Models Transformations for Ubiquitous System Design

Emmanuel Dubois, Christophe Bortolaso, Guillaume Gauffre
University of Toulouse - IRIT
118, route de Narbonne
31 062 Toulouse Cedex 9, France
[firstname.lastname]@irit.fr

ABSTRACT
Many different models and tools exist for supporting the design of ubiquitous interactive systems. Each of them deals with a different point of view. As a result designing such systems has to involve a set of models rather than just one. In this paper we first provide an overview on existing models dedicated to Mixed Interactive Systems, one form of ubiquitous systems. Then, to facilitate the elicitation of the most appropriate model, we organize them along the steps of the development process. Finally, to smoothly guide the use of these different design resources along the development process, we provide an overview of different linking mechanisms between design models for ubiquitous systems and highlight their characteristics.

Keywords
Interaction model, software architecture model, model transformation, mixed interactive system

INTRODUCTION
Among the most recent forms of interactive techniques, one aims at taking advantage of physical objects to support the interaction with a computer system: physical artifacts surrounding users during their activity become part of the loop. Users' everyday objects thus constitute an extension of their body to communicate with the system. Such systems are either called tangible UI, mixed or augmented reality, etc.: we hereafter refer to them with the generic term of Mixed Interactive System (MIS). Such systems are emerging in many different domains, ranging from specific application such as surgery [19] to mass market [15]. It also comes together with the emergence of new usages and the combination of advanced and various technologies. Furthermore, new spaces are now opened to interaction since the interaction is simply requiring the presence of everyday physical objects. According to Weiser's definition, it is therefore a form of ubiquitous interactive system because the interaction mechanisms "*weaved themselves into the fabric of everyday life*" [25].

Nevertheless, the growing interest into the development of such interaction forms is undoubtedly linked to the constant exploration of new sensors, modalities and communication channels: as a result these forms of interaction are very different from traditional WIMP based situations. To better understand their differences and precisely highlight their specificities, efforts has been paid to develop descriptive models: such models express considerations related to the interaction [7], the physical properties of the required entities [16], the abilities of the modalities involved [6], etc. We observe from the diversity of approaches that complementary aspects, relevant for the design of MIS are addressed by different models. In the course of the design, the designer thus has to identify, for each step of the development, the most appropriate model, method or tool supporting the design. Developing a MIS thus appears to be a real challenge [21]. An optimistic solution could rely on the use of a unique and universal approach, aggregating all the required dimensions and enabling the design of all kinds of MIS. However, given the low maturity of the domain, the multiple attempts being developed, such an approach is not yet conceivable

Rather than contributing to the creation of such a unique reference model, we propose to compare existing models according to their role and place in the development process. Then, to facilitate their combined use, i.e. to smoothly guide the use of these design resources along the development of MIS, we explore possible linking mechanisms between models. In this paper, we first give a brief overview and characterization of modeling approaches existing in the field of MIS. We then introduce three fundamental design resources on which we have investigated the development of different model transformations and couplings.

EXISTING MODELS IN MIS
Designing Mixed Interactive Systems (MIS) requires considering many specifics facets: the nature of physical artifacts, the links between these physical objects and digital data and the variety of devices and technologies which can be involved. Consequently, adapted design resources have been developed. Hereafter, we review a set of design resources dedicated to MIS.

First, conceptual frameworks [9,12,16,22] provide a high level of abstraction on the MIS field. They raise questions about the generic role of the system and its place in the physical world. They provide a big picture of the MIS field and somehow help to lead the **analysis** of interactive situation.

Taxonomies and models [6,7,23] have also been defined to understand mixed interactive situations, the elements that characterize them and their advantages. This second set of approaches therefore contributes to a better understanding of the **interaction design** of MIS.

E. Dubois, C. Bortolaso, G. Gauffre: Models Transformations for Ubiquitous System Design. Proc. of 1st International Workshop on Model-based Interactive Ubiquitous System 2011, Pisa, Italy, June 13, 2011, http://ceur-ws.org/Vol-787

Toolkits and frameworks [13,17,21], rapid prototyping environments [5] or runtime platforms [1,18] have been proposed to facilitate the **implementation** of MIS.

Finally, many user experiments results have been published to compare different MIS among them or against WIMP solutions [4]. In addition, **evaluation** methods dedicated to such tests are explored [24] to provide an appropriate form of evaluation.

All these modeling approaches cover different but complementary design considerations. Although their levels of abstraction vary, they offer a clear definition of the development space and constitute a common terminology supporting interdisciplinary communication. As depicted in Figure 1, we also highlight that the different design models and resources, used to develop a MIS, can be organized along the traditional phases of an interactive software design cycle. Existing models dedicated to WIMP can therefore easily be put in parallel with models dedicated to MIS, and either be compared to or used in addition to these dedicated models.

Phase	Ref.	Name	Type
Analysis	9	Fishkin's taxonomy	M
	12	Expected, Sensed, Desired	M
	16	RBI	M
	22	MCPrd	M
A to D	2	MACS	C
	3	KMAD to ASUR	C
Design	6	MIM	M
	7	ASUR	M
	10	ASUR-IL	M
	23	TAC	M
D to P	10	ASUR to ASUR-IL	T
Prototyping	1	DWARF	M
	5	Wcomp	M
	13	Phidgets	M
	17	ARToolkit	M
	18	Papier Mache	M
	21	TUIMS	M
Evaluation	4	RESIM	M, C
	24	design review	M

Figure 1: Existing design Models (M), Couplings (C) and Transformations (T) for MIS development.

The main limits are however that these models and approaches are almost exclusively usable by MIS experts and remain highly compartmentalized. Indeed, even if high-level resources used during the design should guide the implementation, concrete and systematic links have not been clearly expressed yet. To support the design through the four development phases of a MIS and to highlight a chain of design models and tools from the earliest design considerations to the latest in the development process, connections among models are required. We propose in the following section, an overview of different linking mechanisms we have been exploring over the past years. We depict the goal, source and target of the links between two models and describe the resulting overall mechanism (see Figure 7). The presented linking mechanisms are also positioned in Figure 1.

BASIS OF OUR INSTRUMENTED DESIGN PROCESS

The three pillars of our articulating efforts respectively support the abstract description of the user's interaction with a MIS, the software level decomposition required to implement the designed MIS and the concrete component based implementation of the final MIS. We first summarize these models and illustrate them on a case study: the notepad assisted slideshow.

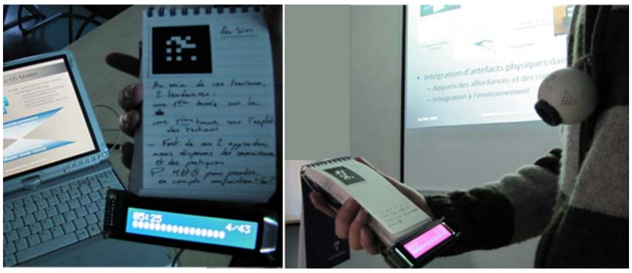

Figure 2: Use of the notepad assisted slideshow

For oral presentations, sequential slideshow systems like PowerPoint are largely used. The prototype we propose is a physical enhancement of a slideshow system: it involves the use of a notepad as "remote control" and feedback source, and associates each page of the notepad to one digital slide (see Figure 2). The speaker can thus write his own comments on the notepad and easily access to the corresponding slide. Potential animation steps of each slide are controlled through user's tap on the notepad. In the next sections, this prototype is used to illustrate the different models used to develop this system and how they have been linked.

Interaction model

Overview

ASUR [7] is a model which provides an abstract view on the user's interaction with a MIS. It describes physical and digital entities, adapters between the two worlds and information channels among them. It is a static representation: it describes a snapshot of the interaction required at a given time to perform a given task. It is also totally independent of the technology since it relies exclusively on an abstract description.

Goal

The goal of this model is to describe the different types of elements and data exchange required to support the interaction with a MIS. Both entities and channels are further characterized by attributes such as the type of entity (real object, adapter bridging the two worlds, etc.), the medium and language of the channels. Additional elements are expressible such as physical constraints among entities, links between a physical and digital entity, etc.

Example

The model of the notepad assisted slideshow is presented in Figure 3. The "user" interacts with the "notepad" for which each page is detected by an adapter ("PageDetector"). Another interaction with a second adapter detects user's tap ("StepDetection"). These two adapters deliver to the "slideshow" some digital data from which the current slide and step in the animation are identified. Finally, the state of the slideshow is rendered through three different adapters to the "user" and the "attendees".

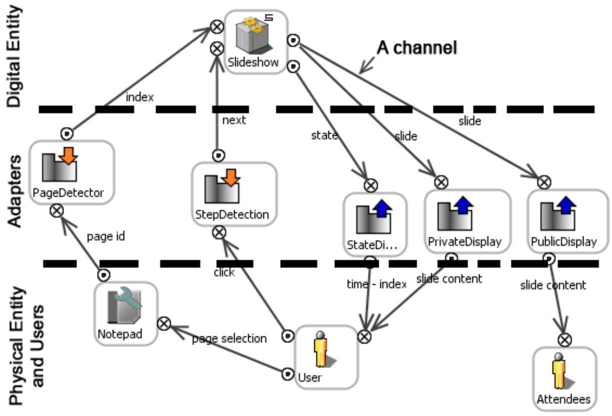

Figure 3: ASUR model for notepad assisted slideshow

Tools

A metamodel of ASUR has been defined [11]. Based on this metamodel a graphical editor has been developed as an Eclipse plug-in within the Eclipse Modelling Project [8].

Software architecture model

Overview

The ASUR-IL model is used to describe the MIS software architecture: it defines the software's skeleton through a components based architecture. As for ASUR it remains sufficiently abstract to be independent of the implementation platform since it relies exclusively on generic descriptions of the components.

Goal

The goal of this second model is to promote the integration of design considerations related to component-based specificities (port, data flow, component) and to software architecture of interactive system (functional core, views and controller). ASUR-IL is thus composed of two types of sub-assemblies:

- Adapters describing the required devices and API to implement the link between physical and digital world
- Entities describing system-dependent components; entities are decomposed according to the MVC pattern. It thus contains a Model, View(s) and Controller(s)

In comparison to ASUR, ASUR-IL adopts a software point of view on the interaction with the MIS. It therefore provides a list and description of the software bricks required, interfaces, ports, data types, etc.

Example

Left side of Figure 4 illustrates the adapter sub-assembly required to implement the "page detector" expressed with ASUR (Figure 3). It involves a digital camera component and a marker detection component. The right side of Figure 4 describes the entity sub-assembly required to implement the "slideshow" expressed in ASUR. This sub-assembly involves four components: two controllers, one model and one view.

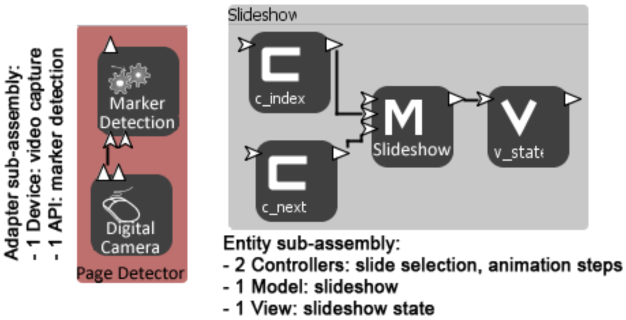

Figure 4: Detailed description of two of the ASUR-IL assemblies involved in the notepad assisted slideshow

The ASUR-IL model which entirely covers the case study is represented in Figure 5.

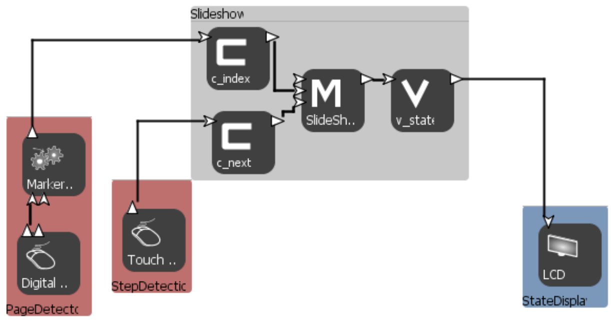

Figure 5: Two of the ASUR-IL assemblies involved in the notepad assisted slideshow

Tools

As for ASUR, a metamodel has been defined, and a graphical editor has been defined as an Eclipse plug-in.

Software component model

Overview

To implement the prototypes that we design with ASUR and ASUR-IL we rely on two existing prototyping platforms: Open Interface (OI) [20] and WComp [5].

Goal

The goal of such platforms is to allow a rapid development of system through manipulation of component assemblies at run-time. In association with each platform, a repository of components is available: it consists in a set of reusable software components ready to be integrated into new assemblies.

Example

The Notepad Assisted Slideshow has been implemented as an assembly of WComp components. This is illustrated in Figure 6. Each element of the assembly corresponds to one of the ASUR-IL element defined in the ASUR-IL model of the system.

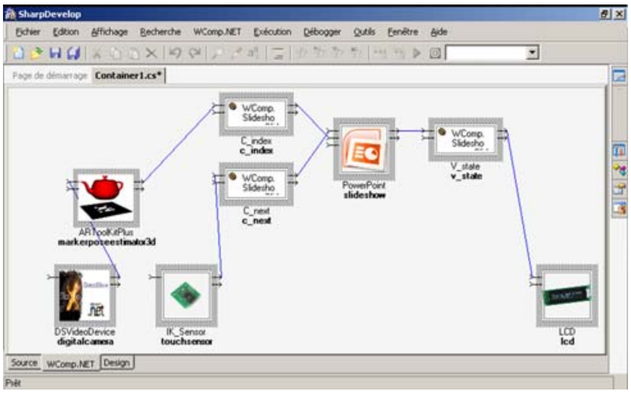

Figure 6: WComp assembly of the notepad assisted slideshow.

Tools

Metamodels of WComp and OI include the three concepts of component based architecture: components, ports and data flows. Each platform also offers a graphical editing environment for creating the appropriate assemblies.

Outcomes and limitations

There are obviously links between concepts expressed in each of these three pillars. But there is no clear constraint that drives their respective use and it is not ensured that the designer will conform to design recommendations made with the other models. For example, so far interaction design decisions expressed with the ASUR model are not constraining the development of the software architecture with ASUR-IL. And yet, these three pillars are required to drive a MIS from its early specification to its final implementation. We therefore complement them with linking mechanisms, such as model transformations or model coupling, in order to support the transitions between several phases of the development process. The goal is to better take advantage of the design choices expressed with the different models.

ARTICULATING DESIGN MODELS

MIS models have been developed to cover different phases of the development process: transformations and couplings are thus required at different places in the process.

From requirement to interaction design

Overview

We explored different linking mechanisms at this level:

- Linking models resulting of a KMAD task analysis to the ASUR interaction model [3]. It involves a unique expert, whose role is to translate the most of a task model into the definition of the contour of the interaction model: concepts expressed in the task tree are mapped to elements of an ASUR model describing an interaction technique supporting the realization of the task.
- Stimulating creativity session with the ASUR model [2]. A Model Assisted Creativity Session (MACS) is based on a scenario and a set of different constraints; it fosters the generation of mixed interaction techniques inside a group of multidisciplinary designers. MACS participants are invited to manipulate elements of an interaction model, in order to augment the potential of variations they might consider to generate ideas.

Goal

The goal of these two linking mechanisms is to ensure that the interaction techniques proposed are really in line with the specified task to achieve (KMAD-ASUR) and with the design problems to solve (MACS). In both cases the result of the linking mechanism is just a partial interaction model.

Tool

KMAD-ASUR and MACS are not so far supported by automatic tools. KMAD-ASUR is based on a set of rules and an algorithm describing the sequence of use: managing the alternatives generated by this transformation is left in charge of the designer. A MACS is composed of a set of steps, guidelines for the facilitator and manual post-treatments for the generated modelled ideas.

From interaction design to software architecture

Overview

This transformation, represented in the left side of Figure 7, converts an abstract specification of the interaction technique into a structured set of required software components: the generation of this software components structure is driven by the type of ASUR entity, attributes and channels involved in the model [10].

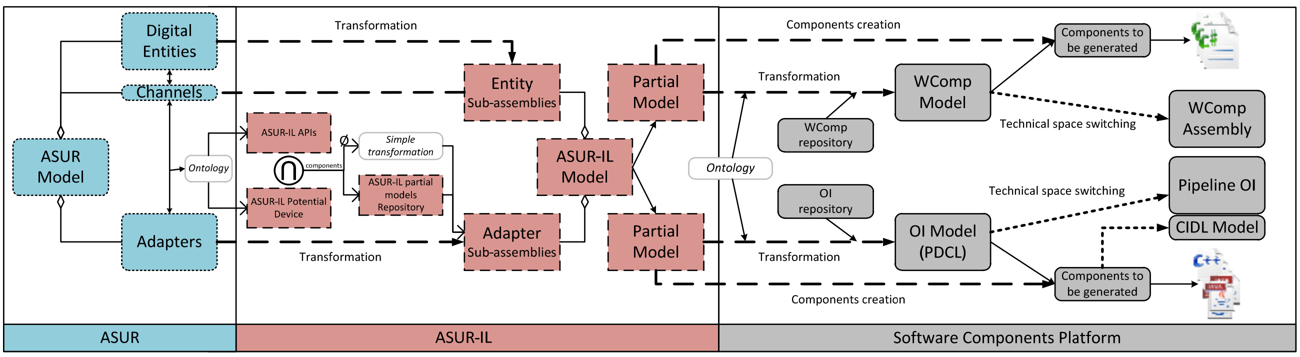

Figure 7: Synthetic summary of the model transformations linking ASUR to an implementation. (available as Eclipse Plugins http://www.irit.fr/recherches/ELIPSE/guideme/)

Goal
This transformation maintains coherence between the interaction specification and the proposed implementation. The software structure produced through it is only partial: indeed information related to the type of data for example, is not expressed in the ASUR model. This refinement of the software architecture design is thus left to the designer.

Tool
To support this transformation, ATL rules automate part of the transformation. It is assisted by the use of an ontology that establishes links between parts of the interaction design and parts of the software architecture definition: the ontology provides additional information to choose among existing components. A repository of already defined and used ASUR-IL components is available. Finally a wizard helps the designer to go through the different steps of the transformation and suggest design options. All these technologies have been packed into Eclipse plug-ins.

From software architecture to implementation

Overview
Translating ASUR-IL model to an implementation produces a running prototype [10]: this transformation is represented in the right side of Figure 7. The prototype is therefore made of an assembly of existing components (either from the OI or WComp platforms) and strictly conforms to the software architecture previously expressed.

Goal
The goal of this final transformation is to concretely instantiate the designed interactive technique. Until this final point, there is no need to pay attention to the soft- and hardware technologies to use. As a result, the running prototype can easily reuse existing bricks, even if they are not all available on the same platform: indeed communication mechanisms among the components have also to be specified.

Tool
To support this transformation, ATL rules, repositories of components, ontology and an interactive wizard have been developed. All these technologies have been packed into Eclipse plug-ins.

From development models to evaluation

Overview
We developed this linking mechanism in order to relate ergonomic recommendations to part of an interaction model describing the MIS [2]. This is based on a formal pattern describing usability recommendations: this pattern involves elements constituting the ASUR interaction model.

Goal
The main objective of this link between evaluation and design model is to facilitate the identification on the model, of part of the solution that is affecting (positively or negatively) the usability of the system. Such links thus potentially reduce the duration of one cycle of the four phases development process.

Tool
So far the navigation through the recommendations is only supported by a multiple criteria query on a web site. Refined tools would be useful so that usability recommendations pop out as soon as one relevant elements of the interaction model is added or selected.

CONCLUSION

In this position paper we highlighted the diversity of design-time and run-time models existing in the field of mixed interactive system, one form of ubiquitous interactive system. This diversity is partly explained by the amount of design considerations to handle when it comes to designing such systems: indeed most models covers only one specific aspect or at the best a limited subset of relevant considerations. However, following our analysis of existing works, we have been able to identify for each of these development resources one of the different phases of a development cycle for which the development resource is dedicated. This is thus classifying these design resources.

To go past the comparison of models through a classification, it is required to chain one model to another. Indeed one model provides one view on the system to design; chaining one to another provides a support for considering different complementary views without leaving one aspect aside. Furthermore, one unique and integrated design platform would be hard to propose because of the multiplicity of options, situations, technologies and usages potentially involved in a MIS. Chaining models to each other allows the definition of different ways in the design process: for example, going from A to B through a transformation in model C (A → C → B), may very flexibly be replaced by a longer transformation chain involving two other models instead of model C (A → D →E →– B). The result is the same, but the specialists of models D and E are no longer enforced to use model C.

Based on the different linking mechanisms between models of different phases of the development process that we have investigated, this paper showed that different forms of transformation exist: they use repositories of partial solutions, graphical representations, manual application of rules, methodological principles or transformation language. Among them, those exploiting Model Driven Engineering (MDE) approach and tools (ASUR to ASUR-IL to WComp/OI) appear to be the most promising: they use a standard language; they are easily supported by tools; they contribute to the definition and diffusion of the metamodels; they support the generation of multiple representations of the same model; they define transformation mechanisms, constitute guides through the design process; finally MDE has already proven its efficiency in classical software engineering. However, using MDE raised new challenges to investigate.

First in terms of properties, what happens to system or interaction properties settled in one model when a transformation is applied to the model? And more generally, are there properties of a transformation that are particularly important for "modiquitous" activities?

Managing retroactive loops in the design process also raises questions: if a model B is generated from a model A, how to ensure that modifications on B are still in line with A? How to send back to A modifications performed on B?

Given that ubiquitous systems are still quickly evolving and adding considerations to new dimensions, technologies, artifacts, etc., how MDE might help integrates these emerging new considerations? What would be the relevant characteristic of an ubiquitous interactive situation that could help identify the most relevant design path among the available models and transformations?

Finally, evaluating ubiquitous systems is a challenge in itself, but "modiquitous" activities are definitely contributing to this challenge through the elicitation of the most relevant design aspect of ubiquitous system, thus emphasizing the need to base the design of ubiquitous system on models.

REFERENCES

1. Bauer, M., Bruegge, B., Klinker, G., et al. Design of a Component-Based Augmented Reality Framework. ACM and IEEE *ISAR'01*, (2001), 45--54.
2. Bortolaso, C., Bach, C., Dubois, E., *A combination of a Formal Mixed Interaction Model with an Informal Creative Session. EICS'11*, 10 pages, 2011 **(in press)**.
3. Charfi, S., Dubois, E., Bastide, R.. *Articulating Interaction and Task Models for the Design of Advanced Interactive Systems. TAMODIA 2007*, Vol. 4849, Springer, LNCS, p. 70-83, 2007.
4. Charfi, S., Dubois, E., Scapin, D.L., *Usability Recommendations in the Design of Mixed Interactive Systems*, *EICS'09*, *USA*, ACM, p. 231-236, 2009.
5. Cheung, D., Tigli, J., Lavirotte, S., et Riveill, M. Wcomp: a Multi-Design Approach for Prototyping Applications using Heterogeneous Resources. *IEEE International Workshop on Rapid System Prototyping,* IEEE Computer Society (2006), 119-125
6. Coutrix, C. et Nigay, L. An Integrated Framework for Mixed Systems. in The Engineering of Mixed Reality Systems - Chap 2. Springer-Verlag London, 2010, 9-32.
7. Dubois, E., Gray, P., Nigay, L. ASUR++ : A Design Notation for Mobile Mixed Systems. Interacting with Computers 15, 4 (2003), 497-520.
8. Eclipse Foundation. Eclipse Modeling Project. 2006. http://www.eclipse.org/modeling/.
9. Fishkin, K.P. A taxonomy for and analysis of tangible interfaces. PUC'04, 8, 5 (2004), 347-358.
10. Gauffre, G., Dubois, E., Bastide, R.. *Domain-Specific Methods and Tools for the Design of Advanced Interactive Techniques.* in : *Models in Software Engineering*. Springer, Vol. 5002, LNCS, 2008, 65-76.
11. Gauffre, G., Dubois, E., Taking Advantage of Model-Driven Engineering Foundations for Mixed Interaction Design. Dans / In : Model Driven Development of Advanced User Interfaces, Springer-Verlag, Vol. 340, Studies in Computational Intelligence, 2011, p. 219-240
12. Gaver, B., Boucher, A., Walker, B., and al. Expected, sensed, and desired:A framework for designing sensing-based interaction. ACM TOCHI 12, 1 (2005), 3-30.
13. Greenberg, S. et Fitchett, C. Phidgets: easy development of physical interfaces through physical widgets. *UIST*, ACM (2001), 209-218.
14. GuideMe. Editor of MIS specific models. 2010. http://www.irit.fr/recherches/ELIPSE/guideme/.
15. Hornecker, E., Shaer, O., Tangible User Interfaces: Past, Present and Future Directions, in Foundations and Trends in HCI, Vol. 2 Nr. 1-2, 2010, pp. 1-138
16. Jacob, R.J., Girouard, A., Hirshfield, L.M., et al. Reality-Based Interaction: A Framework for Post-WIMP Interfaces. CHI'08, ACM (2008), 201-210.
17. Kato, H. et Billinghurst, M. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. IEEE IWAR'99, (1999), 85-95.
18. Klemmer, S.R., Li, J., Lin, J., et Landay, J.A. Papier-Mache: toolkit support for tangible input. CHI'04, ACM (2004), 399-406.
19. Lamata, P., et al., Augmented Reality for Minimally Invasive Surgery: Overview and Some Recent Advances, in Augmented Reality, ISBN 978-953-7619-69-5, (2010), p. 73 – 98.
20. Open Interface. STREP. http://www.oi-project.org/.
21. Shaer, O. and Jacob, R.J. A specification paradigm for the design and implementation of tangible user interfaces. ACM TOCHI, 16, 4 (2009), 1-39.
22. Ullmer, B. and Ishii, H. Emerging frameworks for tangible user interfaces. *IBM Syst. J. 39*, 3-4 (2000), 915-931.
23. Ullmer, B., Ishii, H., et Jacob, R.J.K. Token+constraint systems for tangible interaction with digital information. *ACM TOCHI 12*, 1 (2005), 81-118.
24. Wang, X. Dunston, P.S., Usability Evaluation of a Mixed Reality Collaborative Tool for Design Review, CGIV'06, (2006), p. 448 – 451.
25. Weiser, M., The computer for the 21st century. Scientific American, 3(265):94–104, 1991