Multimodal Fusion on Mobile Devices *

Frédéric Aebi
University of Fribourg, Switzerland
frederic.aebi@unifr.ch

ABSTRACT
The complexity of computer applications has risen enormously during the last decade. Nowadays, machines are equipped with features such as speech or gesture recognition as well as many other modalities. There are also applications that are able to interpret more than one modality at the same time. This is where multimodal fusion comes into play. The paper first introduces multimodal fusion and then talks more in detail about fusion engines. One of the fusion engines we present is specially designed for mobile devices. The paper also presents a state of the art of multimodal fusion on mobile devices.

Categories and Subject Descriptors
D.2.2 [**Software Engineering**]: Design Tools and Techniques—*Modules and interfaces*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input devices and strategies, Interaction styles*

General Terms
Algorithms, Design, Human Factors

Keywords
Multimodal fusion, fusion engines, mobile devices

1. INTRODUCTION
Multimodal fusion comes into play when users are interacting with computers using more than one modality. As explained in [7], the task of multimodal fusion is to combine and to integrate mono-modal interpretations of multiple modalities into one semantic representation of the intended meaning.

Today, the most important challenge of multimodal fusion on mobile devices is the fusion of sensory data provided by accelerometers, gyroscopes, gravity sensors, etc. A good example for such kind of fusion would be an application that calculates the distance walked by a person. To do this, one could just use the data provided by the accelerometer of the mobile phone and get the linear acceleration out of it by removing gravity. The only thing left to do is to integrate the linear acceleration twice to get the travelled distance. In theory, this seems of course very simple, but it is not the case in practice. Accelerometer data is very noisy and it is thus inevitable to apply sensor fusion (with other sensors as the gyroscope) to avoid errors in such calculations.

Multimodal fusion can be applied in different manners. We can either use data-level, feature-level or decision-level fusion. Data-level fusion is the lowest level of fusion and is applied directly on the data. As claimed in [1], this type of fusion is used to merge data from identical sensor types (e.g., two cameras). Feature-level fusion (also known as early fusion) works a bit differently. It takes each stream of sensory data, analyses it, extracts features and finally fuses them. The last one is decision-level fusion (also known as late fusion) which works at the semantic level. Its concept is to fuse individual decisions or interpretations. Fusion is done after the recognition process. As mentioned in [1], it is the most commonly found type of fusion. From [8], we deduce that it is also the level of fusion used on mobile devices.

Section 2 focuses on the CASE and CARE models [1][6]. Fusion engines are discussed in section 3 where we describe how they can be classified and present three of them in particular, namely PATE [7], ICARE [2][3] and ACICARE [8].

2. MODELS OF MULTIMODAL FUSION
Multimodal interaction can be formalized by two conceptual design spaces. As presented in [6], one of them is called the CASE (Concurrent, Alternate, Synergistic and Exclusive) model (see figure 1), which consists of describing the communication types of multimodality on machine-side.

The *sequential* column means that only one modality is used at a given time. The *parallel* column on the other hand stands for interactions where multiple modalities are employed simultaneously. The rows refer to the presence of fusion of modalities. The *combined* row represents all interactions where fusion is necessary. The *independent* row signifies that there is no coreference between the modalities

*Masters Seminar on Multimodal Interaction on Mobile Devices at the University of Fribourg, Switzerland : http://diuf.unifr.ch/main/diva/teaching/seminars/seminar-multimodal-interaction-mobiles-devices

and that fusion is not necessary. As you might probably have guessed, in this paper we will focus on combined interactions, or more exactly on Alternative and Synergistic interaction techniques.

		Use of Modalities	
		Sequential	Parallel
Fusion of Modalities	Combined	**A**lternate	**S**ynergistic
	Independent	**E**xclusive	**C**oncurrent

Figure 1: The CASE model [6]

The second conceptual design space is called the CARE (Complementarity, Assignment, Redundancy and Equivalence) model. It focuses on the human-side of a multimodal system. *Complementarity* stands for all complementary multimodal events (e.g., the user says "I want to put this article in the cart" while pointing his finger at the green sweatshirt). It can occur sequentially or in parallel. *Assignment* means that a given state can be reached only by using one modality (e.g., the user removes the green sweatshirt from the cart by a drag and drop movement). *Redundancy* occurs when multiple modalities have the same expressive power (e.g., the user says "I want to put the green sweatshirt in the cart" while selecting the green sweatshirt and making a drag-an-drop movement to the cart). *Equivalence* means that a given state can be reached by multiple modalities (e.g., the user can remove the green sweatshirt from the cart either by a drag and drop movement or by saying : "I want to remove this article from the cart" while pointing his finger at the green sweatshirt) [1].

3. FUSION ENGINES

As pointed out in [6], a *fusion engine* is a computational element whose task is to combine informations extracted from the actions a user performed on an input device into meaningful commands. As you will read below, there exist a couple of classifications of fusion engines.

3.1 Classification

A first classification of fusion engines has already been discussed in section 1, namely the **level** of fusion (data-level, feature-level or decision-level fusion). A second classification is the **notation** which stands for the language that is used to represent the behaviour of a fusion engine. Another one is the **fusion type** which can be either frame-based (in a tabular form), unification-based (construction of commands according to some rules), procedural (input events are algorithmically managed and combined according to the state space) or hybrid (combination of frame-based and unification). As the authors mention it in [6], the fusion type describes how fusion is performed.

The next element concerns the set of **input devices** which are used for the fusion engine. Some general examples that are listed in [6] are speech, gesture, keyboard, mouse, eye gaze, tactile surface, etc. When talking about mobile devices, there are a lot of other input devices that come into play. Mobile devices are equipped with GPS localization, gyroscopes, accelerometers, proximity sensors, ambient light sensors, etc. (more on that in section 3.2).

The fifth classification is **ambiguity resolution**. As already mentioned in section 1, multimodal fusion is there to reduce uncertainties and to resolve ambiguous hypotheses between the different input events. As pointed out in [6], some fusion engines use a *speech over gesture* mechanism or vice versa, others use a principle called *N-Best list* which is an iterative approach of selecting the best possible input event. Finally, it is also possible to use context-based resolution.

Time representation is the last classification we are going to talk about. As claimed in [6], the notion of time helps fusion engines to transform the events received from the different input devices into specific commands. Time representation can be defined at two different levels. The first one is called **quantitative** and represents temporal behaviour changes either during a certain period or at a precise moment in time. The second level is named **qualitative** and deals with the notion of event ordering (e.g., precedence, succession, simultaneity).

3.2 Examples

The following section describes and presents four fusion engines. As you will see, each of them has been characterized according to the fusion engine classifications explained above.

3.2.1 PATE

PATE [7] is a fusion engine that has been used for a bathroom design tool named COMIC. The idea behind this project is that users can configure and design their own bathroom through a multimodal dialogue system. COMIC makes use of two screens. One of them displays an animated talking head which guides the user through the application. The second screen allows the user to create the bathroom by adding walls, doors, windows and other elements by the use of a pen.

Fusion	
Level	Decision-level
Notation	XML
Fusion Type	Unification and overlay
Input Devices	Speech, pen
Ambiguity Resolution	N-Best list

Time Representation	
Quantitative	Yes
Qualitative	Yes

Models of Multimodal Fusion	
CASE	Synergistic
CARE	Complementarity, Equivalence

Figure 2: Characteristics of PATE [6]

There is a continuous dialogue that takes place between the talking head and the user. For instance when the user draws a wall, the talking head asks the user : "Please tell me the length of this wall." This is where the speech modality comes

into play. The user could then say : "Five meters". This way, the system knows how long the wall should be.

PATE works as a regular production rule system. It has a working memory (WM) which contains working memory elements (WME). A WME is nothing else than the activation value of some input data (e.g., when a user adds a wall). The more a WME is used, the more its activation value increases and vice versa. Figure 3 illustrates the basic architecture of the PATE system.

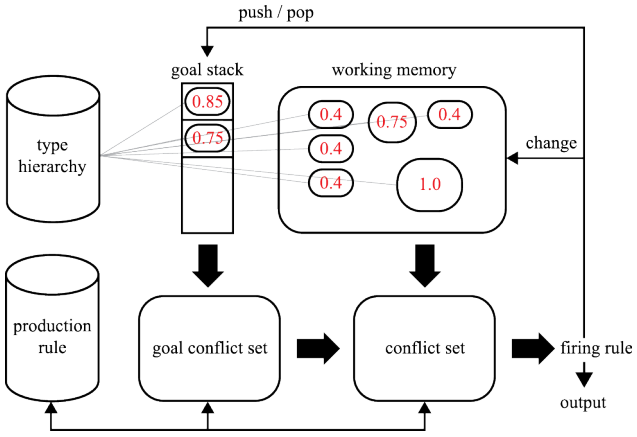

Figure 3: Architecture of the PATE system [7]

The attentional focus of the system is described in the goal stack. The WME on top of it represents the current system focus. Before the input data is added to the goal stack, it is translated to an XML document in the type hierarchy. In case of Complementary (cf. section 2), PATE uses unification and overlay in order to determine if the data structures coming from more than one input modality are consistent. Overlay is also very important in cases where the informations provided by the user are conflicting (e.g., the user points at a purple wall while saying "Show me this green wall").

PATE uses an N-Best list principle to deal with conflict resolution. This happens in the conflict set (cf. figure 3). As explained in [7], the system takes all the elements from the goal stack and the WM and computes a couple of rules that could be triggered according to some conditions defined in the system. Then it picks out the rule which has the highest score and finally executes its associated actions.

3.2.2 ICARE

Another fusion engine we are going to present in this paper is ICARE [2][3], which uses a component-based approach to deal with multimodal fusion. Indeed, there are three types of components, namely Device components, Interaction Language components and Composition components. ICARE refers a lot to the CARE model (cf. section 2). In fact the acronym stands for Interaction-CARE.

This fusion engine has been used to implement a multimodal flight simulator for a French military plane called FACET. As claimed in [3], the simulation allows users to pilot and to navigate the plane, to manage missions and the armory system, and also to communicate with other pilots.

The fact that there are a lot of input modalities involved in this system make FACET and the use of ICARE a very interesting case study for multimodal fusion.

The first input modality is called HOTAS (Hands On Throttle And Stick) and consists of two joysticks, one for the left and the other for the right hand. The purpose of HOTAS is to allow the user to pilot the plane. Another input modality is the helmet visor which is used for target selection. A couple of commands can be triggered through speech inputs. Finally, there is also a tactile surface on the ground.

Fusion	
Level	Decision and data-level
Notation	Melting Pot
Fusion Type	Frame-based
Input Devices	Speech, helmet visor, tactile surface, GPS localization, magnetometer, mouse, keyboard
Ambiguity Resolution	Context-based resolution

Time Representation	
Quantitative	Yes
Qualitative	No

Models of Multimodal Fusion	
CASE	Alternate and Synergistic
CARE	All

Figure 4: Characteristics of ICARE [6]

As you can see in figure 4, ICARE uses a Melting Pot notation which represents an event by a structural part and a temporal information. This implies that fusion will follow the principles of Complementarity, near time and also context rules [6].

As previously mentioned, ICARE uses three type of components. The Device component is an additional layer on top of the driver of a physical device. Such components are there to make an abstraction of the data provided by a certain input device.

The Interaction Language component is there for the fusion engine to establish a link to the Device component. Indeed, an Interaction Language component takes the data collected from the Device component and transforms it into a machine understandable command or representation. As the authors describe it in [3], these two components are the fundamental blocks that help us define modalities.

Finally, there are also Composition components. This is where the fusion of the various input data takes place. Composition components in ICARE are divided into three categories, namely Complementarity, Redundancy and Redundancy/Equivalence. As you may have noticed, the fusion engine does not make use of the Assignment and Equivalence property. The use of Complementarity and Redundancy follows the principles already discussed in section 2. The Redundancy/Equivalence is much more complex than a simple Redundancy Composition component. Indeed, it makes use of two different strategies called *eager* and *lazy*.

The *eager* principle is that when the Composition component receives a command, it does not wait for some additional input that may help the system to understand the intention of the user. On the other hand, the *lazy* strategy

waits for a second input to proceed. The advantages of the *eager* principle is that commands are executed faster. The drawback is that the fact to use only one input modality may lead to a lack of precision. For the *lazy* strategy, it is exactly the opposite. It is slower but much more precise.

3.2.3 ACICARE

In this section we will introduce ACICARE which is adapted and optimized for multimodal fusion on mobile devices. As pointed out in [8], ACICARE is a combination of ICARE and ACIDU which we will talk more about later.

The fusion engine has been used on an SPV c500 mobile phone to implement a contact manager. One can create new contacts using speech, the mobile phone's keyboard and a dedicated key. These three input modalities can be combined according to the Equivalence and Complementarity principle of the CARE model (cf. section 2).

The characteristics of ACIARE are almost the same as for ICARE. The only thing that differentiates both approaches is the ACIDU layer (on top of ICARE) which gathers data from the different used functionalities of the fusion engine. As the authors claim in [8], there is a connection between each ICARE component and ACIDU.

Every time an ICARE component receives an event, its content, timestamp and component name are stored in an ACIDU log file. This way, ACIDU is always informed whether the received event is at Device, at Interaction Language or at Composition level (cf. section 3.2.2). The fusion engine can then assemble all the informations and apply the previously mentioned principles of the CARE model.

As you can see ICARE is a very generic approach to deal with multimodal fusion. In this section we used ACICARE as example but indeed ICARE could be applied to many other multimodal applications that run on mobile devices.

4. CONCLUSION

In this paper, we first introduced multimodal fusion in terms of its purpose and importance according to nowadays' technologies. We also gave an overview of the CASE and CARE model which both formalize the concept of multimodal fusion.

The rest of the paper focused on fusion engines. Here we gave some examples on how multimodal fusion can be performed. According to what the authors pointed out in [6], we saw that fusion engines can be classified in terms of level, notation, fusion type, input devices, ambiguity resolution and finally time representation. We presented three fusion engines in particular, namely PATE [7], ICARE [2][3] and finally ACICARE [8].

Indeed there exist much more fusion engines which we have not addressed in this paper. In [4], the author presents MEngine which uses data-level fusion and which is based on a finite state machine notation and a procedural fusion type. A model-based framework called FAME is presented in [5]. FAME uses a behavioural matrix notation and relies on a hybrid fusion type. It has been used to implement a digital talking book.

As there exist a lot of fusion engines, it is also every developer's duty to tell them apart and to use them appropriately. The classifications of the fusion engine one plan to use for their multimodal applications need to be studied very carefully. For instance, it would not make sense to use PATE for a multimodal application that does not use Complementary.

Multimodal fusion is a very hot topic and a lot of research has been done on it during the last decade. Even more trendier are multimodal applications on mobile devices. In this paper, we talked about ACICARE, which has been used for a multimodal contact manager on an SPV c500 mobile phone. It would also have been interesting to explore multimodal fusion on Google Android or on the Apple iPhone, which are the current leaders on the mobile market. The problem is that such companies do not really reveal how multimodal fusion is done on their devices and it is thus very difficult to do some research on it.

To conclude, we can say that there are still a lot of unanswered questions on how multimodal fusion is done on nowadays' mobile devices. But all the general research that has been done on it leaves us with the assumption that somehow, the techniques that are used on mobile phones must extend some existing approaches that have been discussed in this paper.

5. REFERENCES

[1] J. Bapst, O. Abou Khaled, D. Lalanne, and E. Mugellini. Course material. *Multimodal Interfaces*, 2012.

[2] J. Bouchet and L. Nigay. Icare: A component-based approach for the design and development of multimodal interfaces. *In Extended Abstracts*, pages 1325–1328, 2004.

[3] J. Bouchet, L. Nigay, and T. Ganille. Icare software components for rapidly developing multimodal interfaces. *In Proceedings of the 6th international conference on Multimodal interfaces*, pages 251–258, 2004.

[4] M.-L. Bourguet. A toolkit for creating and testing multimodal interface designs. *In Proceedings of the 14th French-speaking conference on Human-computer interaction*, pages 239–242, 2002.

[5] C. Duarte and L. Carriço. A conceptual framework for developing adaptive multimodal applications. *In Proceedings of the 11th international Conference on intelligent User interfaces*, pages 132–139, 2006.

[6] D. Lalanne, L. Nigay, P. Palanque, P. Robinson, J.Vanderdonckt, and J.-F. Ladry. Fusion engines for multimodal input: A survey. *In Proceedings of the 2009 international conference on Multimodal interfaces*, pages 153–160, 2009.

[7] N. Pfleger. Context based multimodal fusion. *In Proceedings of the 6th international conference on Multimodal interfaces*, pages 265–272, 2004.

[8] M. Serrano, L. Nigay, R. Demumieux, J. Descos, and P. Losquin. Multimodal interaction on mobile phones: development and evaluation using acicare. *In Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pages 129–136, 2006.