THE MSM FRAMEWORK:
A DESIGN SPACE FOR MULTI-SENSORY-MOTOR SYSTEMS

Joëlle Coutaz, L. Nigay, Daniel Salber
Laboratoire de Génie Informatique, Institut IMAG, Grenoble, France

Jean Caelen
Institut de la Communication Parlée, INPG, Grenoble, France

Contact person:
Joëlle Coutaz, LGI-IMAG, BP 53 X, 38041 Grenoble Cedex, France
E-mail: coutaz@imag.fr, tel. +33 76 51 48 54, fax: + 33 76 44 66 75

INTRODUCTION

Parallel to the development of the GUI technology, natural language processing, computer vision, 3-D sound, and gesture recognition have made significant progress. Artificial and virtual realities are good examples of systems that aim to integrate these diverse interaction techniques. Their goal is to extend the sensory-motor capabilities of computer systems to better match the natural communication means of human beings.

The sensory-motor abilities of systems may be augmented with various degrees of sophistication. This extension may range from the construction of new input/output devices to the definition and management of symbolic representations for the information communicated through such devices. The span of possibilities and the novelty of the endeavour explain the variety of the terms, such as multimedia and multimodal, used to qualify these systems. As demonstrated by our framework, multimedia and multimodal systems are both “multi-sensory-motor” (MSM) systems but with distinct degree of built-in cognitive sophistication.

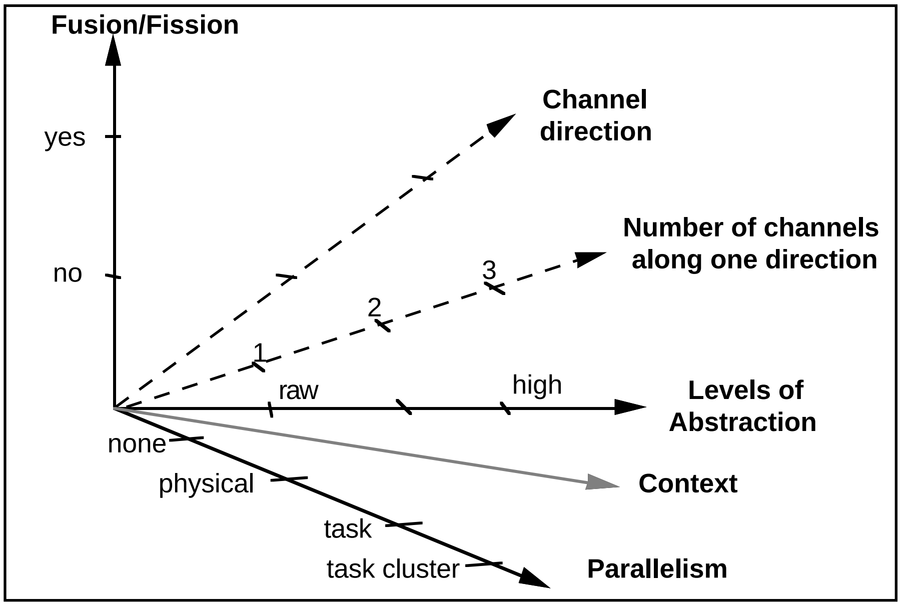

Figure 1: The MSM framework: A 6-D space to characterize multi-sensory-motor interactive systems.

This position paper proposes a framework that shoulp help reasoning about current and future MSM systems. It is a refinement of the dimension space presented in [Coutaz 92, Bourguet 92a, Nigay 93]. To do so, we adopt a system centered perspective although we draw upon the ICS psychological model [Barnard 87]. As shown in Figure 1, our framework is comprised of 6 dimensions:

- The first two dimensions deal with the notion of communication channel: the number and direction of the channels that a particular MSM system supports. Issues related to communication channels and the symetry with ICS are presented in the next section.
- The other four dimensions are used to characterize the degree of built-in cognitive sophistication of the system: levels of abstraction, context, fusion/fission, and granularity of concurrency. These issues are discussed in detail before we comment on the distinction between multimedia and multimodal interactive systems.

COMMUNICATION CHANNELS

A communication channel can be viewed as the temporal, virtual, or physical link that makes the exchange of information possible between communicating entities (e.g., a human being and a computer system). Instead of considering the linkage dimension of communication, we stress the importance of the sources and recipients involved in a communication act. Thus, a communication channel covers a set of sensory (or effector) means through which particular types of information can be received (or transmitted) and processed.

A sensor is a physical device that allows a communicating entity to acquire information from the environment (e.g., another communicating entity). An effector plays the symetrical role for transmitting information to the environment. Interestingly, sensors and effectors are not insulated randomly. Multiple sensors (effectors) may be grouped together to form a cluster associated to a processing facility.This grouping of physical devices under the hat of a processing unit corresponds to a communication channel. This view of a communication channel matches nicely the ICS psychological model.

For example, the retinas capture space-time patterns of photons which are processed by the visual subsystem into a mental form usable by the representational or effector subsystems. The retinas (which are two input physical devices) and the visual subsystem (which is the corresponding processing facility) define a *human communication channel*. As an example from the computer side, the X window server handles both mouse and keyboard input devices. It transforms interrupt signals into a higher level representation, an “X event”, that may be of interest to client processes. The keyboard, the mouse, and the X server define a *digital communication channel.*

Information types conveyed by human and digital communication channels define an abstraction from the physical representations used by I/O devices. This abstraction is the boundary with higher internal representations. It conveys phenomena, not meaning. Meaning is covered by the internal processes, responsible for executing the interpretation and rendering functions.

INTERPRETATION AND RENDERING

Information acquired by input digital channels is transformed through multiple process activities. This sequence of input transformations forms the interpretation function. In the other direction, internal information (e.g., system state) is transformed to be made perceivable to the user. This sequence of output transformations defines the rendering function. The interpretation and the rendering functions can be both characterized with four intertwined ingredients: level of abstraction, context, fusion/fission, and parallelism. These dimensions are presented in the following paragraphs.

Level of abstraction

The notion of level of abstraction expresses the degree of transformation that the interpretation and rendering functions perform on information. It also covers the variety of representations that the system supports, ranging from raw data to symbolic forms. The span of representations should be considered on a per-digital channel basis. Thus, for a given digital input channel, the interpretation function can be characterized by its power of “abstracting” raw data into higher representational expressions. The rendering function is characterized by the level of abstraction it starts from to produce perceivable raw information through output digital channels.

Computer vision, speech recognition as well as speech synthesis systems operate along these principles. For example, speech input may be recorded as a signal, or described as a sequence of phonemes, or interpreted as a meaningful parsed sentence. Each representation corresponds to a particular level of abstraction resulting from an interpretation function. For output, the process is similar: data may be produced from symbolic abstract representations or from a lower level of abstraction without any computational knowledge about meaning. For example, a vocal message may be synthesized from an abstract representation of meaning, from a pre-stored text (i.e., text-to-speech) or may simply be replayed from a previous recording.

Context

The capacity of a system to abstract along a channel may vary dynamically with respect to “contextual variables”. Contextual variables are like cognitive filters. They form a set of internal state parameters used by the representational processes to control the interpretation/rendering function. For example, in vi, when in command mode, typed text is transformed into a high level abstraction whereas the same text entered in input mode is recorded as is without any transformation. Contextual variables constrain the configuration of digital processes used at some point in time to process information. We observe an analogy with the cognitive resources configuration claimed in ICS.

In Figure 1, we have not provided salient values for the “context” dimension. We have however identified one discriminatory feature shared for input by current MSM systems: commands versus task domain data. We have observed that current MSM systems support high level interpretation in the context of commands but very little for task domain related data. The “vi” example mentionned above is one of many illustrations. Although the contextual variable “command/task domain data” may be of interest to characterize current systems, its scope is rather narrow. More work needs to be done to identify additional contextual variables that would be shared by most systems.

Fusion and Fission

Fusion refers to the combination of several chunks of information to form new chunks. Fission refers to the decomposition phenomenon. Fusion and fission are part of the abstracting and materialization phenomena.

The interpretation function and fusion

Considering fusion for the interpretation function:
- at the lowest level, information chunks may (or may not) originate from distinct digital input channels;
- at higher levels, information chunks may (or may not) come from distinct contexts.

For example, the sequence of events “mouse-down, mouse-up” that occurs in the palette of a graphics editor are two information chunks that originate from the same input channel and from the same context (i.e., the palette). They are combined within the context of the palette to form a higher information chunk (i.e., the selection of a geometric class). The drawing area constitutes another context. Events that occur in the drawing area are interpreted as the effective parameters of the geometric function. They are combined with the selected geometric class to complete the function call in the task-domain. Thus, in this example, fusion occurs between information chunks originating from the same digital channel but, as the interpretation proceeds at higher levels of abstraction, it also involves different contexts.

The “put that there” paradigm offers an example of fusion between chunks originating from distinct input digital channels. In this example, fusion is required to solve the coreferences expressed through distinct channels.

The interpretation function and fission

It may be the case that information coming from a single input channel or from a single context need to be decomposed in order to be understood at a higher level of abstraction.

For example, consider the utterance “show me the red circle in a new window”. This sentence, received through a single digital channel, references two domains of discourse: that of the graphics task (i.e., “the red circle”) and that of the user interface (i.e., “a new window”). In order to satisfy the request, the system has to decompose the sentence into two high level functions: “create a window” and “draw a red circle” in the newly created window.

The rendering function and fusion
The rendering function can perform fusion at multiple levels of abstraction. One of them, which takes place at the highest level of abstraction (i.e., the domain adaptor) has been discussed in [Coutaz 91]. At the lowest level, it appears as multiple information chunks rendered through a single output channel.

For example, the picture of a town may be combined to a graphical representation of the population growth. The notions of town and population which are handled by two different contexts within the internal processes of the system are combined at the lowest level and presented through a single output digital channel.

The rendering function and fission
Rendering may also incorporate fission at multiple levels of abstraction. The highest level has been discussed in [Coutaz 91]. At the lowest level, fission occurs when an information chunk gives birth to multiple representations whether it be through a single or multiple digital output channels.

For example, the notion of wall in our mobile robot system [Bass91] may be represented as a line or as a form on the screen. These distinct representations of the same concept use only one digital output channel. Alternatively, the spoken message “watch this wall!” along with a blinking red line on the screen uses two distinct output channels to denote the same wall.

Parallelism

Representation and usage of time is a complex issue. In our discussion, we are concerned with the role of time within the interpretation and rendering functions. How does time relate to levels of abstraction and contexts? How does it interfere with fusion and fission? Parallelism at the user interface may appear at multiple grains: at the physical level, at the task and task cluster levels.

Parallelism at the physical level
For input, the physical level corresponds to the user actions that can be received by input digital channels as an information chunk (e.g., an event). For example, a mouse click, a spoken utterance are information chunks. For output, the physical level denotes output primitives, that is the information chunks that can be produced by output digital channels in one burst. For example, a spoken message or the reverse video of an icon.

For input, parallelism at the physical level allows the user to trigger multiple input devices simultaneously. If these devices are organized along distinct channels, then the user sollicits multiple input digital channels in parallel. Similarly, physical parallelism for output may take the form of simultaneous outputs through distinct digital channels or may occur through a single channel. The fission example “watch this wall” associated with “the blinking red line”, requires parallelism at the physical level using multiple digital output channels.

Parallelism at the task level
From the system’s perspective, a task (i.e, an elementary task) cannot be decomposed further but in terms of physical actions. For input, an elementary task is usually called a command, that is, the smallest fusion/fission of physical user’s actions that changes the system state. For output, an elementary task is the set of output physical primitives used to express a system state change.

True parallelism at the command level allows the user to issue multiple commands simultaneously. It necessarily relies on the availability of parallelism at the physical level. Pseudo-parallelism at the command level, allows the user to build several commands in an interleaved way as in multithread dialogues. Then, parallelism at the physical level is not required.

Figure 2 illustrates all possible relationships between parallelism at the physical level, and fusion and fission, to form commands within the interpretation function. In 2-a, multiple simultaneous inputs from channel 1 must be dispatched into two higher contexts (e.g., agents) to build two distinct commands in parallel. For example, in MMM, two users may manipulate two physical mouses simultaneously to respectively modify the size and color of a shared rectangle. In configuration 2-b, simultaneous actions on distinct input channels must be combined to build a single command (as in the “put that there” paradigm). In 2-c, physical actions follow two independent paths. For example, the user may say “close top window” while moving a file icon in the trash. In this case, two independent commands must be built in parallel.

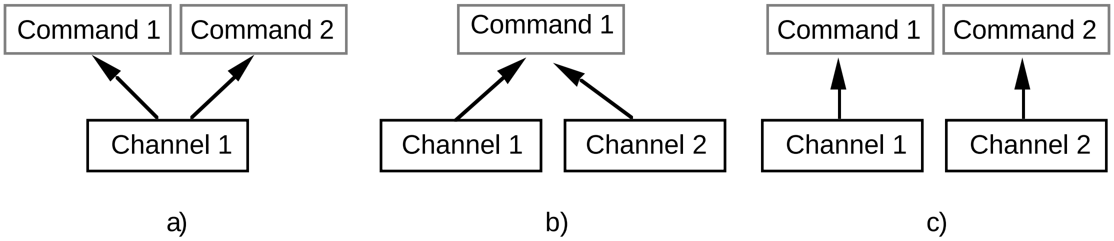

Figure 2: relationships between parallelism at the physical level, fusion and fission, and commands.

The diversity of the relationships shown in Figure 2 is a good indicator of the difficulty to implement the interpretation function. In particular, which criteria should be considered to trigger fusion and which strategy should be adopted? Early experiences with Matis [Nigay 92] and ICP-Plan [Bourguet 92b] show that temporal and structural proximities are valid criteria for fusion. Temporal proximity expresses parallelism between physical actions. (Due to the technological limitations of current speech recognition systems, mouse clicks are detected long before sentences are recognized, although expressed by the user at “the same time”.) Structural relationships expresses syntactic links between inputs. Thus, two inputs linked by temporal and structural properties are good candidates for fusion.

The strategy could be “eager” as opposed to “lazy”. Eager fusion makes attempts to combine inputs as soon as criteria are met at the lowest levels of abstraction. Lazy fusion postpones fusion to the highest levels. The advantage of eager fusion is the ability to generate early feedback. Its drawback is the necessity to be able to perform backtrackuing backtrack. This is particularly true when interleaving or parallelism is supported at the command level.

A similar analysis should be done for output elementary tasks. So far, we have not experienced enough exemplars to generate a sound discussion on this issue. However, we can relate two interesting examples. The first one is usage of time to synchronize information chunks over digital output channels. QuickTime is a good illustration of this capacity. The second example is interleaving between inputs and outputs at the task level. For example, as the system moves an object, the user may dynamically change the speed or the color of the object. We observe a temporal overlap between input and output at the task level. In this example, the duration of the system’s outputs covers the duration of a sequence of user’s commands and can be dynamically affected by these commands. In the other way round, rubber banding or reverse video of candidate recipients in the Macintosh finder are examples of duration of user’s inputs covered by system’s outputs.

Parallelism at the task cluster level

From the system’s perspective, a task is a cluster of tasks that structures the interaction space. For example, in our mobile robot system, the command space is organized into three subspaces: one for providing the robot with cartographic details, the second to specify missions to be accomplished, the third one to observe and control the robot during mission execution. For input, parallelism at the task level expresses how much parallelism (actually pseudo-parallelism) is supported by the system between clusters of commands. Note that parallelism at the cluster level does not necessarily imply parallelism at the command level.

For output, a similar organization in terms of clusters of parallelism may be observed. We have not studied this perspective yet.

MULTIMEDIA AND MULTIMODAL INTERACTIVE SYSTEMS

Both multimedia and multimodal systems are characterized by communicating information either through multiple input digital channels or through multiple output digital channels, or both. The multiplicity of communication channels along one direction (whether it be input or output) provides the basis for multimedia-lity and multimodality.

The distinction between multimedia-lity and multimodality lies in the degree of built-in cognitive sophistication of the system along the axis “level of abstraction”. Multimodality is characterized by the capacity of the system to interpret raw inputs up to high levels of abstraction (e.g., that of the task domain) or to render information starting from high level representations. Although multimedia-lity includes interpretation and rendering, it is not capable of handling the highest task-domain level representations.

An MSM system may be both multimedia and multimodal. For example, an hypermedia system would illustrate task-domain concepts using images and sound replayed from a CD-ROM, and it would be controlled by the user in a multimodal way using both speech and mouse to navigate through the hyper space. Note that current multimedia systems are all able to handle the highest task-domain level representations but they do so for commands only and through a unique channel. Thus any multimedia system is at least monomodal in order to recognize input commands.

SUMMARY

The analysis of the behavior of MSM interactive systems should be considered along the following dimensions:

- the nature of input and output physical devices and their grouping as digital input and output communication channels,
- the granularity of parallelism supported by the system along the input and output channels (i.e., physical actions, task level, task cluster),
- for each channel and context, and for combinaisons of input or output channels, the capacity of the system to support abstraction/materialization through the fusion and fission mechanisms.

As discussed in [Nigay 93], this framework can be used to classify current and future MSM systems. Although system centered, it draws upon psychology and HCI with the notions of communication channel, concurrency at the user interface, and levels of abstraction. By doing so, it identifies salient parameters for protocol studies as in Wizard of Oz experiments [Balbo 93]. In addition, it makes explicit issues, such as fission and fusion, that are relevant to the design of software architecture models and building tools. The MSM framework identifies properties of systems behavior that are currently being captured within a theory of interactors [Duke 92]. We would like this framework to serve as a starting point for a creative discussion during the workshop on “MM&MM Human-Computer Interfaces”.

ACKNOWLEDGMENTS

This work has been supported by project ESPRIT BR 7040 AMODEUS and by PRC Communication Homme-Machine.

REFERENCES

[Balbo 93] S. Balbo, J. Coutaz, D. Salber: Towards Automatic Evaluation of Multimodal User Interfaces; International Workshop on Intelligent User Interfaces, Orlando, USA, Jan., 1993.

[Barnard 87] P. Barnard, “Cognitive Resources and the Learning of Computer Dialogs", in Interfacing Thought, Cognitive aspects of Human Computer Interaction, J.M. Carroll Ed., MIT Press Publ., pp. 112-158.

[Bass 91] L. Bass, J. Coutaz: Developing Software for the User Interface; Addison Wesley, 1991.

[Bourguet 92a] M.L. Bourguet, J. Caelen: Interfaces Homme-Machine Multimodales : gestion des événements et représentation des informations; ERGO-IA’92 Proceedings, pp. 124-134, 1992.

[Bourguet 92b] M.L. Bourguet: Conception et réalisation d’une interface de dialogue personne-machine multimodales; Thèse de docteur de l’INPG, mars 1992.

[Coutaz 91] J. Coutaz, S. Balbo: Applications: A Dimension Space for User Interface Management Systems. In Proc. CHI'91, ACM Publ., May, 1991, pp. 27-32.

[Coutaz 92] Multimedia and Multimodal User Interfaces: A Taxonomy for Software Engineering Research Issues, East-West HCI’92, St Petersburg, August, 1992., August, 1992.

[Duke 92] D. Duke, M. Harrison: Abstract Models for Interaction Objects; ESPRIT BR 7040 Amodeus Project document, System Modelling/WP1, Nov. 1992.

[Nigay 92] L. Nigay: the MATIS system (Multimodal Air traffic Information System). Under press, 1992.

[Nigay 93] L. Nigay, J. Coutaz: A design space for multimodal interfaces: concurrent processing and data fusion, Interchi’93, Amsterdam, May, 1993.