component is able to send an interaction input including diverse input parameters plus confidence scores (see Listing 3).

B. Interplay of Fission and Fusion

Up to now, we described the common HCI loop, as it can be more or less realized in other approaches, too. Next, we describe how the three aspects of interplay between fission and fusion are realized by extending the architecture with the *nomination manager*, the *content manager*, and the possibility to realize a *dialog management bypass*. Concrete examples from the scenario elucidate the realized interplay.

Tailoring the fusion to an adaptive UI: Fission and fusion both work on different models and abstraction levels that best fit their respective purposes. Once the fission component has decided on how to present an interface to the user, the fusion component needs to be informed on the resulting interaction possibilities. Based on that, the fusion is able to decide if different user inputs are ambiguous, conflicting, or reinforce each other.

Therefore the fusion component needs to be provided with an Abstract Interaction Model (AIM) that states all actions in the domain at hand the user could possibly trigger via the available input device components. In addition, the AIM must contain all domain specific knowledge on how different inputs should be semantically combined. The AIM uses the concept of graphs with nodes (that represent possible inputs from input device components) and edges (that represent their combinations) to hold this information. Specification is done in GraphML syntax [22] the details of which are out of scope here. Within our approach, the content manager (cf. Figure 1) is responsible to provide this kind of information. After the fission reasoned about the concrete output configuration, the content manager inspects the resulting I_{out} (see Listing 2) and identifies all objects that can be part of a user interaction. The content manager then uses this information to create an AIM from it as visualized in Figure 2.

```
<?xml version="1.0" encoding="utf-8"?>
<interactionOutput dialogID="cable_selection" language="en">
  <outExpression deviceID="PC_7" deviceComponentID="TouchScreen">
    <topic><text objectID="topic">cable selection</text></topic>
    <dialogAct>
      <selection objectID="cable_selection_container">
        <text objectID="cable_selection_container"
          informationID="cable_selection_prompt">Which cable do you want to use
                                                 to connect the devices?</text>
        <!-- two other selectionItems [...] -->
        <selectionItem objectID="hdmi" informationID="hdmi_information">
          <picture objectID="hdmi"
            informationID="hdmi_information">data/c_HDMI.png</picture>
          <text objectID="hdmi"  informationID="hdmi_information">HDMI
            cable</text>
        </selectionItem>
      </selection>
    </dialogAct>
  </outExpression>
  <outExpression deviceID="PC_7" deviceComponentID="ASR">
    <dialogAct>
      <selection objectID="cable_selection_container">
        <!-- two other selectionItems [...] -->
        <selectionItem objectID="hdmi" informationID="hdmi_information">
            <recognitionChoices objectID="hdmi" informationID="hdmi_information">
          <choice text="HDMI" />
          <choice text="HDMI cable"/>
          <choice text="H. D. M. I. cable"/>
        </recognitionChoices>
        </selectionItem>
      </selection>
    </dialogAct>
  </outExpression>
</interactionOutput>
```

Listing 2. Excerpt from the interaction output for the dialog output from Listing 1. Besides the visual output to be presented on a touchscreen, the model provides recognition choices to build up a grammar for automatic speech recognition (ASR).

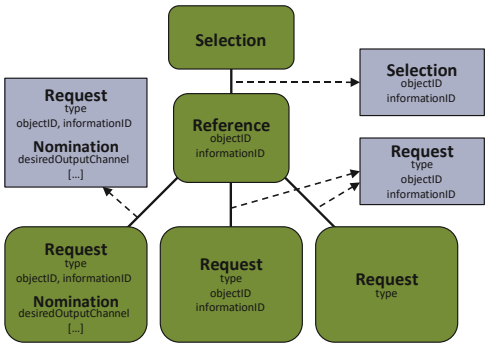

Figure 2. The AIM created by the Content Manager from the interaction output of Listing 2. The graph expresses all possible inputs (green) and their resulting combinations (rectangles in blue).

The AIM shows, that the user can state *selections*, can make *references* to objects, and can perform *requests* for additional information in several ways. To elucidate this, imagine the situation where the user states “Give me more information about that” and at the same time points at the HDMI cable. In such a situation, the ASR component would raise an input containing just a *request* of *type* ’explanation’. The gesture component would raise an input containing a *reference* to the *objectID* ’hdmi’ and *informationID* ’hdmi information’. As defined in the AIM via the edges between the input nodes, the fusion component is able to combine these two inputs and create a complete request, that contains the type ’explanation’, as well as the objectID and informationID of the HDMI cable. In addition, the confidence values given by the input components are taken into account, to make sure only the most probable input results are forwarded to the dialog management.

The main benefit of using a dedicated component like the content manager to create the AIM is based on the fact that it allows the input fusion to be domain independent and reusable in completely different applications. Domain specific knowledge like what kind of inputs exist and how these have to be combined is completely stated in the AIM. *User demands for information presentation:* We introduce the nomination manager (depicted in Figure 1) to tailor the interaction’s information flow towards the user. We assume that respecting user’s explicit demands for any modality can increase a user’s perceived credibility and reliability of an intelligent system. It is the fission component, which is able to respect those user-initiated UI demands at runtime. But it is the fusion component, which is able to identify those configuration nominations on a semantic level.

In our second example the user utters the wish: “Show me more information about the HDMI cable.” The ASR component analyzes the utterance and sends a user request as interaction input I_{in} to the fusion (see Listing 3).

```
<?xml version="1.0" encoding="utf-8"?>
<interactionInput dialogID="cable_selection"
                  dateTime="2013-05-17T16:31:27.2633254+01:00"
                  deviceID="PC_7" componentID="ASR" >
  <listen>
    <request type="explanation" objectID="hdmi"
             informationID="hdmi_information" confidence="0.82">
      <nomination desiredOutputChannel="visual"/></request>
    <request type="explanation" objectID="cinch"
             informationID="cinch_information"
             confidence="0.18">
      <nomination desiredOutputChannel="visual"/></request>
  </listen>
</interactionInput>
```

Listing 3. The interaction input message as sent by *PC_7*’s *ASR* component after the user said: “Show me more information about the HDMI cable.”

The message contains the device and component IDs as well as the ASR’s observed input possibilities including the speech recognizer’s confidence values. As described earlier, the ASR component can make use of different pre-defined recognition templates, which refer to different desires or dislikes of a nomination. In Figure 3 the XML Schema for a request and the included nomination is visualized, where one can see that nominations can express diverse desires and dislikes.

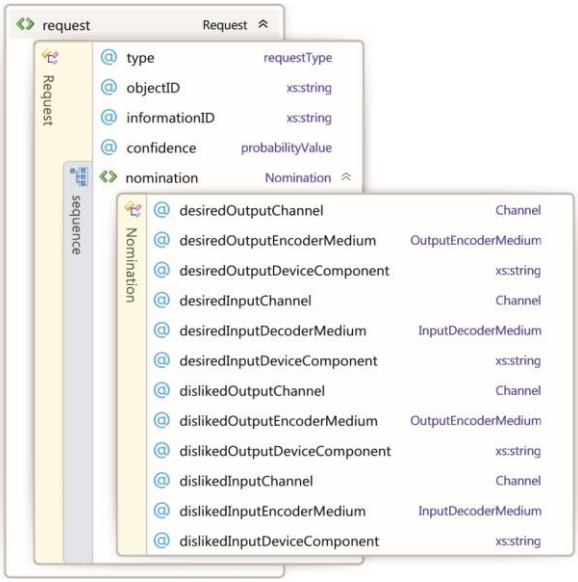

Figure 3. Visualization of the XML Schema definition for a request with nomination specification. It includes diverse attributes that are used to characterize the user’s demands on the UI configuration.

The fusion component analyzes its input and is able to recognize the distinct nomination for the visual channel. Based on the different confidence values the fusion decides that the current input represents a request concerning HDMI. Accordingly, the fusion informs the nomination manager with a new nomination containing only one identified desire (see Listing 4). The desire’s probability is set to 1.0, since the fusion does not have further indications for any other desired output channel concerning the requested HDMI explanation. The nomination manager is able to aggregate different nominations for any specified dialog output (referenced by a dialog ID) or information (referenced by an information ID).

Right after sending the nomination message, the fusion passes the modality-independent HDMI explanation request to the dialog manager. In turn, the dialog manager responds with a suitable dialog output containing the HDMI explanation. The fission inspects the output and identifies the dialog ID **hdmi_explanation**. While reasoning about the dialog’s modality-specific representation, the fission consults the nomination manager concerning nominations for the actual dialog output. The fission’s reasoning algorithm is able to respect any stored nomination with a certain dialog or information ID that occurs within the processed dialog output description. The resulting visual rendering is displayed in Figure 4.

```
<?xml version="1.0" encoding="utf-8"?>
<nomination dialogID="hdmi_explanation">
  <desiredOutputChannel channel="visual" probability="1"/>
</nomination>
```

Listing 4. The nomination for the desire to perceive the hdmi_explanation via the visual channel based on the user’s demand.

In that way the nomination manger supports the fission with additional knowledge for its reasoning process. Be-

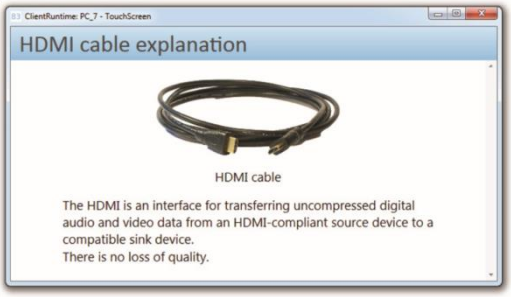

Figure 4. The visual HDMI explanation as requested by the user. Alternatively the system is able to adapt the UI to other specified nominations (e.g. including additional audio comments as multimodal output).

yond the given scenario the nomination manager is able to handle nominations with a reference to certain desired device components or even expressing a certain dislike.

Resolution of ambiguities using a bypass: In contrast to conventional GUI input technologies like mouse and keyboard, emerging technologies used for multimodal interaction, such as speech or gesture recognition, often provide inputs affected with uncertainty. And it is the interaction management that must deal with this new kind of ambivalent data. Be it that sensors report false or unclear interpretations or that the user itself performs ambiguous or even conflicting inputs. Accidentally or on purpose, it can easily happen, that input fusion cannot clearly decide on the user's intentions. In such cases, an intelligent system should deal with such a situation by providing helpful feedback to the user, and offer him the possibility to resolve existing ambiguities.

In our scenario imagine the situation where a user wants to select the SCART cable. He is pointing on the correct cable but mixes their names up and says: "HDMI cable" (as illustrated in Figure 5). This results in interaction inputs from the ASR component as well as from the gesture recognition component. Both inputs contain different object references. The input fusion component then identifies these as conflicting inputs and hence is not able to derive a definite user input.

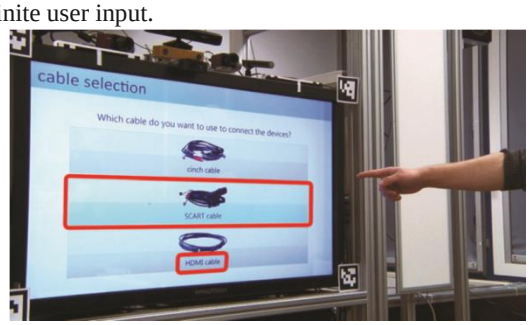

Figure 5. The rendered selection dialog act. A user performs an ambiguous input at the selection task. Pointing on *SCART cable* but accidentally saying: "The *HDMI cable*." (The red colored overlay is not displayed by the system.)

In order to deal with such a situation, we propose a direct cooperation of fusion and fission to resolve the ambivalent input. This allows bypassing the dialog management resulting in a shortened HCI loop (cf. Figure 1). As all necessary information is already present within the fusion module, a generic dialog output as shown in Listing 5 can automatically be constructed and forwarded to the fission module to be rendered resulting in the output shown in Figure 6. This relieves the dialog management from the necessity to explicitly model such additional dialogs, which do not contribute to the overall dialog flow.

```
<?xml version="1.0" encoding="utf-8"?>
<dialogOutput dialogID="cable_selection">
  <topic>
    <abstractInformation objectID="topic" informationID="conflict_topic"/>
  </topic>
  <dialogAct>
    <selection objectID="cable_selection_container"
               informationID="selection_conflict">
      <abstractInformation objectID="scart"
informationID="scart_information"/>
      <abstractInformation objectID="hdmi"
informationID="hdmi_information"/>
    </selection>
  </dialogAct>
</dialogOutput>
```

Listing 5. An intermediate dialog output created by the fusion module when conflicting user inputs are detected. Using a bypass, this dialog output is directly forwarded to the fission module.

Figure 6. Ad hoc conflict resolution by the interaction management. The inferred interaction output as displayed on PC 7's touch screen.

At the time when the ambiguity is resolved by the user, the fusion detects a valid input and sends a dialog input to the dialog management. In turn the normal dialog sequence is continued. This bypass approach addresses interaction related conflicts within the interaction management according to the principle: solve the problems where they occur. This generic approach works without depending on the dialog management's functionality concerning conflict resolution.

V. Discussion

The presented architecture extends the interaction management of existing approaches by introducing three conceptual connections between fission and fusion. The architecture is realized in a running system that exemplarily supports the user in setting up a home cinema system, by advising and explaining alternative connections between different devices. The presented approach is domainindependent, pure model-based, and realizes adaptive behavior in real time. Applied models can be different depending on the domain and AIMs being inferred at runtime depend on this current context.

The *content manager* establishes the connection between the adaptive fission component and the input fusion component working on their respective models by creating an abstract interaction model (AIM). This allows for a flexible and dynamic system output, while preserving a robust input recognition tailored to the current output configuration. Using a dedicated component for this task facilitates domain independence, helps separating responsibilities, provides clear interfaces, and facilitates debugging. Currently, the content manager performs a hard coded model transformation using the abstract interaction output of the fission component. This slightly conflicts the otherwise purely model based approach.

The concept of the *nomination manager* allows to affect the fission's decision process. It works like a structural facade design pattern, which provides a uniform interface to influence the fission process. Nominations can affect the system's output on different levels. They can reference a dialog ID (as used in the example) to bias the fission's decision for each information item within the referenced dialog act, or they can selectively refer to specific information IDs instead. Referencing information IDs leads to a biased fission decision only for the specified information items within an arbitrary dialog act. In our current implementation the fission "consumes" each of the provided nominations that match within a given HCI loop. With the appearance of a new dialog output older nominations get removed. The removal is motivated by the assumption that user-initiated UI nominations represent an amendment to the fission's reasoning result, which is based on knowledge representing a certain context of use (CoU). In other words: nominations for adaptive behavior lose their validity from the moment when a changing CoU initiates an adaption of the present UI configuration. The nomination as amendment addressed the previous UI configuration in the former CoU. Early tests arouse the suspicion that this assumption may be not correct in some situations. Further investigation is necessary to identify the relevant parameters of a CoU, which cause a user to emend a certain UI using the nomination concept. It might be interesting to analyze the history of interaction, the fission's decisions, as well as the user's nominations as amendment in a certain CoU. Correlations in the temporal evolution could be used to apply supervised learning approaches to improve the fission's results.

The proposed shortcut in the HCI loop for resolving ambiguities in the user input stems from the fact, that recognition based input methods are applied (e.g. speech and gesture recognition). These do not offer the decidedness of classic mouse and keyboard inputs and can lead to situations, where inputs are unclear or even contradict each other. In addition, it may be the user itself that performs such ambiguous inputs. Using a direct connection between the ambiguity detecting input fusion and the output generating fission component, such ambiguities can be resolved in specific ad hoc interactions with the user. It turned out that this bypass concept works very reliable and user-friendly. Furthermore, separating the dialog management from the used input techniques dismantles the obligation to explicitly model these additional dialogs that do not contribute to the overall dialog flow. Currently, the generated dialogs are quite simple and just present a list of possible alternatives the user is supposed to select. Additional information might be useful that reveals the reasons for the system's current indecisiveness. This might also lead to a system's increasing credibility and perceived trustworthiness. Further investigation is needed to check if this is appropriate for all kinds of misunderstandings or if an additional component (like the nomination manager) that exclusively handles such queries could be advantageous.

VI. Conclusion and Future Work

As specified, for an adaptive multimodal system that shall meet the high standard of human communication and interaction skills, fission and fusion can gain benefit from each other while enhancing their reliability. Present approaches still follow a strictly one track way in realizing the HCI loop, where information seems to be caught in one big stream.

In this paper we proposed possible direct connections of multimodal fission and fusion realized by open flows and exchange of information that allow a more exhaustive utilization of their respective capabilities. All presented components support our model-driven interaction concept and work domain independent.

The presented architecture reveals three ways of collaboration between fission and fusion that enhance the capabilities of the interaction management not only by coordination, but also by providing additional functionality. That in turn helps fulfilling the users' individual needs and demands. The *content manager* provides the fusion component with all information that is necessary to tailor its functionality to a dynamic and adaptive user interface specified by the fission component at runtime. The *nomination manager* provides additional functionality that allows the user to directly and naturally state different demands on the presentation of information without the use of additional tools. And finally, the *bypass approach* allows a direct resolution of occurring ambiguities within the interaction management, without straining the dialog management. Based on a model-driven approach, these automated processes add value that can easily be applied to other domains without additional engineering effort for developers.

The described concepts should enhance the usability of adaptive user interfaces. To support this assumption, we plan to perform an analysis of variance of two systems, where only one is enhanced with the mentioned aspects of interplay of fission and fusion. Amongst other measures, we plan to record user satisfaction, error rates, and task completion times.

The different types of collaboration proposed here require different work to be conducted in the future. We will explore ways of configuring and specifying the content manager's currently hard coded automatic transformation of the interaction output to the abstract interaction model.

Due to the fact, that all data exchange within our approach is done via XML, applying XSLT transformations is an obvious approach worth looking into. We further plan to adjust the fission's reasoning algorithm by analyzing occurring nominations. Applying conclusions to the reasoning process can lead to an automatic learning of the fission component.

References

[1] 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.

[2] M. Blumendorf, D. Roscher, and S. Albayrak, "Dynamic user interface distribution for flexible multimodal interaction," in ICMI - Workshop on Machine Learning for Multimodal Interaction, ser. ICMI-MLMI '10. New York, NY, USA: ACM, 2010, pp. 20:1–20:8.

[3] C. Duarte and L. Carric¸o, "A conceptual framework for developing adaptive multimodal applications," in IUI '06: Proc. of the 11th int. conf. on Intelligent User Interfaces. New York, USA: ACM, 2006, pp. 132–139.

[4] S. Oviatt, "Multimodal interfaces," in The HumanComputer Interaction Handbook, A. Sears and J. A. Jacko, Eds. CRC Press, 2007, pp. 413–432.

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

[6] 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.

[7] D. Costa and C. Duarte, "Adapting multimodal fission to user's abilities," in Proc. of the 6th int. conf. on Universal access in human-computer interaction: design for all and eInclusion - Volume Part I, ser. UAHCI'11. Berlin: Springer, 2011, pp. 347–356.

[8] M. E. Foster, "State of the art review: Multimodal fission," University of Edinburgh, Public Deliverable 6.1, September 2002, cOMIC Project.

[9] C. Rousseau, Y. Bellik, F. Vernier, and D. Bazalgette, "A framework for the intelligent multimodal presentation of information," Signal Process., vol. 86, no. 12, pp. 3696– 3713, 2006.

[10] M. D. Hina, C. Tadj, A. Ramdane-Cherif, and N. Levy, "A Multi-Agent based Multimodal System Adaptive to the User's Interaction Context," in Multi-Agent Systems – Modeling, Interactions, Simulations and Case Studies. InTech, 2011, ch. 2, pp. 29–56.

[11] D. Roscher, M. Blumendorf, and S. Albayrak, "A meta user interface to control multimodal interaction in smart environments," in Proc. of the 14th int. conf. on Intelligent user interfaces, ser. IUI '09. NY, USA: ACM, 2009, pp. 481–482.

[12] F. Honold, F. Schussel, and M. Weber, "Adaptive prob-¨ abilistic fission for multimodal systems," in Proceedings of the 24th Australian Computer-Human Interaction Conference, ser. OzCHI '12. New York, NY, USA: ACM, November, 26–30 2012, pp. 222–231.

[13] P. Atrey, M. Hossain, A. El Saddik, and M. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," Multimedia Systems, vol. 16, pp. 345–379, 2010.

[14] L. Nigay and J. Coutaz, "A generic platform for addressing the multimodal challenge," in CHI '95: Proc. of the SIGCHI conf. on Human factors in computing systems. ACM Press, 1995, pp. 98–105.

[15] M. T. Vo and C. Wood, "Building an application framework for speech and pen input integration in multimodal learning interfaces," in Int. Conf. on Acoustics, Speech, and Signal Processing, 1996. ICASSP-96, vol. 6. IEEE, May 1996, pp. 3545 – 3548 vol. 6.

[16] H. Holzapfel, K. Nickel, and R. Stiefelhagen, "Implementation and evaluation of a constraint-based multimodal fusion system for speech and 3D pointing gestures," in Proc. of the 6th Int. Conf. on Multimodal Interfaces, ser. ICMI '04. New York, NY, USA: ACM, 2004, pp. 175–182.

[17] N. Pfleger, "Context based multimodal fusion," in ICMI '04: Proc. of the 6th int. conf. on Multimodal interfaces. NY, USA: ACM, 2004, pp. 265–272.

[18] J. Bouchet, L. Nigay, and T. Ganille, "Icare software components for rapidly developing multimodal interfaces," in ICMI '04: Proc. of the 6th int. conf. on Multimodal interfaces. New York, NY, USA: ACM, 2004, pp. 251–
258.

[19] B. Dumas, R. Ingold, and D. Lalanne, "Benchmarking fusion engines of multimodal interactive systems," in ICMIMLMI '09: Proc. of the 2009 Int. Conf. on Multimodal Interfaces. New York, NY, USA: ACM, 2009, pp. 169–
176.

[20] L. Wu, S. Oviatt, and P. Cohen, "From members to teams to committee-a robust approach to gestural and multimodal recognition," IEEE Transactions on Neural Networks, vol. 13, no. 4, pp. 972 – 982, Jul 2002.

[21] F. Schussel, F. Honold, and M. Weber, "Using the transfer-¨ able 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.

[22] U. Brandes, M. Eiglsperger, I. Herman, M. Himsolt, and M. Marshall, "Graphml progress report: Structural layer proposal," in Graph Drawing, ser. LNCS, P. Mutzel, M. Junger, and S.

Leipert, Eds.¨ Springer Berlin, 2002, vol. 2265, pp. 501–512.

[23] M. Schroder, "The semaine api towards a standards-based¨ framework for building emotion-oriented systems," Adv. in Hum.-Comp. Int., vol. 2010, p. 21, January 2010.

[24] A. Schmidt, "Implicit human computer interaction through context," Personal Technologies, vol. 4, pp. 191–199, June 2000.

[25] F. Honold, F. Schussel,¨ 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.