

# New Human-Computer Interactions Using Tangible Objects: Application on a Digital Tabletop with RFID Technology

Sébastien Kubicki<sup>1</sup>, Sophie Lepreux<sup>1</sup>, Yoann Lebrun<sup>1</sup>, Philippe Dos Santos<sup>1</sup>,  
Christophe Kolski<sup>1</sup>, and Jean Caelen<sup>2</sup>

<sup>1</sup> LAMIH - UMR8530, University of Valenciennes and Hainaut-Cambrésis,  
Le Mont-Houy, F-59313 Valenciennes Cedex 9, France  
{firstname.name}@univ-valenciennes.fr

<sup>2</sup> Multicom, Laboratoire d'Informatique de Grenoble (LIG)  
BP 53, 38041 Grenoble cedex 9, France  
jean.caelen@imag.fr

**Abstract.** This paper presents a new kind of interaction between users and a tabletop. The table described is interactive and associated with tangible and traceable objects using RFID technology. As a consequence, some Human-Computer Interactions become possible implying these tangible objects. The multi-agent architecture of the table is also explained, as well as a case study based on a scenario.

**Keywords:** Human-Computer Interaction, RFID, tabletop, tangible objects, Multi-Agent System.

## 1 Introduction

Tabletops are a far cry from the personal computers currently used. Indeed, with the concept of the interactive table, we can imagine a collaborative and co-localized workspace allowing us to bring several users into work at the same time. Dietz and Leigh [6] propose an interactive table called “DiamondTouch”. They suggest an example of application concerning a plumber and an electrician working together on the same table. In this example, each participant can modify only the plans associated with his/her field.

Nowadays, applications and platforms which allow simultaneous collaboration between users, such as the multi-finger or sharing of documents in real time [11], are unusual. Therefore, current researches aim at exploring the possibilities of such new technologies [4]. Shen *et al.* [9] propose a software named DiamondSpin to simplify the development of interactive applications using the DiamondTouch tactile interactive table. DiamondSpin is a toolkit for the efficient prototyping and experimentation with multi-user, concurrent interfaces for interactive shared displays. It allows document positioning and orientation on a tabletop surface and also supports multiple work areas within the same digital tabletop. Besacier *et al.* [3] propose a whole of metaphors [2] in

direct relationship with the traditional use of a paper sheet on an interactive table. It would be possible to handle a document in a virtual way (for example to turn over it or group a whole of documents). These new interactions make it possible several co-localized users around an interactive table to work in a collaborative way.

However, even though several applications have been proposed for tabletops, very few of them use tangible objects to interact with users.

In this paper, we describe a new type of tabletop based on RFID (Radio Frequency Identification) technology which enables the users to manipulate tangible objects equipped with RFID tags (offering the possibility to store data of different types); so, several participants around the table can interact and work in a collaborative way on applications using physical objects (like in design or production tasks, games, etc.).

This paper details the TTT project, the multi-agent architecture of the tabletop as well as a case study.

## 2 TTT Project

The TTT (interactive Table with Tangible and Traceable objects) project proposes an alternative vision of the way of using tangible objects in conjunction with an interactive tabletop. For that, a new technology was implemented by using RFID tags stuck on objects by the RFIdées<sup>1</sup> Company. Four partners are involved in the TTT project: two laboratories (LAMIH<sup>2</sup>, LIG<sup>3</sup>) and two companies (CEA<sup>4</sup>, RFIdées). The magnetic table is connected to a computer, including an array of RFID antennas in the form of "tiles". The Figure 1(a) shows the prototype v1 of the digital table. It is possible to distinguish the different RFID antennas which compose the table delimited by the black line. The prototype is composed of so called tiles (Fig. 1(b)) each containing 64 antennas (8 x 8) per 2.5 cm<sup>2</sup>. Each tile contains a DSP processor which reads the RFID antennas, an antenna multiplexer and communication processor. The reading strategies are prioritized and the code is distributed between the processor reader antennas, the processor in charge of multiplexing and the host computer. The table measures about one meter square and contains 25 tiles (5 x 5) or 1600 antennas in total. The tiles are associated to each other via a control interface connected to the host computer by an Ethernet bus.

At this time, the prototype v3 can communicate with all the layers of the structure. We will explain this layered structure in the next part. The delay between two displacements is acceptable, we are able to play with some RFID tags like a marbles game but the delay could be again improved on the prototype v4.

### 2.1 Structure and Communication

An architecture including three layers has been adopted for the table (Fig. 2):

1. The *Capture and Interface* layer handles tangible objects provided with one or more tags per object and creates a java object by associating it to a form.

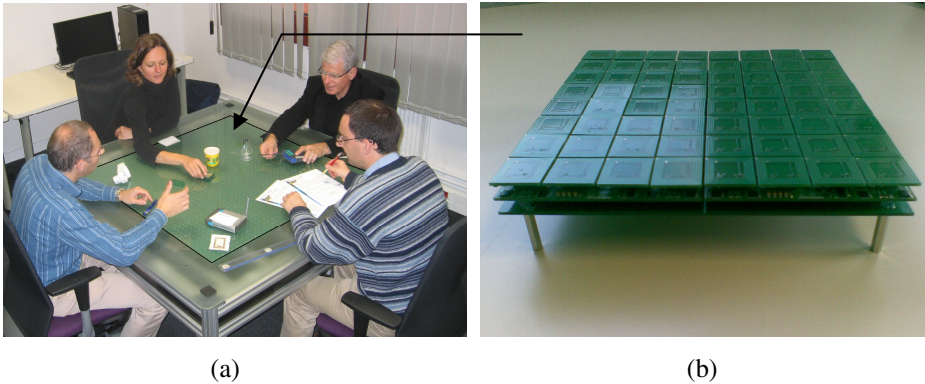
---

<sup>1</sup> [www.rfidees.fr](http://www.rfidees.fr)

<sup>2</sup> [www.univ-valenciennes.fr/LAMIH](http://www.univ-valenciennes.fr/LAMIH)

<sup>3</sup> [www.liglab.fr](http://www.liglab.fr)

<sup>4</sup> [www.cea.fr](http://www.cea.fr)



**Fig. 1.** (a) Prototype v1 of the table and (b) a tile containing 8x8 antennas (prototype v3)

2. The *Traceability* layer handles events associated to the objects and communicates the modifications of object positions to the applicative layer.
3. The *Application* layer manages the specificities of the application associated to the table.

This figure 2 also shows the data flows between each one of the layers. The data flows can only move from one layer into the adjacent one and must pass through an application interface. The applicative interface is used as connection between the layers and defines the exit and entrance points. It is via this interface that two layers are linked and will be able to communicate.

The applicative layer is broken up into two parts:

- The part integrating the *Multi-Agent System* (MAS) whose the supervisor agent called *Genius* has a total vision of the virtual and physical (tangible) objects, knowing all the characteristics of the application (role of each object, rules of the game, and so on).
- The *Human-Computer Interaction* (HCI) part which is given the responsibility of communicating with the users and which makes it possible to transmit virtual information (for example the displacement of a virtual object by the user).

## 2.2 The Multi-Agent System

The structure of a Multi-Agent System could be described by multiple ways according to the context of the application concerned. The main concepts used are the following:

- Tangible agents (represented by physical objects such as a book, a counter, a mobile phone),
- Virtual agents (displayed if they represent a digital object such as a colored zone for example),
- The users (people in interaction with the application).

So, it is possible to emphasize the various possible strong interactions (Fig. 4) (a strong interaction is an interaction which results from the dependence of an element

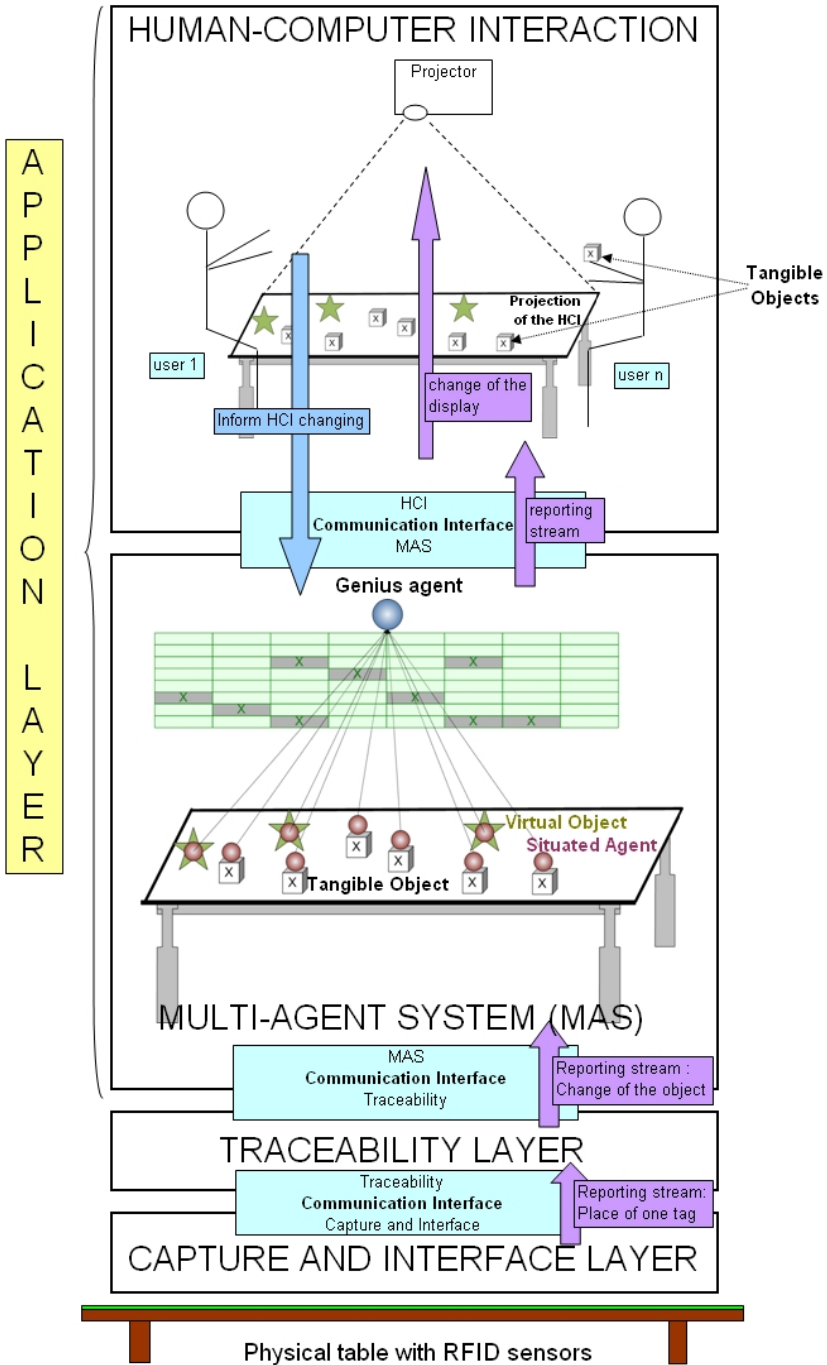


Fig. 2. The three layers composing the TTT Project



**Fig. 3.** An example of tactile display (a) and inclusion of RFID tags on a glove (b)

compared to another). We find in this case the different interactions between the tangible or virtual agents and between the users and the tangible or virtual agents. As a consequence, we can deduce for example that a tangible agent can act on the virtual agent location and not the opposite. A tangible object can modify the virtual object location (please note that the reverse goes against all physical laws and it is not, in our case, possible). It will be possible, with this interactive table, to interact with some tangible and virtual objects but for virtual objects, a tactile technology must be available. For that, two solutions are possible:

1. The next prototype can include a tactile display (Fig. 3) and permit to interact directly with fingers.
2. We can adopt the solution of an interactive glove with RFID tags to simulate a tactile technology.

That is why we distinguish the possibilities of interactions between user and agents (Fig. 4). In the case when the tactile is not implemented, the user can only interact with tangible objects; so, tangible agents associated to each object are involved. If the tactile is available, the user could interact with the tangible and virtual objects.

The *Genius* agent used is a software entity able to answer any question of the users about the objects location and their roles. This agent is the central point of the Multi-Agent System; the agents of the application interact with it to announce or modify their location.

The multi-agent organization proposed follows a hierarchical structure with flexible levels. At the top of this hierarchy, the *Genius* agent can be considered as an observant agent (or even coordinator depending on applications) which establishes the link between the users and the agents.

On the last level, the located agents (tangible agents dependant on a physical object), propose a reflection of the tangible objects present on the interactive table. Each object is associated with an agent which has the characteristics of this object such as: role, location, environment. These agents return the various internal modifications to the *Genius* which knows the general map of the table.

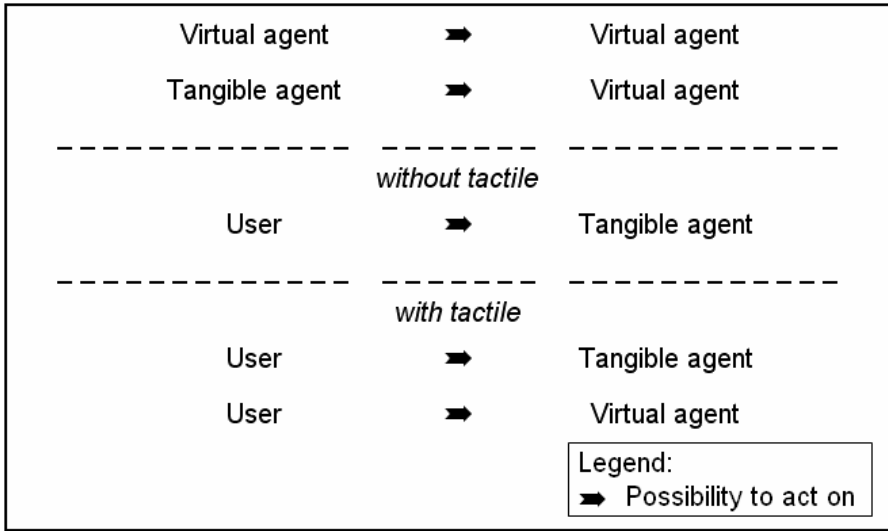


Fig. 4. The possible interactions between agents and users

### 2.3 Interactions Between the Table and Users

Initially, we have to agree that all the virtual objects have to be defined when designing the application. All these objects are initialized when the application is starting. After that, the user can interact with these virtual objects to move them, for example, with a finger or an object equipped with a RFID tag. The first stage to develop consists in initializing the virtual and tangible objects which will be used in the application; that is why we distinguish two types of users:

- The *end user* of the application does not need particular knowledge to use the application. He or she can be of any age, knowledgeable in data processing or not; he or she has just to know the rule(s) of the game or the principle of the application.
- The user known as *administrator* has a complete knowledge of the application; the administrator can intervene on the application (internal modification).

However, one can suppose that an *end user* could take the role of *administrator* for some operations. The Human-Computer Interface will have to be easy to use.

We have presented the layered structure defined during the TTT project. The *Captures and Interface* layer manages the interactions on the table; it transmits information to the *Traceability* layer which creates the historic of the different objects. Then the *Traceability* layer transmits information to the *Applicative* layer, more exactly to the *Multi-Agent System* layer which gives a role to each object and informs the *Human-Computer Interaction* layer to display the result to the users.

To illustrate an application using the table, we propose an example which uses the entire layered structure and shows the possible interactions with users.

### 3 Case Study and Modeling

We propose an example of application which points out the structure presented before. After a presentation of the application called *Luminous Zone*, we present a scenario using it.

The scenario emphasizes the utilization of *Luminous Zone* but we explain more especially the first stage which consists in initializing the different objects. In fact, before using the application with some objects, the user must initialize all the objects which have to be used. Two sequence diagrams (Fig. 5 and 6) modeled with UML2 will show the communication between each layer.

#### 3.1 Example

In order to illustrate the used architecture (Fig. 2), we propose an example in which the table has to illuminate a zone specified beforehand according to the location of a switch object (a tangible object initialized previously).

In this example, there are several objects, each with a role. First, the tangible objects could be everything (a pen, a counter, a book for example). If the user places a tangible object in one of the virtual colored zones (projected), the lighting zone (LED included in the table) lights up color in which the tangible object (the switch) is located. It is possible to use several switches. For example, the colored zone would be divided according to the number of users (let us suppose a side for a user, therefore, four users is the maximum).

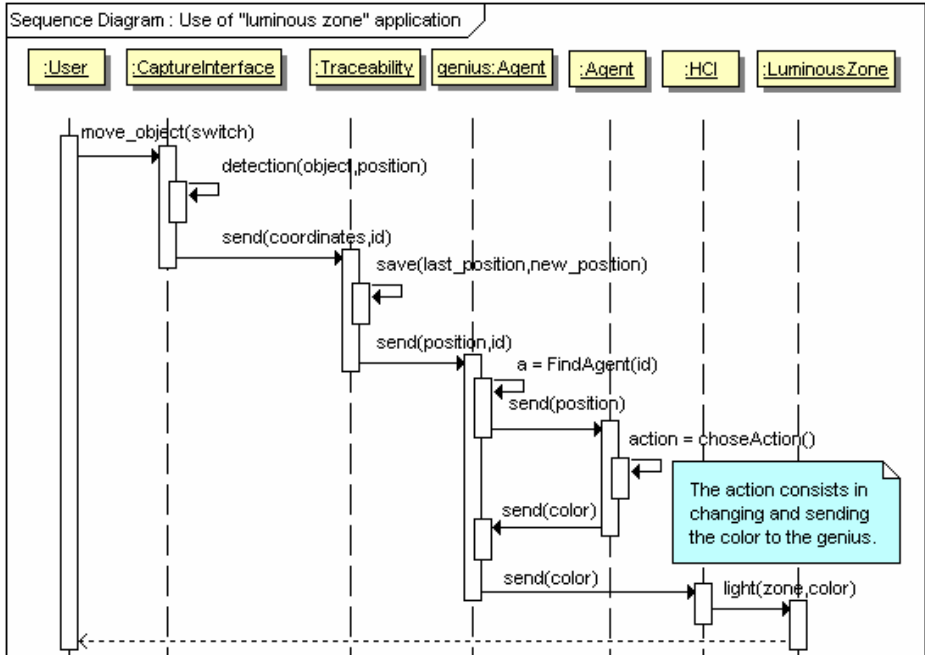


Fig. 5. A scenario using the table with *Luminous Zone* application

### 3.2 Scenario

We propose a sequence diagram which is modeled with UML2 (Fig. 5) to illustrate a scenario using the table with the *Luminous Zone* application.

The user moves an object having for role “switch”. Displacement is detected by the *Capture and Interface* layer which transmits information to the *Traceability* layer.

This one sends the new location of the object to the *Genius*. In *MAS* layer, each object (tangible or virtual) is represented by an agent. Each agent questions its local environment in order to know its location compared to the other objects. Here, the agent associated with the “switch” object checks if it is set in a colored zone; if necessary, this agent transmits to the *Genius* the need for lighting one of the luminous zones. After reception of the data by the *MAS* layer, the *HCI* layer assigns a color to the luminous zone and light it.

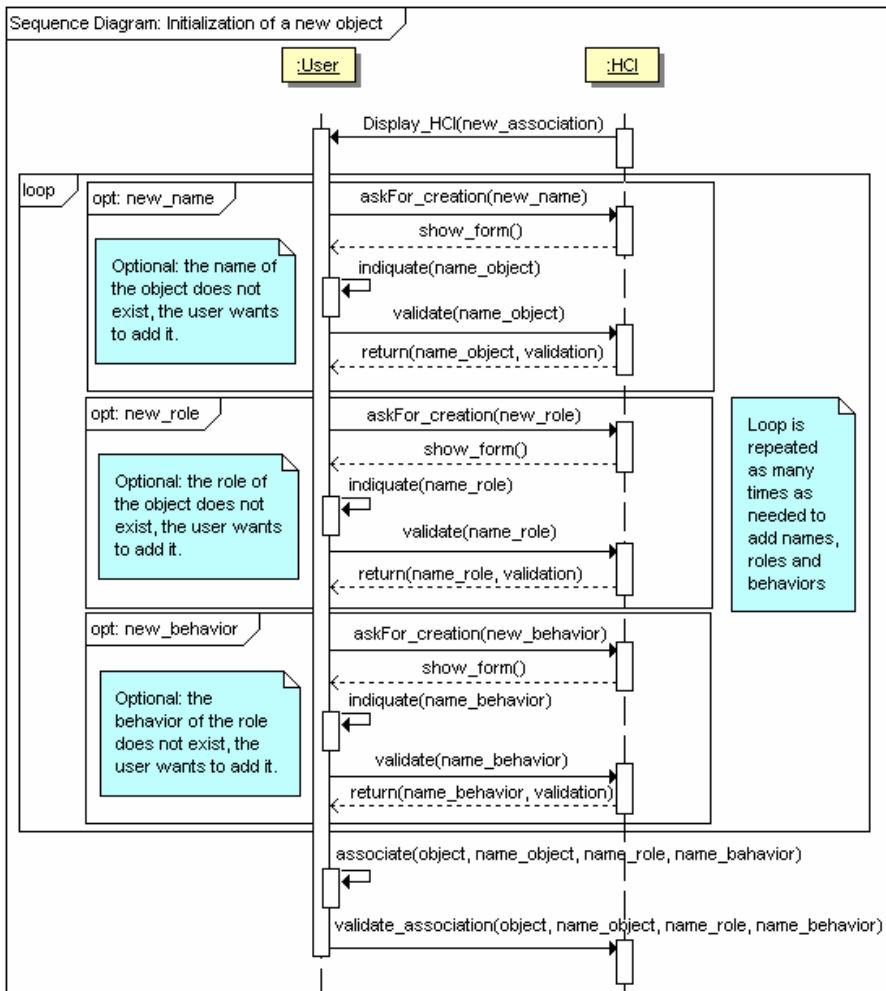


Fig. 6. Initialization of a new object and association of it with a role and behavior



To use the application, all the objects must be initialized. For that, a user interface is necessary. This one has to offer the choice to name an object, define these roles and behaviors. The sequence diagram (Fig. 6) shows such aspects.

## 4 Conclusion

The table has the original characteristic of interacting directly with users and tangible objects. This new working aspect is different compared to the interactive tables available currently. It permits to explore a new way of research in HCI as well as in MAS. The association between interactive table and Multi-Agent System is original and brings promising possibilities [1]. HCI will be used for direct interaction with the users allowing a simple and intuitive use of the applications of the table. It will propose innovations in terms of HCI in the use of an interactive table which manages tangible and traceable objects. Some examples of this are the modification of context [5] and [7] during the initialization of a tangible object (detection of the user, loading of personal parameters...) and so on, the adaptation to the context [10] and [8] (mono or multi-user, modification of environment, and so on).

Our objective is now to apply this research and use the different UML diagrams to develop a set of applications using the table and its specificities. At this time, the first application *Luminous Zone* is under test. A new version of the table is under development by the RFIdées Company.

## Acknowledgements

The present research work is supported by the "Agence Nationale de la Recherche" (ANR). We would also like to thank our two partner companies in the TTT project: *RFIdées* and the *CEA*.

## References

1. Adam, E., Mandiau, R.: Flexible roles in a holonic multi-agent system. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) *HoloMAS 2007*. LNCS, vol. 4659, pp. 59–70. Springer, Heidelberg (2007)
2. Agarawala, A., Balakrishnan, R.: Keepin' it real: pushing the desktop metaphor with physics, piles and the pen. In: *Proc. CHI*, pp. 1283–1292 (2006)
3. Besacier, G., Rey, G., Najm, M., Buisine, S., Vernier, F.: Paper metaphor for tabletop interaction design. In: Jacko, J.A. (ed.) *HCI 2007*. LNCS, vol. 4551, pp. 758–767. Springer, Heidelberg (2007)
4. Couture, N., Rivière, G., Reuter, P.: GeoTUI: A Tangible User Interface for Geoscience. In: *Proceedings of the second ACM International Conference on Tangible and Embedded Interaction*, Bonn, Germany, pp. 89–96 (2008)
5. Dey, A.K., Salber, D., Futakawa, M., Abowd, G.D.: An architecture to support context-aware applications, *GVU Technical Reports*
6. Dietz, P., Leigh, D.: Diamondtouch: A multiuser touch technology. In: *UIST 2001: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pp. 219–226. ACM Press, Orlando (2001)

7. Hariri, M.A., Tabary, D., Lepreux, S., Kolski, C.: Context aware business adaptation toward user interface adaptation. *Communications of SIWN* 3, 46–52 (2008)
8. Lepreux, S., Hariri, M.A., Rouillard, J., Tabary, D., Tarby, J.-C., Kolski, C., Jacko, J. (eds.): Towards Multimodal User Interfaces Composition based on UsiXML and MBD principles, *HCI International 2007, 12th International Conference 2007*, pp. 134–143 (2007)
9. Shen, C., Vernier, F., Forlines, C., Ringel, M.: DiamondSpin: An extensible toolkit for around-the-table interaction. In: *CHI 2004 International Conference on Human Factors in Computing Systems*, pp. 167–174. ACM Press, New York (2004)
10. Sottet, J.-S., Calvary, G., Coutaz, J., Favre, J.-M., Vanderdonckt, J., Stanculescu, A., Lepreux, S.: A Language Perspective on the Development of Plastic Multimodal User Interfaces. *Journal of Multimodal User Interfaces* 1, 1–12 (2007)
11. Wu, M., Balakrishnan, R.: Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: *UIST 2003: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pp. 193–202. ACM, Vancouver (2003)