Chapter 3

TOWARDS A SYSTEM OF PATTERNS FOR THE DESIGN OF MULTIMODAL INTERFACES

Guillaume Godet-Bar, Sophie Dupuy-Chessa, and Laurence Nigay
CLIPS-IMAG-IIHM, BP 53, F-38041 Grenoble Cedex 9 (France)
E-Mail: {guillaume.godet, sophie.dupuy, laurence.nigay}@imag.fr –
Web: http://iihm.imag.fr/site/
Tel.: + 33 4 76 63 59 70 – Fax: +33 4 76 44 66 75

Abstract Since R. Bolt's seminal "Put that there" demonstrator, more and more robust and innovative modalities can be used and empirical work on the usage of multiple modalities is now available for guiding the design of efficient and usable multimodal interfaces. This paper presents a system of patterns for capitalizing and formalizing this design knowledge about multimodal interfaces as patterns. Patterns are used for illustrating our system of patterns.

Keywords: Patterns, Multimodal interaction.

1. INTRODUCTION

The first multimodal interface was designed more than twenty years ago, with the seminal demonstrator of Bolt [2] combining speech and gesture: the "Put that there" interaction paradigm. Since then modalities as well as the knowledge on how to design and develop multimodal interfaces have evolved. Indeed, in addition to more and more robust and innovative modalities such as the Phicons [10], conceptual and empirical work on the usage of multiple modalities [17] is now available for guiding the design of efficient and usable multimodal interfaces. However, most of the results on the design of multimodal interfaces have not yet been formalized or included in a tool for helping a non-specialist to design a multimodal interface.

Our work addresses the problem of capitalizing and formalizing good design practices so as to help designers to design efficient and usable multimodal interfaces. Towards this goal we have chosen a pattern-based approach because patterns provide an efficient way for addressing the problem of capturing experiences related to recurring design problems. Their goal is to provide solutions general enough to be adapted to different contexts. Patterns have become a well-known way of organizing knowledge and experiences in various design domains.

In this paper, we focus on patterns for multimodality as a means of both facilitating multimodal interaction design and providing a basic tool support for multimodal interface design process. Following a review on interaction patterns in the Computer-Human Interaction domain in Section 2 we recall the criteria that a system of patterns should fulfill. We then present our system of patterns in Section 3 that we illustrate by considering two patterns.

2. INTERFACE DESIGN: THE PATTERN APPROACH

2.1 Existing Systems of Patterns for Interaction Design

Several pattern catalogs are dedicated to interaction design. Most of them focus either on traditional WIMP interface design or on web application design. These patterns include those described in [3,6,20,21]. They are generally classified according to usability criteria. None of them take into account multimodal interaction design.

Complementary to those patterns dedicated to interaction design, we also found patterns for the software design of the interfaces. For example, Bass *et al*. [1] identify patterns that relate a usability scenario with an architectural mechanism. As opposed to Gamma *et al*. [8], Bass *et al*. [1] propose architectural mechanisms that do not provide a software design solution but rather a set of requirements that must be addressed for the software design. Their goal is “to couple specific aspects of usability and architecture”. Again such software patterns do not cover the case of multimodal interaction.

Patterns described in [22] constitute the first approach to structured multimodal interface design and software design, defining patterns dedicated to output multimodality for presenting large information spaces. Several patterns for output multimodal interaction design are presented and linked with software patterns based on the PAC-Amodeus software architecture model [14]. Output multimodality has also been studied by Nesbitt [13]: a large set of multi-sensory interface guidelines (the MS-Guidelines) are defined as usability and design rules along with a process for multi-sensory output design (the MS-Process). Such results therefore do not follow the problem/solution structure usually found in patterns.

To sum up, in Human-Computer Interaction, several pattern catalogs have been proposed for interaction design or software design. The existing patterns for multimodality are dedicated to output multimodal interaction and none of the existing system of patterns addresses the design of multimodal interaction as a whole, i.e. input and output multimodal interaction.

For defining a system of patterns for multimodality, the first step consists of adopting a suitable pattern language. One approach could be to extend one of the existing systems of patterns to the case of multimodality. Nevertheless although the above-mentioned systems of patterns differ in terms of formalisms, none of them is adequate for our goal to encompass both multimodal design products and processes and to integrate the patterns in a tool:

- The patterns are presented in a narrative way [1,6], or are loosely structured [3,20,22]. It is therefore difficult to navigate in the set of patterns.
- The solutions provided by the patterns require a great deal of adaptation to be instantiated by the designer to her/his specific needs. Moreover the solutions provided by the patterns are not formal enough. Patterns are therefore difficult to formalize and consequently complex to integrate into tools.
- Patterns are often related by dependence links [3,6,20,21,22]. Those links participate in creating a hierarchal map of the patterns, and to a larger extent a pattern language, as explained by [18]. However there is no refinement to those dependencies, i.e. the links are not typed. This tends to create a loose pattern language structure, giving less opportunity for the designer to explore alternatives, refinements, etc.

As opposed to the above-mentioned studies, we aim at addressing the issue of interaction/software design for input and output multimodality, using a structured pattern-based approach.

2.2 System of Patterns for Multimodality: Criteria

Schmidt *et al*. [19] highlighted a set of criteria for useful and usable patterns. We have adapted them to our specific goals.

1. The patterns should attempt to federate good practices in multimodal interaction design. Such results in multimodality must be proven to increase the usability of the interactive system either by increasing the productivity, the comfort, the flexibility or the robustness of the interaction.
2. The patterns should be aimed at designers knowledgeable in traditional interface design, but lacking insight on multimodal interface design. The patterns should also provide sufficient knowledge concerning psycho-cognition, usability, and examples, so as to help make informed decisions about the design of a multimodal system.
3. The system of patterns should offer a set of process patterns that are in-

dependent of a specific development process. It should be possible to integrate the system of patterns into any kind of development process, using both the links between patterns and the contextual information provided by each pattern.

4. Solutions should be presented in a formal way when possible, using well-established formalisms, in order to provide the designer with semi-automatic evaluation and code generation tools.

We show in the following section how our system of patterns fulfills the above criteria by first presenting the adopted P-Sigma formalism and then the system of patterns illustrated by two patterns.

3. A SYSTEM OF PATTERNS FOR MULTIMODAL INTERACTION DESIGN

3.1 The P-Sigma Formalism

For describing our patterns, we apply the P-Sigma formalism [4]. P-Sigma is a structured approach for describing patterns that supports extensions in order to match specific domain-related needs. The P-Sigma formalism allows the description of three types of patterns: product, process and documentation. Process patterns address methodological problems and are usually used to break down development methods into smaller steps that in turn are also described using patterns. Product patterns answer design problems at any level of abstraction by proposing structuring models. For example, patterns by Gamma *et al.* [8] provide product patterns that describe solutions with the class diagrams. Additionally, the three types of P-Sigma patterns rely on the same problem/solution structure that is typically found in pattern description. Indeed, each pattern is divided into three major sections: Interface, Realization and Relations, which allow efficient internal navigation.

- The Interface section includes elements for selecting a pattern: a description of the designer's problem, of the context in which the pattern may be used (in terms of available products or patterns already applied), its strengths, how it classifies into the pattern language. Those aspects can be expressed either in textual form or more formally.
- The Realization section defines a solution as a product or a process, depending on the pattern type. This section can be described textually or formally by providing models, either by applying well-known formalisms or by using less defined descriptions. This section also describes application cases and consequences.
- The Relation section enables us to position a pattern with regard to other patterns within the system, thus allowing external navigation amongst the

patterns of the system. This section of the patterns also defines the overall organization of the patterns as a hierarchy. Internal and external navigation as well as the pattern hierarchy contribute to a definition of a system of patterns as defined in [18]. Additionally, P-Sigma supports relation typing. The relation types are: use, refinement, requirement and alternative. A "Use" relation acts as a pointer to another pattern referenced in the Realization section. A "Refinement" relation can be used to describe the current pattern as a specialization of another one. A "Requirement" relation shows which patterns need to be applied before using the current pattern. An "Alternative" relation points at patterns that respond to the same problem as the current one, but with different forces.

3.2 Our Patterns

In P-Sigma, patterns are classified explicitly in the "Classification" section of each pattern so as to facilitate the navigation amongst patterns: the classification (product/process/documentation) reflects the type of solution (methodology, models, etc.) that will be applied when using the pattern. We refine this classification to be specific to multimodal interfaces and we also add the position of the pattern according to the software design phases. Currently most of our patterns are dedicated to the specification phase. The elements of classification specific to our system of patterns are described below:

Product patterns are divided into:

- Interaction design patterns that describe the interaction between the user and the system, using specific models and constraints,
- Software design patterns akin to *Gamma et al*. [8] that describe the software design techniques supporting the interaction techniques.

Process patterns are separated into:

- Consistency patterns that aim at helping to check the overall consistency of the interactive system model,
- Methodological guidelines that fill in the procedural loopholes between the models either deduced from previous phases of development or extracted from product patterns.

Documentation patterns include:

- Patterns that provide sufficient theoretical basis for non-specialists to understand the taxonomies and models used within the product and process patterns. For example, one of the documentation patterns describes the ASUR formalism [7] that we apply to depict interaction design techniques in our product patterns.

- Patterns related to cognitive models of user's behavior, which are useful when designing multimodal interfaces. For example, one of the documentation patterns explains why the user tends to interact with a system in a multimodal way when manipulating a large number of concepts [17].

Our main contribution concerns the set of product and process patterns that we describe in the following sections.

3.2.1 Product Patterns

As explained above, our system of patterns includes two types of product patterns, which target different stages in the design (i.e., interaction design and software design). Interaction design patterns define interaction techniques according to different criteria that include:

- **User tasks**. For example, if the user has to perform critical tasks, information feedback and redundancy (see the CARE properties [15]: Complementarity, Assignment, Redundancy and Equivalence as relationships between modalities) are predominant design issues. For example we have defined a pattern that enforces the redundancy property by giving indications on how to add redundant modalities for the accomplishment of a task.
- **Concept domains**. For example, what would be an efficient interaction technique if the user has to manipulate objects on a 2D representation, e.g., a map. Fig. 1 provides an effective solution to such a problem.

The identification of the above criteria usually results from the application of process patterns, as explained in the following section (section 3.2.2 on process patterns). Additionally, the “Context” section in Fig. 1 describes which elements from the prior development phases will be necessary when applying the pattern.

Software patterns complement interaction patterns in that their solutions represent typical software design choices. However the developer is not limited to those and may provide her/his own software design. Indeed, the models and constraints detailed in the interaction design patterns offer sufficient information so as to allow the designer to build her/his own software architecture for supporting the designed interaction.

The distinction between interaction and software patterns is specified in the “Classification” section. For example, in Fig. 1, the “Classification” section defines that the pattern is a product interaction pattern, which can be used during the specification phase of a design process. Additionally, the section defines that the pattern addresses a task and/or a concept-specific problem.

Additionally, the formal solutions provided by both types of product patterns rely on a common formalism. Although still under study, its current form, which is based on the ASUR formalism [7], already constitutes a solid base for modeling user interaction and its software counterpart. The "Formal model" section of Figure 1 includes an example of solution described using ASUR:

- It describes interaction in terms of abstract modalities (i.e. abstract devices and abstract interaction languages as defined in [14], thus representing only the main characteristics that need to be integrated into the future system. As described in Figure 1, an efficient way to manipulate objects on a map would be speech inputs combined with pen inputs as well as graphical outputs.
- Physical proximity between devices is represented by a (=) relation. In Figure 1, such a relation is specified between the pen input device and the graphical output device.

The "Textual solution" section provides additional constraints that, among others, specify the nature of the coupling between the devices. The use of text is due to the limits of our actual description formalism. These constraints will also be integrated into the "Formal model" section. The "Application cases" section gives examples of systems which feature a spatial input interaction. We also provide references to the sources that contributed in identifying the strengths of the interaction pattern. The "Application consequences" section of Fig. 1 provides additional support for implementing the fusion of the speech and pen-input modalities, as well as details on alternatives that the developer may wish to explore.

Identifier	Spatial input interaction
Classification	{Product ˆ Specification ˆ Task or concept-specific ˆ Interaction}
Context	{Projected task tree ˆ Domain concepts ˆ Deployment environment ˆ Usability prescriptions}
Problem	One needs to design a multimodal input interaction adapted to tasks that involve the expression of spatial information. Such information includes: selecting objects in a 2D space (e.g. a map), modifying spatial attributes of an object (including position, size and orientation), designating points in a 2D space (e.g. a specific place on a map).
Strength(s)	• Increases the compactness of input expressions (efficiency) • Brings person-system interaction closer to person-person interaction (naturalness) • Allows the user to change modalities when realizing the tasks in an opportunistic way (flexibility)
Formal strength(s)	{Efficiency ˆ Naturalness Flexibility}
Solution	

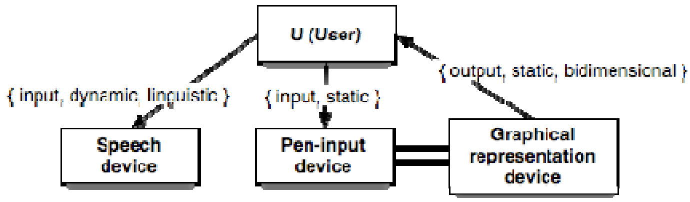
<table>
<tr><td>Formal model</td><td>
</td></tr>
<tr><td>Textual solution</td><td>The following constraints condition the application of this pattern.
• “Pen-input” and “graphical representation” modalities must be coupled.
• This occurs as a superposition of modalities, e.g. using the same device.
• The “Direct Manipulation” paradigm must be applied.
• The “Sketch& Speech” [12] paradigm must be applied.</td></tr>
<tr><td colspan="2" align="center">Application cases</td></tr>
<tr><td colspan="2">Geographic Information Systems (GIS) are good examples of interactive systems with spatial tasks. Users of such systems need to designate places, add objects on a map and move them around, give directions [16] or describe places and itineraries [12], using the “Sketch & Speech” paradigm.</td></tr>
<tr><td colspan="2" align="center">Application consequences</td></tr>
<tr><td colspan="2">• The designer may want to provide the user with a hand-free interaction model, using gesture recognition techniques that could allow her/him to directly interact on a large display with hands, as described in the pattern “Gesture-based input interaction”.
• The designer may want to use the pattern “Software aspects of spatial input interaction” for details on how to implement this interaction model.</td></tr>
</table>

Figure 1. An example of an interaction product pattern.

3.2.2 Process Patterns

Our system pattern includes two types of process patterns: the consistency and methodological patterns. The consistency patterns define methods to check the consistency of the interaction model in term of usability. For instance, one process pattern focuses on how to evaluate whether the selected input modalities may be conflicting, e.g. when two parallel tasks use the same input modality, thus creating ambiguities in input expressions to be interpreted by the system. The methodological process patterns provide the designer with indications on how and when to apply product and process patterns. The methodological patterns also allow the designer to adopt different types of development: either fast-paced design or more thorough design processes. The former implies only domain concept analysis (e.g., which concept attributes will be manipulated and which modalities would fit these attributes) while the latter includes a detailed analysis of the user’s tasks and therefore allows us to define more specific product patterns, such as the one described in the previous section.

Fig. 2 presents an extract of our root pattern, whose aim is to help the designer decide whether designing a multimodal input interface is appropriate. The elements of the “Classification” section help situate the pattern in the system and also characterize its generic aspect. In the “Context” section of Fig. 2, it is specified that the domain concepts, task tree and usability prescriptions have already been described in prior development phases. The “Formal steps” section’s activity diagram states that an analysis should be conducted on the artifacts specified in the “Context” section, while the “Textual steps” section gives indications on this analysis, more specifically on what criteria need to be fulfilled and on how to evaluate whether a given system would benefit from the application of multimodal input interaction.

Let us apply the activity diagram of Fig. 2 to a simple example. For instance, we consider the case where the user’s task analysis (as specified in the “Formal steps” section of Fig. 2) leads the designer to identify that the most often performed task is to view a small set of images (e.g., photographs) displayed on a grid. Based on the criteria described in the “Textual steps” section of Fig. 2, the designer may conclude that the user would probably not benefit from multimodal input interaction. We now consider the same system, but with a large set of images: The user may need to scale up or down, reorient and move images. By applying the criteria of the “Textual steps” section, the evaluation of the analysis clearly shows that, depending on the complexity of the tasks (e.g., the user might need to apply several transformations to a single image, scaling it down while rotating it), using a multimodal input interface might increase the efficiency and usability of the interactive system.

Identifier	Applicability of multimodal input interaction
Classification	{Process ˆ Specification ˆ Generic ˆ Consistency}
Context	{Task tree ˆ Domain concepts ˆ deployment environment ˆ usability prescriptions}
Problem	One needs to decide if, given a set of user’s tasks, a multimodal input interface needs to be designed.
Strength(s)	• Increases command expression density (efficiency) • Allows the user to change modalities depending on the context of use (flexibility) • Allows the integration of equivalent modalities, selected by the user according to her/his needs (flexibility) • Increases input and output data confidence, when they are expressed by different modalities (robustness) • Allows the user to interact with the system in a more intuitive and/or natural way (naturalness)

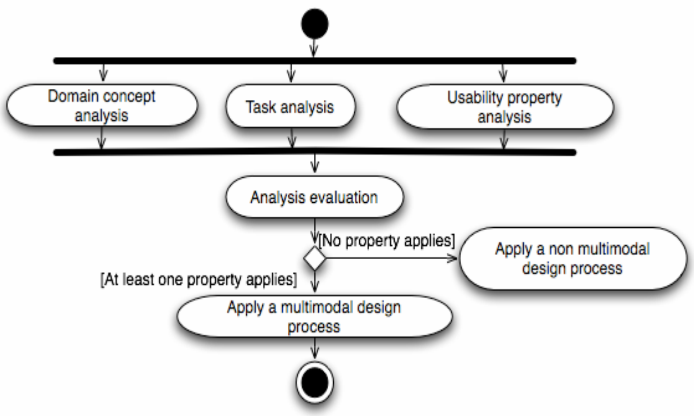
Solution	
Formal steps	
Textual steps	**Apply multimodality if at least one of the following assertions is verified:** **Tasks** • The user has to select objects from a large set of objects. • Some tasks imply the manipulation of many concepts. • Some tasks are critical (user's error rate has to be minimal). **Domain concepts** • Some concepts can be described using spatial data, e.g. its size, location, quantity, orientation or shape. • Some concepts will have to be selected amongst a large graphically represented set. **Usability properties** • Extend system's accessibility to diverse or non-specialist users. • Extend system's accessibility to users varying in age, skill level, sensory or motor skills. • The environment in which some tasks will be performed compromises the use of some modalities (noisy or dark environments, mobile situation, etc.). The system should therefore remain usable despite changing and constraining environments. • Some tasks need to be performed without compromising the user's privacy or security. For example the system should make it possible for the user to avoid using vocal commands when s/he does not want to be heard and should provide alternate input modalities.

Figure 2. An example of a methodological process pattern

3.2.3 Pattern relations

As stated in Section 3.1, the P-Sigma formalism allows us to link patterns in different ways (i.e. the four relation types: use, refinement, requirement and alternative). Fig. 3 shows examples of relations amongst our patterns. We can observe within this small sample of pattern relations that different levels within the hierarchy of patterns emerge: high-level process patterns, interaction design patterns and then software design patterns. Each level represents different phases within the design process of a system. Likewise, those relations clearly illustrate the external navigability between the patterns of our system.

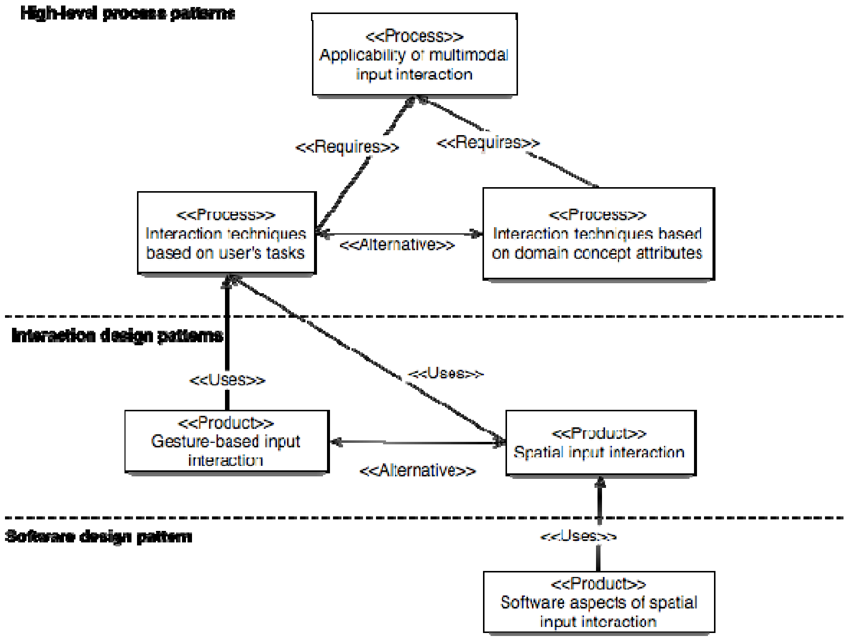


Figure 3. Sample of the relations between patterns

4. CONCLUSION AND FURTHER WORK

We have introduced our system of patterns for the design of multimodal interfaces. As a starting point for our system of patterns, we have described 15 patterns: 5 product patterns, 5 process patterns and 5 documentation patterns using the P-Sigma formalism. These patterns are structured as follows:

- **5 product patterns:** Spatial input interaction, Software aspects of spatial input interaction, Gesture-based input interaction, Input modalities for critical tasks, Modality availability.
- **5 process patterns:** Applicability of multimodal interaction, Interaction

techniques based on user’s tasks, Interaction techniques based on domain concept attributes, Input disambiguation, Consistency of input modalities for parallel user tasks.
- **5 documentation patterns**: Modality description, ASUR formalism, ASUR formalism for interaction techniques, ASUR formalism for software aspects of multimodal interaction, User behaviour and multimodal input.

Our current process and product patterns constitute the basis of a generic method for designing multimodal input interaction. Also, we have formalized some more specific interaction techniques. Our system of patterns fulfills the criteria described in Section 2.2:
- Our patterns implement good practices in multimodal interaction design since they are based on the domain literature (including the proceedings of the International Conference on Multimodal Interaction) and especially empirical results on usability such as the “ten myths of multimodal interaction” [16]. Based on the literature, we identified recurring practices that we formalized using the P-Sigma formalism. Novel multimodal interaction techniques may also be included into our system of patterns if they provide more efficient and usable techniques.
- Our system of patterns includes documentation on psycho-cognition, usability and insightful examples, similar to what has been done in [13], so as to help make informed decisions about the design of a multimodal system.
- Our patterns are independent from a particular development process. Nevertheless, they can be included in one. An ideal situation would be the development process described using P-Sigma, such as in [9].
- Solutions are presented in a formal way. Depending on the pattern type, different formalisms are used. Processes are represented using UML activity diagrams and products are described using a specific formalism.

While the format of the patterns is finalized and the way to formalize the patterns with the P-Sigma language is established, further work needs to be done on the models that are provided to the designer including their semantics and their description completeness as well as on the various classification schemes used for interaction design such as the user's task classification, and the modality classification.

Adopting the P-Sigma pattern language formalism enables us to employ our system of patterns using the AGAP tool [5]. AGAP supports the description of any pattern language as long as its grammar can be formally described. In addition, to support the specification of patterns, AGAP provides extraction of patterns to define browsable methodological guides. Once our patterns will be specified in AGAP, we will use its generation capability to obtain a website. This website can be used by designers on its own or can also be linked to other guides via HTML hyperlinks. For instance, we plan to integrate it within an extension to the development process described in [9], for post-WIMP interactive systems [11]. That will be a first attempt to validate our system of patterns.

However validating a system of patterns is a difficult task: since the very first system of patterns (e.g., [8]), validation has consisted of recognizing the patterns as a successful federation of accepted practices. Therefore our aim is to evaluate whether our system of patterns does indeed provide "logical" solutions for designing multimodal interfaces, and whether its use does help in creating usable and efficient multimodal systems. Towards this goal we plan to test our system of patterns on different case studies. Additionally, we envision conducting an experiment with masters students: for the design of an interactive system, we plan to provide one group of students with the entire system of patterns, instrumented with the AGAP tool, while another group will only rely on design methods as taught during the multimodal HCI course.

ACKNOWLEDGMENTS

This work is partly funded by the multidisciplinary French project VERBATIM and by the SIMILAR network of excellence (http://www.similar.cc), the European research task force creating human-machine interfaces similar to human-human communication of the European Sixth Framework Programme (FP6-2002-IST1-507609).

REFERENCES

[1] Bass, L., John, B. E., Juristo, N., and Sanchez-Segura, M-I., *Usability-supporting Architectural Patterns*, in Proc. of 26th Int. Conf. on Software Engineering ICSE'2004 (Edinburgh, 23-28 May 2004), IEEE Computer Society, Los Alamitos, 2004, pp. 716-717.

[2] Bolt, R.A., *"Put-that-there": Voice and Gesture at the Graphics Interface*, in Proc. of the 7th Annual ACM Conf. on Computer Graphics and Interactive Techniques SIGGRAPH'80 (Seattle, 14-18 July 1980), Computer Graphics, Vol. 14, No. 3, 1980, pp. 262-270.

[3] Borchers, J.O., *A Pattern Approach to Interaction Design*, in Proc. of ACM Conf. on Designing Interactive Systems: Processes, practices, methods, and techniques DIS'2000 (New York, 17-19 August 2000), ACM Press, New York, 2000, pp. 369-378.

[4] Conte, A., Fredj, M., Giraudin, J-P., and Rieu, D., *P-Sigma: un formalisme pour une représentation unifiée de patrons*, in Proc. of XIXth Congrès Informatique des Organisations et Systèmes d'Information et de Décision INFORSID'2001 (Martigny, 24-27 May 2001), 2001, pp. 67-86.

[5] Conte, A., Fredj, M., Hassine, I., Giraudin, J-P., and Rieu, D., *AGAP: an Environment to*

Design and Apply Patterns, in Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics SMC'2002 (Hammamet, 6-9 October 2002), Vol. 7, IEEE Computer Society Press, Los Alamitos, 2002, pp. 135-146.

[6] Coram, T. and Lee, J., *Experiences–A Pattern Language for User Interface Design*, Proc. of Joint Pattern Languages of Programs Conferences PLOP'96 Workshop on Patterns (Allerton Park, 4-6 September 1996), 1996, accessible at http://www.maplefish.com/todd/papers/Experiences.html

[7] Dubois, E., Nigay, L., and Troccaz, J., *Assessing Continuity and Compatibility in Augmented Reality Systems*, International Journal on Universal Access in the Information Society, Vol. 1, No. 4, 2002, pp. 263-273.

[8] Gamma, E., Helm, R., Johnson, R.E., and Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, Reading, 1995.

[9] Hassine, I., Rieu, D., Bounaas, F., and Seghrnouchni, *Symphony: a Conceptual Model Based on Business Component*, in Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics SMC'2002 (Hammamet, 6-9 October 2002), Vol. 3, IEEE Computer Society Press, Los Alamitos, 2002.

[10] Ishii, H. and Ullmer, B., *Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms*, in Proc. of ACM Conf. on Human Factors in Computing Systems CHI'97 (Atlanta, 22-27 March 1997), ACM Press, New York, 1997, pp. 234-241.

[11] Juras, D., Rieu, D., Dupuy-Chessa, S., *Vers une méthode de développement pour les systèmes mixtes*, in Proc. of Journée NEPTUNE'2006 (Paris, 16 May 2006), to appear.

[12] Lee, B-W. and Yeo, A.W., *Integrating Sketch and Speech Inputs Using Spatial Information*, in Proc. of 7th Int. Conf. on Multimodal Interfaces ICMI'2005 (Trento, 4-6 October 2005), ACM Press, New York, 2005, pp. 2-9.

[13] Nesbitt, K.V., *Multi-sensory Display of Abstract Data*, Ph.D. thesis, School of Information Technology, University of Sidney, Sydney, 2003.

[14] Nigay, L. and Coutaz, J., *A Generic Platform for Addressing the Multimodal Challenge*, in Proc. of the ACM Conf. on Human factors in computing systems CHI'95 (Denver, 7-11 May 2005), ACM Press, New York, 1995, pp. 98-105.

[15] Nigay, L. and Coutaz, J., *Multifeature Systems: The CARE Properties and their Impact on Software Design*, in J. Lee (ed.), Proc. of First International Workshop on Intelligence and Multimodality in Multimedia Interfaces: Research and Applications IMMI'1 (Edinborough, 1995), AAAI Press, 1995.

[16] Oviatt, S., *Ten Myths of Multimodal Interaction*, Communications of the ACM, Vol. 42 No. 11, 1999, pp. 74-81.

[17] Oviatt, S., Coulston, R., and Lunsford, R., *When do we Interact Multimodally? Cognitive Load and Multimodal Communication Patterns*, in Proc. of 6th ACM Int. Conf. on Multimodal Interfaces ICMI'2004 (State College, 13-15 October 2005), ACM Press, New York, 2004, pp. 129-136.

[18] Salingaros, N., *The Structure of Pattern Languages*, Architectural Research Quarterly, Vol. 4, 2000, pp. 149-161.

[19] Schmidt, D. C., Fayad, M., and Johnson, R.E., *Software patterns*, Communications of the ACM, Vol. 39, No. 10, October 1996, pp. 37-39.

[20] Tidwell, J., *Designing Interfaces*, O'Reilly & Associates, 2005.

[21] van Welie, M. and Trætteberg, H., *Interaction patterns in user interfaces*, in Proc. of 7th Pattern Languages of Programs Conference PLoP'2000 (Monticello, 13-16 August 2000), Washington University Technical Report number wucs-00-29, accessible at http://jerry.cs.uiuc.edu/~plop/plop2k/proceedings/Welie/Welie.pdf.

[22] Vernier, F., *La multimodalité en sortie et son application à la visualisation de grandes quantités d'information*, Ph.D. thesis, University of Grenoble, Grenoble, 2001.