Multimodal Interfaces: A Survey of Principles, Models and Frameworks

Bruno Dumas[1], Denis Lalanne[1], and Sharon Oviatt[2]

[1] DIVA Group, University of Fribourg
Bd de Pérolles 90, 1700 Fribourg, Switzerland
{bruno.dumas,denis.lalanne}@unifr.ch
[2] Incaa Designs
821 Second Ave., Ste. 1100, Seattle WA. 98104
oviatt@incaadesigns.org

Abstract. The grand challenge of multimodal interface creation is to build reliable processing systems able to analyze and understand multiple communication means in real-time. This opens a number of associated issues covered by this chapter, such as heterogeneous data types fusion, architectures for real-time processing, dialog management, machine learning for multimodal interaction, modeling languages, frameworks, etc. This chapter does not intend to cover exhaustively all the issues related to multimodal interfaces creation and some hot topics, such as error handling, have been left aside. The chapter starts with the features and advantages associated with multimodal interaction, with a focus on particular findings and guidelines, as well as cognitive foundations underlying multimodal interaction. The chapter then focuses on the driving theoretical principles, time-sensitive software architectures and multimodal fusion and fission issues. Modeling of multimodal interaction as well as tools allowing rapid creation of multimodal interfaces are then presented. The article concludes with an outline of the current state of multimodal interaction research in Switzerland, and also summarizes the major future challenges in the field.

1 Introduction

Of the numerous ways explored by researchers to enhance human-computer communication, multimodal interaction has shown much development in the past decade. On one hand, multimodal interfaces target a more "human" way of interacting with computers, by means of speech, gestures or other modalities, as well as being preferred over unimodal interfaces by users [49]; on the other hand, multimodal interfaces have been demonstrated to offer better flexibility and reliability than other human/machine interaction means [51].

As a research subject, multimodal interaction encompasses a broad spectrum of research domains, from cognitive psychology to software engineering, including human-computer interaction, which is already cross-disciplinary. While cognitive psychologists study how the human brain processes information and interacts through various modalities, interaction practitioners are interested by how humans use multimodal interfaces, and finally software engineers are interested in building tools and

D. Lalanne and J. Kohlas (Eds.): Human Machine Interaction, LNCS 5440, pp. 3–26, 2009.

systems supporting the development of such multimodal interfaces, thus studying software architectures and multimodal processing techniques.

Cognitive psychologists have extensively studied how humans perceive, process, and express multimodal information; their conclusions are of interest for developers and HCI practitioners. The creation of a typical multimodal application requires a number of different components and careful implementation work. Hence, "good practices" and algorithms regarding the general architecture of a multimodal application, its fusion and fission engines or dialogue management components emerged during the past 20 years [13, 62]. In a more theoretical way, modeling of multimodal interaction and, generally speaking, of the underlying human-machine dialog has seen extensive work. This theoretical work leads to the definition of a number of languages dedicated to multimodal data description, multimodal human-machine dialog modeling or multimodal applications scripting. Together with these different languages, different tools targeted at expediting the creation of multimodal interfaces have appeared.

This chapter runs the spectrum from cognitive foundations to development tools, with a particular emphasis on the multimodal processing aspects. The article is not an exhaustive summary of the findings and issues in this broad and multidisciplinary field, but rather presents the major issues and findings, with an emphasis on the driving principles for the creation of multimodal interfaces, their models, and programming frameworks. The chapter begins with a global view on multimodal interaction, with a presentation of its aims and advantages, its features, and cognitive foundations underlying multimodal systems; seminal works, findings and guidelines particular to multimodal interaction conclude this second section. The third section gives a detailed look at theoretical and practical principles of multimodal systems, architectures and key components of such systems; among those key components, fusion engines, fission engines and dialog management all have a dedicated subsection. The third section ends with a view of potential uses of machine learning for multimodal interaction. The fourth section focuses on modeling and creation of multimodal interfaces, with subsections detailing models, modeling languages and programming frameworks for multimodal interaction. The fifth section is devoted to multimodal applications in Switzerland, and the sixth and last section concludes this chapter with future directions.

2 Foundations, Aims and Features of Multimodal Interaction

This section will present the aims underlying multimodal interaction research, as well as the distinctive features of multimodal interfaces compared to other types of interfaces. The first part will present a general view of multimodal systems, and more specifically their aims and advantages. The section continues with a part focused on particular features of multimodal interfaces, compared to standard GUI interfaces. The third part introduces cognitive theories linked to multimodal interaction design. Finally, the fourth part presents seminal works, findings and guidelines in the field of multimodal interaction.

2.1 Aims and Advantages of Multimodal Systems

Multimodal systems are computer systems endowed with multimodal capabilities for human/machine interaction and able to interpret information from various sensory and communication channels. Literally, multimodal interaction offers a set of "modalities" to users to allow them to interact with the machine. According to Oviatt [49], « *Multimodal interfaces process two or more combined user input modes (such as speech, pen, touch, manual gesture, gaze, and head and body movements) in a coordinated manner with multimedia system output. They are a new class of interfaces that aim to recognize naturally occurring forms of human language and behavior, and which incorporate one or more recognition-based technologies (e.g. speech, pen, vision)* ». Two unique features of multimodal architectures and processing are: (1) the fusion of different types of data; and (2) real-time processing and temporal constraints imposed on information processing [46, 54].

Thus, multimodal systems represent a new class of user-machine interfaces, different from standard WIMP interfaces. They tend to emphasize the use of richer and more natural ways of communication, such as speech or gestures, and more generally all the five senses. Hence, the objective of multimodal interfaces is twofold: (1) to support and accommodate users' perceptual and communicative capabilities; and (2) to integrate computational skills of computers in the real world, by offering more natural ways of interaction to humans.

Multimodal interfaces were first seen as more efficient than unimodal interfaces; however, evaluations showed that multimodal interfaces only speed up task completion by 10% [50]. Hence, efficiency should not be considered the main advantage of multimodal interfaces. On the other hand, multimodal interfaces have been shown to improve error handling & reliability: users made 36% fewer errors with a multimodal interface than with a unimodal interface [50]. Multimodal interfaces also add greater expressive power, and greater potential precision in visual-spatial tasks. Finally, they provide improved support for users' preferred interaction style, since 95%-100% of users prefer multimodal interaction over unimodal interaction [50].

2.2 Features

Compared to other types of human/computer interaction, multimodal interaction seeks to offer users a more natural and transparent interaction, using speech, gestures, gaze direction, etc. Multimodal interfaces are hence expected to offer easier, more expressively powerful and more intuitive ways to use computers. Multimodal systems have the potential to enhance human/computer interaction in a number of ways:

- Enhanced robustness due to combining different partial information sources;
- Flexible personalization based on user and context;
- New functionality involving multi-user and mobile interaction.

When comparing multimodal user interfaces (MUI) with standard graphical user interfaces (GUI), it is possible to draw the following differences [54]:

Table 1. Differences between GUIs and MUIs

GUI	MUI
Single input stream	Multiple input streams
Atomic, deterministic	Continuous, probabilistic
Sequential processing	Parallel processing
Centralized architectures	Distributed & time-sensitive architectures

In standard WIMP interaction style (Window, Icon, Menu, Pointing device), a singular physical input device is used to control the position of a cursor and present information organized in windows and represented with icons. In contrast, in multimodal interfaces, various modalities can be used as input streams (voice, gestures, facial expressions, etc.). Further, input from graphical user interfaces is generally deterministic, with either mouse position or characters typed on a keyboard used to control the computer. In multimodal interfaces, input streams have to be first interpreted by probabilistic recognizers (HMM, GMM, SOM, etc.) and thus their results are weighted by a degree of uncertainty. Further, events are not always clearly temporally delimited and thus require a continuous interpretation. Due to the multiple recognizers necessary to interpret multimodal input and the continuous property of input streams, multimodal systems depend on time synchronized parallel processing. Further, as we will see in the following section, the time sensitivity of multimodal systems is crucial to determining the order of processing multimodal commands in parallel or in sequence. Finally, multimodal systems often implement a distributed architecture, to deal out the computation and insure synchronization. Multimodal systems can be very resource demanding in some cases (e.g., speech/gesture recognition, machine-learning augmented integration).

2.3 Cognitive Foundations

The advantages of multimodal interface design are elucidated in the theory of cognitive psychology, as well as human-computer interaction studies, most specifically in cognitive load theory, gestalt theory, and Baddeley's model of working memory [5, 53, 55]. Findings in cognitive psychology reveal:

- humans are able to process modalities partially independently and, thus, presenting information with multiple modalities increases human working memory;
- humans tend to reproduce interpersonal interaction patterns during multimodal interaction with a system;
- human performance is improved when interacting multimodally due to the way human perception, communication, and memory function.

For example, when processing both auditory and visual information during speech, a listener is able to extract a higher rate of lexical intelligibility (Grant & Greenberg [24]). This section thus presents works from cognitive science related to multimodal interaction, following cognitive load theory, gestalt theory and Baddeley's model of

working memory; the section ends with the description of a framework aimed at human performance prediction.

Mousavi et al [44] experimented with presenting students content using partly auditory and partly visual modes. The split-attention effect (Sweller et al. [66]) that resulted "*suggested that working memory has partially independent processors for handling visual and auditory material.*" The authors argued that if working memory is a primary limitation in learning, then increasing effective working memory by presenting information in a dual-mode form rather than a purely visual one, could expand processing capabilities. The results of Mousavi et al. were confirmed by Tindall-Ford et al. [67], who used more general types of tasks than pure mathematical ones, and by Mayer & Moreno [39] who studied the same effect with multimedia learning material. All this work is in line with the cognitive load theory, which assumes a limited working memory in which all conscious learning and thinking occurs, and an effectively unlimited long-term memory that holds a large number of automated schemas that can be brought into working memory for processing. Oviatt [53] applied these findings to educational interface design in testing a number of different user-centered design principles and strategies, showing that user-interface design that minimizes cognitive load can free up mental resources and improve student performance. One strategy for accomplishing this is designing a multimodal interface for students.

In the design of map-based pen/voice interfaces, Oviatt et al. [55] demonstrated that Gestalt theoretic principles successfully predicted a number of human behaviors, such as: users consistently followed a specific multimodal integration pattern (i.e. sequential versus simultaneous), and entrenched further in their pattern during error handling when you might expect them to switch their behavior. Gestalt theory also correctly predicted in this study a dominant number of subjects applying simultaneous integration over sequential integration.

The original short-term memory model of Baddeley & Hitch [6], refined later by Baddeley [5], described short-term or working memory as being composed of three main components: the central executive (which acts as supervisory system and controls the flow of information), the phonological loop, and the visuo-spatial sketchpad, with the latter two dedicated to auditory-verbal and visuo-spatial information processing, respectively. Although these two slave processors are coordinated by a central executive, they function largely independently in terms of lower-level modality processing. This model was derived from experimental findings with dual-task paradigms. Performance of two simultaneous tasks requiring the use of two perceptual domains (i.e. a visual and a verbal task) were observed to be nearly as efficient as performance of individual tasks. In contrast, when a person tries to carry out two tasks simultaneously that use the same perceptual domain, performance is less efficient than when performing the tasks individually. As such, human performance is improved when interacting with two modalities that can be co-processed in separate stores.

Wickens [72][73] also developed a framework, the "multiple resource model", aimed at performance prediction involving coordination between user input and system output modes for different types of tasks. This model suggests that four different dimensions are to be taken into account when predicting coordination versus interference during human task processing involving different modes. The four dimensions considered are stages (perceptual/cognitive vs. response), sensory

modalities (auditory vs. visual), codes (visual vs. spatial) and channels of visual information (focal vs. ambient).

2.4 Seminal Works, Findings and Guidelines

Multimodal interfaces emerged approximately 30 years ago within the field of human/computer interaction with Richard Bolt's "Put-That-There" application [9], which was created in 1980. First multimodal systems sought ways to go beyond the standard interaction mode at this time, which was graphical interfaces with keyboards and mice. Bolt's "Put-that-there" processed spoken commands linked to a pointing gesture using an armrest-mounted touchpad to move and change shapes displayed on a screen in front of the user. Since this seminal work, multimodal interaction practitioners have strived to integrate more modalities, to refine hardware and software components, and to explore limits and capabilities of multimodal interfaces. Historically, the main trend has focused on pointing and speech combined using speech/mouse, speech/pen [17], speech/gesture [45], or speech/gaze tracking [31]. Later multimodal interfaces evolved beyond pointing into richer interaction, allowing users to produce symbolic gestures such as arrows and encircling.

Another direction in multimodal research has been speech/lip movement integration [57][12], driven by cognitive science research in intersensory audio-visual perception. This kind of work has included classification of human lip movement (visemes) and the viseme-phoneme mappings that occur during articulated speech. Such work has contributed improving robustness of speech recognition in noisy environments. For more details about these systems, see [8].

Table 2. 10 myths of multimodal interaction (We acknowledge ACM for allowing the reprint of this table)

Myth #1: *If you build a multimodal system, users will interact multimodally.*
Myth #2: *Speech and pointing is the dominant multimodal integration pattern.*
Myth #3: *Multimodal input involves simultaneous signals.*
Myth #4: *Speech is the primary input mode in any multimodal system that includes it.*
Myth #5: *Multimodal language does not differ linguistically from unimodal language.*
Myth #6: *Multimodal integration involves redundancy of content between modes.*
Myth #7: *Individual error-prone recognition technologies combine multimodally to produce even greater unreliability.*
Myth #8: *All users' multimodal commands are integrated in a uniform way.*
Myth #9: *Different input modes are capable of transmitting comparable content.*
Myth #10: *Enhanced efficiency is the main advantage of multimodal systems.*

In the course of the last decade, researchers have highlighted particular empirical findings that have guided the design of multimodal interfaces compared to other sorts of human-computer interfaces. Key findings are illustrated in the following "10 myths" shown in Table 2, which exposed common engineering myths regarding how people interact multimodally [52]. Based on empirical findings, Oviatt distilled implications for how more effective multimodal interfaces could be designed.

In more recent years, research has also focused on mainstreaming multimodal interfaces. In this trend, Reeves et al. defined the following "guidelines for multimodal user interface design" [59]:

- Multimodal systems should be designed for the broadest range of users and contexts of use, since the availability of multiple modalities supports flexibility. For example, the same user may benefit from speech input in a car, but pen input in a noisy environment.
- Designers should take care to address privacy and security issues when creating multimodal systems: speech, for example, should not be used as a modality to convey private or personal information in public contexts.
- Modalities should be integrated in a manner compatible with user preferences and capabilities, for example, combining complementary audio and visual modes that users can co-process more easily.
- Multimodal systems should be designed to adapt easily to different contexts, user profiles and application needs.
- Error prevention and handling is a major advantage of multimodal interface design, for both user- and system-centered reasons. Specific guidelines include integrating complementary modalities to improve system robustness, and giving users better control over modality selection so they can avoid errors.

3 Principles of User-Computer Multimodal Interaction

The driving principles of multimodal interaction are well described in numerous surveys[8][26][51][54][62]. The following concepts are popularly accepted: fusion (also called multimodal signal integration), fission (also called response planning), dialog management, context management and time-sensitive architectures. In the following subsections, we introduce these concepts, at a high level first to illustrate how they are organized around a common conceptual architecture, and later at a lower level to probe key principles.

3.1 Theoretical Principles

Inspired by Norman's action cycle [47], and based on well accepted findings and taxonomies, the following model of multimodal man-machine communication can be drawn, together with the major concepts that should be considered when building a multimodal system (Figure 1): the fusion of multimodal inputs, and the multimodal fission to generate an adequate message to the user, according to the context of use, preferences and profile.

When a human interacts with a machine, his communication can be divided in four different states. The first state is a *decision state*, in which the communication message content is prepared consciously for an intention, or unconsciously for attentional content or emotions. The second state is the *action state*, where the communication means to transmit the message are selected, such as speech, gestures or facial expressions. The machine, in turn, will make use of a number of different modules to grasp the most information possible from a user, and will have similarly four main states

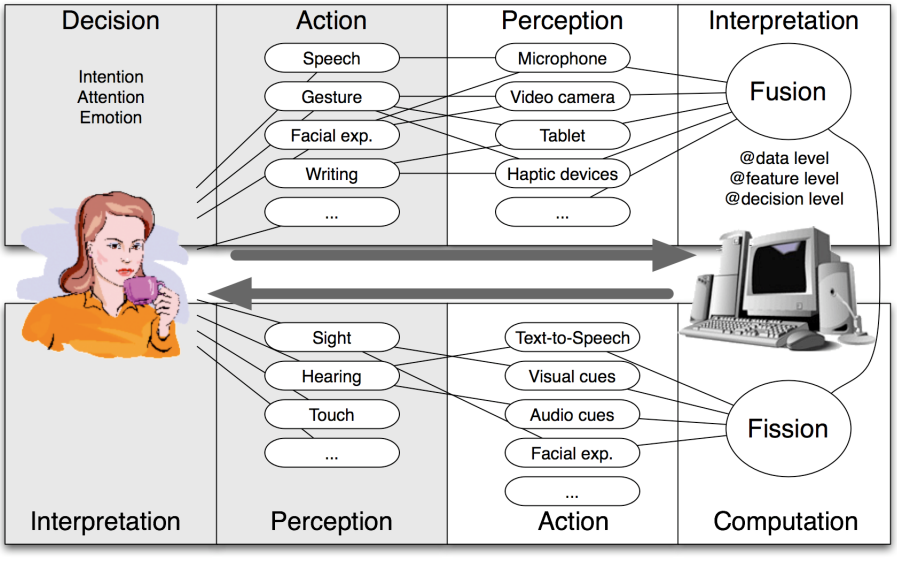

Fig. 1. A representation of multimodal man machine interaction loop

(Figure 1). At first, the messages are interpreted in the *perception state*, where the multimodal system receives information from one or multiple sensors, at one or multiple levels of expression. In the *interpretation state*, the multimodal system will try to give some meaning to the different information it collected in the perception state. This is typically the place where fusion of multimodal messages takes place. Further, in the *computational state*, action is taken following the business logic and dialogue manager rules defined by the developer. Depending on the meaning extracted in the interpretation state, an answer is generated and transmitted in the *action state*, in which a fission engine will determine the most relevant modalities to return the message, depending on the context of use (e.g. in the car, office, etc.) and the profile of the user (blind user, elderly, etc.).

3.2 Computational Architecture and Key Components

The previous section illustrated multimodal man-machine interaction underlying features. In this section, we describe multimodal interaction from the machine side, and the major software components that a multimodal system should contain. The generic components for handling of multimodal integration are: a fusion engine, a fission module, a dialog manager and a context manager, which all together form what is called the "integration committee". Figure 2 illustrates the processing flow between these components, the input and output modalities, as well as the potential client applications. As illustrated in the figure, input modalities are first perceived though various recognizers, which output their results to the *fusion engine*, in charge of giving a common interpretation of the inputs. The various levels at which recognizers' results can be fused are described in the next section, together with the various

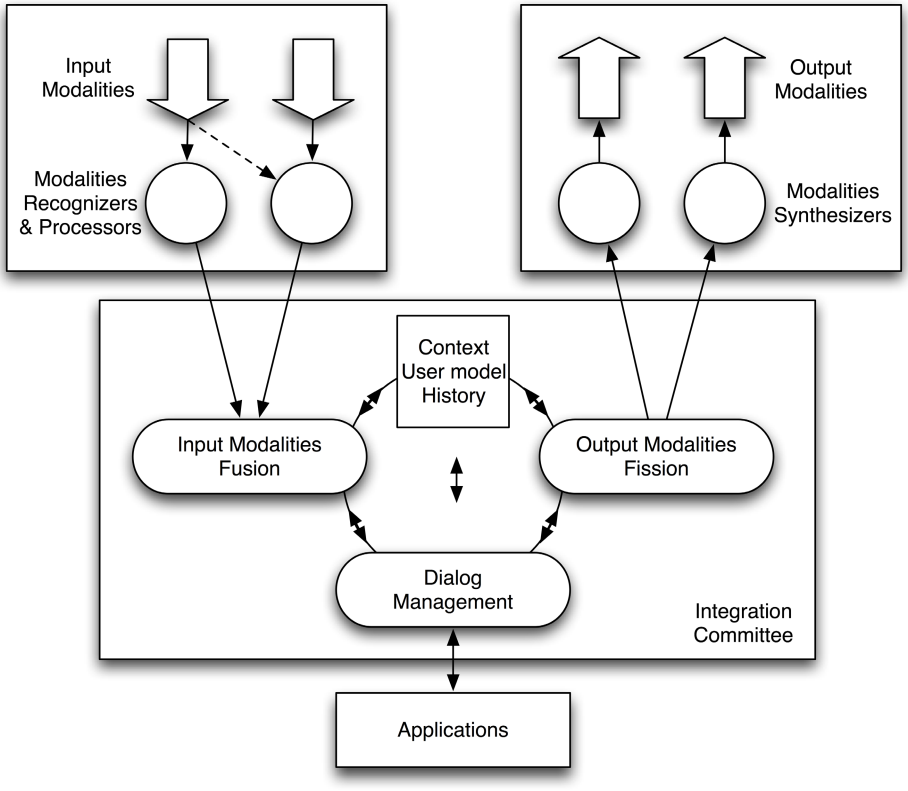

Fig. 2. The architecture of a multimodal system, with the central integration committee and its major software components

fusion mechanisms. When the fusion engine comes to an interpretation, it communicates it to the *dialog manager,* in charge of identifying the dialog state, the transition to perform, the action to communicate to a given application, and/or the message to return through the *fission component*. The fission engine is finally in charge of returning a message to the user through the most adequate modality or combination of modalities, depending on the user profile and context of use. For this reason, the *context manager*, in charge of tracking the location, context and user profile, closely communicates any changes in the environment to the three other components, so that they can adapt their interpretations.

3.3 Fusion of Input Modalities

Fusion of input modalities is one of the features that distinguish multimodal interfaces from unimodal interfaces. The goal of fusion is to extract meaning from a set of input modalities and pass it to a human-machine dialog manager. Fusion of different modalities is a delicate task, which can be executed at three levels: at data level, at feature level and at decision level. Three different types of architectures can in turn

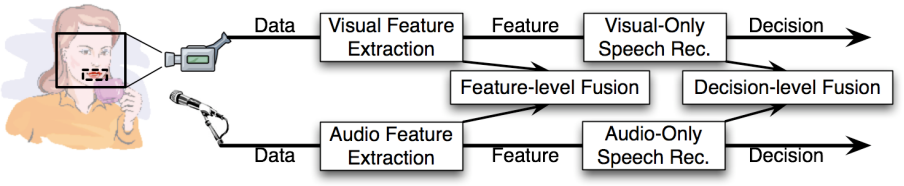

Fig. 3. The various levels of multimodal fusion

manage decision-level fusion: frames-based architectures, unification-based architectures or hybrid symbolic/statistical fusion architectures.

Sharma et al. [62] consider these three levels for fusion of incoming data. Each fusion scheme functions at a different level of analysis of the same modality channel. As a classic illustration, consider the speech channel: data from this channel can be processed at the audio signal level, at the phoneme (feature) level, or at the semantic (decision) level (Figure 3).

- *Data-level fusion* is used when dealing with multiple signals coming from a very similar modality source (e.g., two webcams recording the same scene from different viewpoints). With this fusion scheme, no loss of information occurs, as the signal is directly processed. This benefit is also the main shortcoming of data-level fusion. Due to the absence of pre-processing, it is highly susceptible to noise and failure.
- *Feature-level fusion* is a common type of fusion when tightly-coupled or time synchronized modalities are to be fused. The standard example is the fusion of speech and lip movements. Feature-level fusion is susceptible to low-level information loss, although it handles noise better. The most classic architectures used for this type of fusion are adaptive systems like artificial neural networks, Gaussian mixture models, or hidden Markov models. The use of these types of adaptive architecture also means that feature-level fusion systems need numerous data training sets before they can achieve satisfactory performance.
- *Decision-level fusion* is the most common type of fusion in multimodal applications. The main reason is its ability to manage loosely-coupled modalities like, for example, pen and speech interaction. Failure and noise sensitivity is low with decision-level feature, since the data has been preprocessed. On one hand, this means that decision-level fusion has to rely on the quality of previous processing. On the other hand, unification-based decision-level fusion has the major benefit of improving reliability and accuracy of semantic interpretation, by combining partial semantic information coming from each input mode which can yield "mutual disambiguation" [49].

Table 3 below summarizes the three fusion levels, their characteristics, sensitivity to noise, and usage contexts.

Table 3. Characteristics of fusion levels

	Data-level fusion	Features-level fusion	Decision-level fusion
Input type	Raw data of same type	Closely coupled modalities	Loosely coupled modalities
Level of information	Highest level of information detail	Moderate level of information detail	Mutual disambiguation by combining data from modes
Noise/failures sensitivity	Highly susceptible to noise or failures	Less sensitive to noise or failures	Highly resistant to noise or failures
Usage	Not really used for combining modalities	Used for fusion of particular modes	Most widely used type of fusion
Application examples	Fusion of two video streams	speech recognition from voice and lips	Pen/speech interaction

Typical architectures for decision-level fusion are frame-based fusion, unification-based fusion and hybrid symbolic/statistical fusion.

- *Frame-based fusion* [70] uses data structures called frames or features for meaning representation of data coming from various sources or modalities. These structures represent objects as attribute-value pairs.
- *Unification-based fusion* [27] is based on recursively merging attribute-value structures to obtain a logical whole meaning representation.
- *Symbolic/statistical fusion* [74] is an evolution of standard symbolic unification-based approaches, which adds statistical processing techniques to the fusion techniques described above. These kinds of "hybrid" fusion techniques have been demonstrated to achieve robust and reliable results. An example of a symbolic-statistical hybrid fusion technique is the Member-Team-Committee (MTC) architecture used in Quickset [75].

3.4 Fission of Output Modalities

When multiple output modalities such as text-to-speech synthesis, audio cues, visual cues, haptic feedback or animated agents are available, output selection becomes a delicate task to adapt to a context of use (e.g. car, home, work), type of task (e.g., information search, entertainment) or type of user (e.g. visually impaired, elderly).

Fission techniques [23] allow a multimodal application to generate a given message in an adequate form according to the context and user profiles. Technically speaking, fission consists of three tasks:

- Message construction, where the information to be transmitted to the user is created; approaches for content selection and structuring revolve mainly around either schema-based approaches or plan-based approaches [40, 43].

- Output channel selection, where interfaces are selected according to context and user profile in order to convey all data effectively in a given situation. Characteristics such as available output modalities, information to be presented, communicative goals of the presenter, user characteristics and task to be performed are forms of knowledge that can be used for output channel selection [2, 3].
- Construction of a coherent and synchronized result: when multiple output channels are used, layout and temporal coordination are to be taken into account. Moreover, some systems will produce multimodal and cross-modal referring expressions, which will also have to be coordinated.

3.5 Dialogue Management and Time-Sensitive Architectures

The time constraint is highly important in multimodal systems and all the modalities should be properly time-stamped and synchronized. Time-sensitive architectures need to establish temporal thresholds for time-stamping start and end of each input signal piece, so that two commands sequences can be identified. Indeed, when two commands are performed in parallel, in a synergistic way, it is important to know in which order the commands have been entered because the interpretation will vary accordingly. For instance, in the following application, in which voice and gestures are used simultaneously to control a music player, depending on the order in which modalities are presented the interpretation varies:

- <pointing> “Play next track”: will result in playing the track following the one selected with a gesture;
- “Play” <pointing> “next track”: will result in first playing the manually selected track and then passing to the following at the time “next is pronounced”;
- “Play next track” <pointing>: In this case, the system should interpret the commands as being redundant.

The dialog management system and synchronization mechanism should consider multiple potential causes of lag:

- delay due to technology (e.g. speech recognition);
- delay due to multimodal system architecture;
- user differences in habitual multimodal integration pattern [51][55].

For this reason, multi-agent architectures (or similar architectures such as components-based systems) are advantageous for distributing processing and for coordinating many system components (e.g., speech recognition, pen recognition, natural language processing, graphic display, TTS output, application database).

Bui [13] considers four different approaches to dialog management:

- *Finite-state and frame-based approaches*: in this kind of dialog management approach, the dialog structure is represented in the form of a state machine. Frame-based models are an extension of finite-state models, using a slot-filling strategy in which a number of predefined information sources are to be gathered [16].
- *Information state-based and probabilistic approaches*: these approaches try to describe human-machine dialog following information states, consisting of five main

components: informational components, formal representations of those components, a set of dialog moves, a set of update rules and an update strategy [68].

- *Plan-based approaches*: the plan-based approaches are based on the plan-based theories of communicative action and dialog [16]. These theories claim that the speaker's speech act is part of a plan and that it is the listener's job to identify and respond appropriately to this plan [15].
- *Collaborative agents-based approaches*: these approaches view dialog as a collaborative process between intelligent agents. The agents work together to obtain a mutual understanding of the dialog. This induces discourse phenomena such as clarifications and confirmations [48].

3.6 Machine Learning for Multimodal Interaction

Machine learning techniques play an important role in multimodal interfaces [26], and most certainly will continue to extend this role. Indeed, many parts of multimodal systems are likely to receive support from machine learning. Modality recognizers already make extensive use of machine learning: speech recognition, face detection, face recognition, facial expression analysis, gesture recognition or eye tracking are examples of different domains of interest both for multimodal interaction and machine learning.

Aside from modality handling, machine learning has been applied for fusion of input recognizers' data, mainly at the feature level. Fewer works have been achieved on decision level fusion with assistance from machine learning. An example of such work is Pan et al. [55], who proposed context-dependent versions of Bayesian inference method for multisensory data fusion. Nonetheless, Jaimes & Sebe [26] reckon that "*further research is still required to investigate fusion models able to efficiently use the complementary cues provided by multiple modalities*". User, task and context modeling also can benefit from machine learning techniques. Novel research fields related to machine learning, such as social signal processing [64], will help building a refined representation of the user in her collaborative context. Adaptability can then be addressed with the help of machine learning, by watching the users' behavior in the sensed context [21].

As Jaimes & Sebe [26] highlight, currently "*most researchers process each channel (visual, audio) independently, and multimodal fusion is still in its infancy*". Thus, multimodal interaction researchers have work to achieve in order to attain efficient multimodal fusion, with careful consideration of the different available modalities and the way modalities interlock. Machine learning will be of interest in order to attain such a goal. Besides multimodal fusion, machine learning will help multimodal applications take into account the affective aspect of communication – emotions based on their physiological manifestations [41], such as facial expressions, gestures, postures, tone of voice, respiration, etc.

4 Modeling Languages and Frameworks

There have been several attempts to model and formalize multimodal interaction. This section presents several different levels of modeling. The first part introduces two

abstract models designed to help developers evaluate the different types of multimodal interaction, viewed first from the machine side, then from the user side. The second part lists a number of languages used for multimodal recognizer output and multimodal synthesizer input representations, and modeling languages used to configure multimodal systems. The final part displays different programming frameworks for rapid creation of multimodal interfaces.

4.1 Multimodal Interaction Modeling

Modeling multimodal interaction is no simple task, due to the multiple input and output channels and modes, and the combination of possibilities between data coming from different sources, not to mention output modality selection based on context and user profile.

The shape taken by formal modeling of multimodal interaction depends on the level of abstraction considered. At lower levels of abstraction, formal modeling would focus on tools used for modality recognition and synthesis. At higher levels of abstraction, multimodal interaction modeling would focus more on modality combination and synchronization.

Formal modeling can also focus on the "pure" technical part as well as on the user-machine interaction. Two formal models exist for modality combination description:

- The CASE model [46], focusing on modality combination possibilities at the fusion engine level;
- the CARE model [18], giving attention to modality combination possibilities at the user level.

The CASE model introduces four properties: Concurrent – Alternate – Synergistic – Exclusive (figure 4). Each of those four properties describes a different way to combine modalities at the integration engine level, depending on two factors: combined or independent fusion of modalities, and sequential or synergistic use of modalities on the other hand. "Fusion of modalities" considers if different modalities are combined or managed independently, whereas "Use of modalities" observes the way modalities are activated: either one at a time, or in a synergistic manner.

The CARE model is more focused on the user-machine interaction level. This model also introduces four properties, which are Complementarity – Assignment – Redundancy – Equivalence. Complementarity is to be used when multiple complementary modalities are necessary to grasp the desired meaning (e.g. "put that there" [9] would need both pointing gestures and voice in order to be resolved). Assignment indicates that only one modality can lead to the desired meaning (e.g. the steering wheel of a car is the only way to direct the car). Redundancy implies multiple modalities which, even if used simultaneously, can be used individually to lead to the desired meaning (e.g. user utters a "play" speech command and pushes a button labeled "play", but only one "play" command would be taken into account). Finally, Equivalence entails multiple modalities that can all lead to the desired meaning, but only one would be used at a time (e.g. speech or keyboard can be used to write a text).

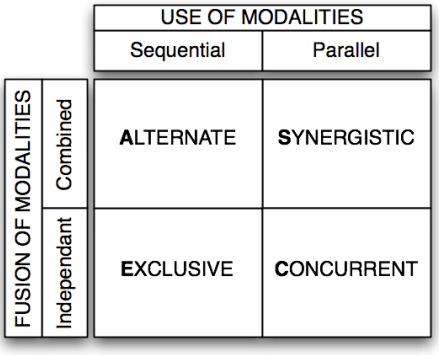

Fig. 4. The CASE model

4.2 Multimodal Interaction Modeling Languages

Interesting attempts at creating a full-fledged language for description of user-machine multimodal interaction have arisen in the past few years. Most of the approaches presented below revolve around the concept of a "multimodal web", enforced by the World Wide Web Consortium (W3C) Multimodal Interaction Activity and its proposed multimodal architecture [71]. This theoretical framework describes major components involved in multimodal interaction, as well as potential or existent markup languages used to relate those different components. Many elements described in this framework are of practical interest for multimodal HCI practitioners, such as the W3C EMMA markup language, or modality-focused languages such as VoiceXML or InkML. The work of the W3C inspired Katsurada et al. for their work on the XISL XML language [28]. XISL focuses on synchronization of multimodal input and output, as well as dialog flow and transition. Another approach of the problem is the one of Araki et al. [4], who propose MIML (Multimodal Interaction Markup Language). One of the key characteristics of this language is its three-layered description of interaction, focusing on interaction, tasks and platform. Finally, Stanciulescu et al. [64] followed a transformational approach for developing multimodal web user interfaces based on UsiXML, also in the steps of the W3C. Four steps are achieved to go from a generic model to the final user interface. Thus, one of the main features of their work is a strong independence to the actual input and output available channels.

Sire and Chatty describe in [63] what one should expect from a multimodal user interfaces programming language. From their proposal, the following requirements for a multimodal description language have been derived.

- Such a language should be *modality agnostic*, as research in input and output modalities continues to evolve today.
- A *binding mechanism to link the definition of the user interface composition with its runtime realization* should be provided.
- *Explicit control structures* should be present, such as conditional clauses and loops.
- *Extensible event definition mechanisms* are also needed for communication between user interface objects and the interaction model.

- *Data Modeling* should be carefully planned, as application data tends to be distributed in multiple places.
- Finally, a major requirement for a multimodal integration description language is the definition of *reusable components*.

"Modality agnostic" is the most debatable of those requirements, as one could argue that such a requirement will never be achievable, as every modality has its own particularities. Our interpretation of this requirement is the following: "modality agnostic" means that the language should not be specific for each individual modality, as modalities are all different; the language should be flexible enough (or canonic enough) to be adapted to a new and different modality. Hence, if a scripting or programming language can be in principle modality agnostic, such cannot be said of the fusion engine that needs to take into account the specificities of each modality to fuse data or features correctly.

A last point that stems from these six guidelines is *readability*: a language for description of multimodal interaction should be readable, as much in regard to the machine as to humans.

Formal languages for description of multimodal description can be approached from two different directions: either from expressiveness, or from usability. Expressiveness covers technical features such as extensibility, completeness, reusability, or temporal aspects considerations; usability covers more human features such as programmability or readability. Any formal language will have to find its place between those two general requirements; some languages will tend more toward expressiveness or usability. An interesting approach is to seek balance between usability and expressiveness: that is, a language able to configure a multimodal system, with high level modeling, and readable enough to be used as a learning tool, or even a communication tool.

4.3 Programming Frameworks

Further to multimodal interface creation, a number of tools have become available in recent years. Krahnstoever et al. [32] proposed a framework using speech and gestures to create a natural interface. The output of their framework was to be used on large screen displays enabling multi-user interaction. Fusion was done using a unification-based method. Cohen et al. [17] worked on Quickset, a speech/pen multimodal interface, based on Open Agent Architecture, which served as a test bed for unification-based and hybrid fusion methods. Bourguet [11] endeavored in the creation of a multimodal toolkit in which multimodal scenarios could be modelled using finite state machines. This multimodal toolkit is composed of two components, a graphical user interface named IMBuilder which interfaces the multimodal framework itself, named MEngine. Multimodal interaction models created with IMBuilder are saved as a XML file. Flippo et al. [22] also worked on the design of a multimodal framework, geared toward direct integration into a multimodal application. One of the most interesting aspects of their work is the use of a parallel application-independent fusion technique. The general framework architecture is based on agents, while the fusion technique itself uses frames. Configuration of the fusion is done via an XML file, specifying for each frame a number of slots to be filled and direct link to actual resolver implementations. Lastly, Bouchet et al. [10] proposed a component-based approach called

Table 4. Characteristics of different tools for creation of multimodal interfaces

	ICARE – OI [10]	OpenInterface [61]	IMBuilder/ MEngine [11]	Flippo et al. [22]	Krahnstoever [32]	Quickset [17]	Phidgets [25]	Papier-Mâché [30]
Architecture traits								
Finite state machine			x					
Components	x	x					x	
Software agents				x		x		
Fusion by frames					x			
Symbolic-statistical fusion						x		
Reusability easiness								
No programming kit					x	x		
Low-level programming (e.g. via API)				x			x	x
Higher-level Programming								
Visual Programming tool	x	x	x					
Characteristics								
Extensibility		x	x	x		x		
Pluggability							x	
Reusable components	x	x				x		
Open Source	x	x						x

ICARE thoroughly based on the CARE [18] design space. These components cover elementary tasks, modality-dependent tasks or generic tasks like fusion. Finally, communication between components is based on events. The components-based approach of ICARE has provided inspiration for a comprehensive open-source toolkit called OpenInterface [61]. OpenInterface components are configured via CIDL XML files, and a graphical editor.

Table 4 summarizes the different characteristics of the systems described above: extensible systems (i.e. toolkits) have the potential ability to add other input modalities in a practical way. Pluggability refers to the ability of a toolkit to insert itself into an architecture without having to rewrite everything. The other characteristics are self-explanatory.

5 Multimodal Interfaces in Switzerland

5.1 Multimodal Interfaces in IM2

The Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2) is one of the 20 Swiss National Centers of Competence in Research (NCCR). IM2 aims at developing natural multimodal interfaces for human-computer interaction and to foster collaboration, focusing on new

multimodal technologies to support human interaction, in the context of smart meeting rooms and remote meeting assistants.

The Individual Project on "Human Machine Interaction" is part of the NCCR IM2. While other activities in IM2 develop multimodal analysis and recognition technologies, the primary objective of IM2.HMI is to build cutting-edge technologies to develop interactive multimodal meeting browsers. The main goal of IM2.HMI is to design, develop and evaluate, with human subjects, novel interactive multimodal meeting browsers/assistants.

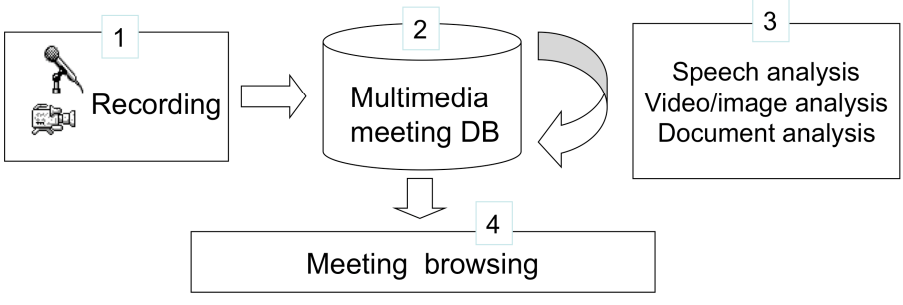

Fig. 5. Multimodal processing chain in IM2 meeting application

In order to support the development of so-called meeting browsers (4), and facilitate access to multimodal data and annotations (2), the JFerret framework has been designed and implemented. Using the JFerret framework, and taking benefits of most of the multimodal analysis, multimodal input recognizers and multimodal indexing and retrieval strategies made available in IM2, various meeting browsers have been implemented [33]. Those meeting browsers take benefit of most of the annotations made available by the other IM2 IPs: speech browsers (accelerated and overlapped), document-centric meeting browsers (JFriDoc, FaericWorld) [60], Dialog-centric browsers (TQB) [58], multimodal enabled browsers (Archivus, HephaisTK), multilingual (M3C) and recently personalized browsers (WotanEye) [34]. Most of these meeting browsers are in fact complete and transversal systems that access the multimodal meeting data, analyse them, process high level indexes and provide interactive user interfaces so that the user can browse the meeting corpora through multimodal queries. In the last couple of years, IM2.HMI has gently shifted towards online, a.k.a real-time, meeting assistance leveraging on past works. This includes new research on personalized meeting browsing, mobile and remote access to meetings [38], and meeting assistance before, during and after meetings.

IM2.HMI has tackled multimodality both at the content and at the interaction levels. While projects handling multimodality at the content level try to use the best of multimodal data indexing in order to create useful and usable meeting browsers, research projects handling multimodality at the interaction level study and build novel multimodal interaction paradigms, benefiting from various input modes.

Archivus, developed in the framework of IM2, is a good example of a research project handling multimodality both at the content and interaction levels. Archivus is a multimodal (pen, voice, mouse and keyboard) language-enabled dialogue-based

interface for browsing and retrieving multimodal meeting data [1]. It allows users to access a multimedia database of recorded and annotated meetings, containing the original video and audio streams, electronic copies of all documents used or referred to as well as handwritten notes made by participants during the meeting, and a text transcript of the meeting itself [37, 42]. Multimodal man-machine interaction in this context has been carefully studied. Large-scale Wizard of Oz experiments with the system (involving 91 users) were carried out and it resulted in 180 hours of video data and 70MB of text log files. The data was analyzed along several different lines including the modalities most often used, contexts of use, relationships between modalities, usage change over time, training impact, etc. [36]. To summarize the major findings: exposure and training can have a strong impact on the way people use multimodality, and speech is a preferred modality both at the content and interaction levels, i.e. as a cue for querying the multimodal database and as an interaction channel.

HephaisTK, developed both in the framework of the NCCR IM2 and of the MeModules project presented in chapter 5, handles multimodality at the interaction level and aims at providing a tool allowing developers to easily prototype multimodal interfaces [20]. The HephaisTK toolkit has been designed to plug itself in a client application that wishes to receive notifications of multimodal events received from a set of modality recognizers. It is based on a software agents architecture, in which agents, collaborating through a blackboard, are dispatched to manage individual modality recognizers, handle fusion and dialog management. HephaisTK can be configured with the SMUIML language (*Synchronized Multimodal User Interfaces Markup Language*) [19], allowing a clear description of the human-machine multimodal dialog and control over the way multiple input modalities have to be fused. More details about this tool can be found in chapter 5 of this book.

5.2 Multimodal Interfaces in the MMI Program

The IM-HOST project, described in detail in chapter 4 of this book, is representative of one class of multimodal applications, although it focuses on a single modality: speech, which has been historically the leading modality in multimodal interaction. The IM-HOST project targets voice-enabled man-machine interaction in noisy environments. However, still, current performances of voice applications are reasonably good in quiet environments but the surrounding noise in many practical situations drastically deteriorates the quality of the speech signal and, as a consequence, significantly decreases the recognition rate. The major scenario considered in this project is a person using voice command in an outdoor environment: a racing boat. For this reason, the project explores new interaction paradigms enabling voice recognition in a hostile environment.

The MeModules project, fully detailed in chapter 5 of this book, has the objective of developing, experimenting and evaluating the concept of tangible shortcuts to multimedia digital information. Moreover, it investigates the opportunity of a more complex, multi-sensorial combination of physical objects with multimedia information by associating tangible interaction with multiple other interaction modalities such as voice, gesture, etc. One of the expected research outcomes of the project is to assess which modalities are best combined with tangible interaction depending on the context and application.

6 Future Directions and Conclusions

Although many issues have been addressed well in the multimodal interaction research and systems literature, such as fusion of heterogeneous data types, architectures for real-time processing, dialog management, map-based multimodal interaction, and so forth, nonetheless the field is still young and needs further research to build reliable multimodal systems and usable applications. Machine learning methods have begun to be applied to a number of different aspects of multimodal interfaces, including individual modality recognition, early or late modality fusion, user-machine dialog management, and identification of users' multimodal integration patterns. But future work clearly is needed to work toward the design of usable adaptive multimodal interfaces. Multimodal dialog processing also will gain in the future from the recent and promising subfield of social signal processing, which can assist dialog modeling by providing a dialog manager with real-time information about a given user's state and her current social and collaborative context.

Other important future directions for multimodal research include human/machine interaction using new tangible interfaces such as digital paper and pen, and multitouch tables, surfaces and screens. Further modeling of multimodal interaction still is needed too, in areas such as multimodal educational exchanges, collaborative multimodal interaction, multimodal interaction involving diverse and underserved user groups, and mobile multimodal interaction with emerging cell phone applications. Finally, further work is needed to improve tools for the creation of multimodal applications and interfaces so they can become more mainstream, especially since multimodal interfaces are viewed as the most promising avenue for achieving universal access in the near future.

References

1. Ailomaa, M., Lisowska, A., Melichar, M., Armstrong, S., Rajmanm, M.: Archivus: A Multimodal System for Multimedia Meeting Browsing and Retrieval. In: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, Sydney, Australia, July 17th-21st (2006)
2. Allen, J.F., Perault, C.R.: Analyzing Intentions in Dialogues. Artificial Intelligence 15(3), 143–178 (1980)
3. André, E.: The generation of multimedia documents. In: Dale, R., Moisl, H., Somers, H. (eds.) A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text, pp. 305–327. Marcel Dekker Inc., New York (2000)
4. Araki, M., Tachibana, K.: Multimodal Dialog Description Language for Rapid System Development. In: Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue (July 2006)
5. Baddeley, A.D.: Working Memory. Science 255, 556–559 (1992)
6. Baddeley, A.D.: Working Memory. In: Bower, G.A. (ed.) Recent advances in learning and motivation, vol. 8. Academic Press, New York (1974)
7. Arens, Y., Hovy, E., Vossers, M.: On the knowledge underlying multimedia presentations. In: Maybury, M.T. (ed.) Intelligent Multimedia Interfaces, pp. 280–306. AAAI Press, Menlo Park (1993); Reprinted in Maybury and Wahlster, pp. 157–172 (1998)