Preliminary Design on Multi-touch surfaces Managed by Multi-agent Systems

Claude Moulin, Alistair Jones, Jean-Paul Barthès, Dominique Lenne

Université de Technologie de Compiègne, France
Heudiasyc, CNRS JRU 6599
{claude.moulin,alistair.jones,jean-paul.barthes,dominique.lenne}@utc.fr

Abstract

This paper presents a software architecture based on two multi-agent systems for supporting the work of experts during the collaborative preliminary design phases of a project. The working sessions involve a multi-touch interactive table and an interactive whiteboard. Moreover, the participants may be helped by personal assistant agents situated on small devices such as tablets. The paper describes the different phases of preliminary design and introduces the functionalities of the system allowed by the different devices and the agents. Then, the organization of the agents in the systems, their role and the objectives of their communication are presented.

Keywords: *Multi-agent system, Multi-touch surface, preliminary design, gesture input, collaboration.*

1. Introduction

Projects involving interactive tabletops and whiteboards have become increasingly popular. Recent research shows the advantages of direct manipulation of objects on multi-touch surfaces. Rekimoto's Augmented Surfaces project [1] is one of the first real works integrating multi-surface interaction techniques, and since, the possibilities for applications have been growing rapidly. The research community is just now developing examples of environments fully dedicated to supporting specific applications and proposing new scenarios involving such professions as physicists or medical doctors [2].

Our efforts are focusing on the field of preliminary design, which employs mostly highly collaborative methods, such as brainstorming or causal analysis. The project that we describe in this paper is based on a system that integrates two interactive surfaces; a tabletop prototype we have previously constructed named TATIN (TATIN stands for INteractive Tactile TAble in French) and a whiteboard. We observed that some phases of preliminary design are more efficient when performed on a horizontal surface like a table and others on a vertical one like a whiteboard.

We based our observations on a first summative evaluation of the TATIN tabletop, which reveals certain usability issues. Following this, we conducted a formative evaluation of a traditional pen and paper whiteboard and tabletop environment where design teams practice causal analysis to better understand the interactions which take place in this environment. From the results of these studies, we have built upon our interactive tabletop prototype and designed a multi-surface and multi-modal interactive environment. We also observed that the different steps involved in preliminary design are not completely separated nor sequentially ordered, and are frequently highly connected and dependent on one another. In this way, we built a sufficient model of preliminary design methodology, and then it was necessary to develop compatible multi-user software for supporting the different phases of preliminary design. We selected the brainstorming phase as our first candidate for implementation, and upon the completion of the software, we conducted usability experiments.

The results of our first experiments were very encouraging but it was clear we did not take all the advantages that the use of digital devices could bring. In particular, supporting facilities could help participants in their project. Capitalization of information was not maximal. First experiments also showed the difficulty to enter texts on objects displayed on interactive tables. Users often found it cumbersome to have to open and hide virtual keyboards, edit texts, and so on. Thus we decided to add to our system the voice input modality in order to be able to give some commands by interpreting the sentence generated by a speech-to-text engine.

The work presented in this paper is part of the new TATIN-PIC (PIC stands for Intelligent Platform for Design in French) project that benefits from previous analyses and experiments and adds several ambitious challenges. A larger description of the TATIN-PIC project in presented in [15]. Our goal is also to provide an assistance system for the participants of a preliminary design project [17]. It is based on the use of different information sources: other similar projects, documents bases, knowledge bases. To take all these features into account, we decided to give a multi-agent architecture to the system we developed. The objective of this paper is to describe this architecture. It is organize as follows. Section 2 presents the main elements necessary for a good understanding of the notion of preliminary design. Section 3 describes the different functionalities that support the work of participants during the identified phases of preliminary design. Section 4 presents the situation of the agents, their role and their communication.

2. Preliminary Design

In recent years, the field of product design has evolved and progressed with the introduction of notions like design thinking and user-centered design. For example, mechanical engineering projects have begun to take notice, and generally begin with a preliminary design project phase. Such a project requires only an initial problem statement to begin. Then the participants try to explore enough of the solution space for determining the feasibility of certain options. The result of a successful application of preliminary design ultimately leads to a working copy of a detailed project plan, completed with the justified approach to the solution, and a timeline describing the stages of project.

Important aspects of preliminary design include the design phases necessary to analyze different aspects of the future project and the roles played by the participants. The design phases frequently performed in preliminary design for engineering projects include brainstorming, functional analysis, risk analysis, causal analysis, and project planning. Participants are led to use a methodological circulation between these different phases allowing the groupwork to pass through phases to propose and explore ideas, resolve diverging points of view, focus on some elements and validate main tasks. The responsibility of the passage between very well identified design phases, falls on the moderator of the session, whose role is to coordinate discussion, help to reach consensus, and intervene before the team experiences diminishing returns on their discussions.

For example, in a simplified scenario, a preliminary design phase might open with a brainstorming session for identifying different causes of the problem. Several resulting ideas from this brainstorming might inspire the group to complete a causal analysis. These two phases permit the team to explore the solution space and entertain diverging ideas. Next, after the causal analysis, the team has a greater understanding of the causes of the problem, and the team decides to focus on one that they find particularly salient. The moderator listens to group and suggest performing a risk analysis to better understand the pitfalls of pursuing this as a solution. Here, the team enters a stage of convergence and focus. Participants are free to return to exploring more of the solution space with additional brainstorming sessions and causal analyses at any time. More details about preliminary design can be found in [14].

Users must be encouraged to pass with flexibility and speed from one phase to another, and to perform a methodological circulation through the different conceptual stages of the project. Thus, it is necessary to provide efficient and compatible tools able to present different views on same data and to allow adapted gesture to manipulate and interact with the objects that display the data. For brainstorming and causal analysis sessions we adopted the metaphor reproducing the traditional approach consisting in creating and organizing logically digital post-it notes. However, capitalizing on some results and adapting light transitions between two representations of data at different stages is more difficult. Digital tools must truly assist the participants in rendering components reusable, allowing pieces of previous work to be translated into different contexts and expanded upon in more detail with greater ease [3]. Having two possible devices, the table and the whiteboard, enables us to choose the more appropriate surface to display the object model of a phase, according to the roles of the participants or their preferences. Figure 1 shows the surfaces and the logics of communications between them.

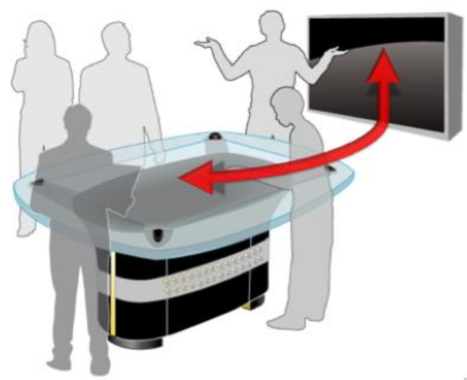

Figure 1: Multi-touch Surfaces

The following figure (see fig 2), taken during the first experiments of the early prototype, shows an actual situation where the group of participants has just finished a phase of task brainstorming. The moderator has decided to go on the preliminary design in front of the whiteboard and begins a planning phase. For this new phase, the content of post-it notes and their organization in group are transferred from the table to the whiteboard. The same information is displayed by different widgets. The direction of objects on the whiteboard is now logically vertical from top to bottom. The same phase should have been conducted on the table, but the moderator chose the first solution.

Figure 2: Simultaneous Use of Table and Whiteboard

3. System Functionalities

3.1 Devices

In our project, we consider two main devices, a multi-touch table and an interactive whiteboard. They are managed by two different computers. The whiteboard is not necessarily multi-touch because it may be used during a phase where only the moderator has the main role and is solely responsible for the modification of the data. Obviously both computers have access to the local area network and to the Internet. Each participant to a session is also connected to a voice interface allowing them to communicate with a small device (laptop, tablet, etc.). The system enables any communication between all the devices; this implies it must manage up to ten elements when eight people are working around the table.

3.1.2 Interactive Table

We have built an interactive multi-touch tabletop (TATIN for short) for supporting a collaborative preliminary design sessions. The TATIN platform uses two HD video projectors positioned side by side to render the final double full-HD 83-inch image (1920 pixels × 2160 pixels and 1.60 m × 1.40 m). The input device of the platform TATIN is based on LLP (Laser Light Plane) technology [4, 5]. Infrared lasers augmented by linear filters are used to create a laser plane flush with the top surface of the table. All objects or users' fingers in contact with the surface of the table disrupt the laser plane. Two infrared-sensitive cameras beneath the table are responsible for tracking the fingers illuminated by lasers. Image-processing treatment (extraction of background, high-pass filter, etc.) is applied to the camera images to determine the position of different contact points on the surface of the table and transform them into software events.

The dimensions of the table may accommodate up to eight participants for a work session. However, we can say that six people are more comfortable (see Figure 3). All the participants work on the same phase of the preliminary design of a project. The software developed for supporting the phases is based on a set of workbenches, one for each participant, giving access to the tools (virtual keyboard, menus, palette) allowing to perform actions that the phases require. Some actions may be directly managed by gestures on the table. For example, a longer tap on the table makes a circular menu appear. In other works we conducted experiments to determine natural gestures that people would use, rather than gestures they would be instructed to use. However, some gesture recognition is difficult to implement with current technologies and the gesture input remains quite unfamiliar to participants and requires more cognitive effort than intuitive touch input [18]. See also [13] for analysis of natural gestures.

During brainstorming sessions, it is possible to create, edit, delete or group several post-it notes. Internally, a post-it note is a structure containing a text and eventually some pictures. It also contains all the elements that are useful for its identification. A workbench is a conceptual notion and is not linked to a specific zone on the table. All objects created on the table are common to all participants even if they refer to the workbench they come from.

Figure 3: Work around the TATIN Prototype

Participants may also use the vocal interaction in order to perform some but not all actions. The system managing the voice interaction is linked to one workbench. Concretely, a message is sent by that system to the corresponding workbench. For example it is possible to group some post-it notes using their identifiers. A workbench is also a handler for tools that allow participants getting information from different sources like document or knowledge bases or even the Internet.

The workbench of the moderator also contains a menu that allows her to open a session of a project or to create a new one and to switch the current phase on the table. Each phase is in charge of creating a data model corresponding to the objects displayed on the table. When changing a working phase, the new one is responsible for the deployment of the objects of its model on the table. The moderator menu also allows the objects of a phase model to be sent to the whiteboard.

3.1.3 Interactive Whiteboard

During a session, whenever needed, it is possible to work on a phase around the table and to observe the transfer of the content of another phase of the same project on the whiteboard. The moderator is in charge of this transfer and can use either a menu on the table or a menu on the whiteboard according to her position in space. It is also possible to switch from one phase to another one on the whiteboard. When displaying elements on the whiteboard, each participant can see these elements with the same orientation whereas in the case of tabletop displays, users look at the screen from multiple viewing angles, requiring special considerations to the orientation of information [16]. That means that this vertical surface is more appropriate for phases where elements are highly structured as within a table or a chart, for example.

3.1.4 Small Devices

Each small device is connected to a workbench on the table. It contains the speech-to-text engine that transforms voice commands into messages sent to the corresponding workbench that analyzes them and produces the intended actions. It also contains a dialog system that may ask for missing information before sending a complete message. Small devices also contain modules that can be personalized and adapted to the user profile. They can assist a participant during a phase of the project. They are able to analyze the current content of the project, compare it to other ones by determining a similarity measure between projects. A very simple way is to consider a project as a document (composed of the pieces of texts contained in project elements) and use natural language processing tools to compare two projects. Semantics techniques can also be used. There is no limitation on tools that can be installed on small devices in order to provide assistance to participants.

3.2 Synthesis of the functionalities

We consider three kinds of facilities that the system will provide for.

- Facilities linked directly to the work of the participants around the table and in front of the whiteboard.
- Facilities giving assistance to the participants.
- Facilities around the capitalization of information.

3.2.1 Project Phase Support

Groupwork using preliminary design methodology involves several different phases which our system must support, i.e. brainstorming, functional analysis, risk analysis, causal analysis, and project planning. Each of these phases uses their own unique model to structure the content produced by the team, and effective preliminary design is contingent upon being able to quickly move or duplicate the content from one phase into another. For example, after brainstorming about the project's problem statement, the moderator may wish to move some of the ideas into a causal analysis for further refinement and exploration.

Then again, the moderator might instead wish to move the brainstorming from the table to the whiteboard in order to bring focus to the teamwork. The kinds of models used by the phases are diverse; for example, brainstorming typically results in labeled groups comprised of ideas, while causal analysis involves a hierarchical tree to represent the chain of causality which produces a problem. Providing this kind of functionality in our system requires an organized, coordinated way of translating and transferring the model of one phase to the model of another phase.

The first group of functionalities allows users to manipulate (create, delete, group, organize, etc.) the different types of objects that the phase models require. In fact, users manipulate graphical views associated to internal objects. The same object may have different graphical representations according to the phase in which it is involved. Every module implementing these functionalities on the table is developed with the same framework (MT4J) and the compatibility between elements is obvious.

Functionalities on the whiteboard are also developed with the MT4J framework. However, it is often useful to represent data on the whiteboard with other specific software, having their own data model and representation. For now, we have studied the interoperability with application where the model structure is highly described. For example, we are able to transform data for risk analysis phase into a Microsoft Excel document and transform data for brainstorming phase into FreeMind[1] mind maps.

3.2.2 Facilities Giving Assistance to the Participants

As we propose now to use interactive tabletop computing to perform preliminary design, instead of traditional tabletops and whiteboard, we may also introduce innovative ways to augment the work of the group in areas which were previously labor intensive. For example, in brainstorming, after the members have individually produced their ideas, they must now turn to the task of eliminating duplicates and categorizing the ideas into group. This task can be especially tedious in brainstorming with a large number of listed ideas.

The system may be able to assist with the detection of duplicates and assist the participants in sorting and grouping ideas and also by proposing potentially related ideas. In this sense, the system has a proactive activity. We consider that these kinds of functionalities must be untied of the first group and we develop specific modules installed on the small devices for that purpose. The drawback of this choice is that it is necessary for these modules to be aware of the situation of elements on the table, i.e. the phase models. The proactive functionalities are not limited to the current phase or the current project and may exploit any available data sources.

3.2.3 Capitalization of Information

Preliminary design groupwork is characterized by the coordinated efforts of experts from a variety of domains (e.g. mechanical engineers, project managers, usability experts and research scientists) in order to solve a problem. Over the course of the discussions, these experts may wish to access a variety of data from their different fields, such as international standards, patent information, photos of solutions from competing projects, or research papers. Moreover, reusing pieces of previous projects that have already treated similar tasks, risks, plans, etc. is a strategy often used by design teams. They must have access to previously completed projects, which are stored in the system's databases or knowledge bases.

The necessity of having a homogeneous system running on the interactive table leads to develop tools that allow participants to access document bases (mainly for pdf documents), the Internet and the base of projects built with our environment. Our system is also open because it is easy to add pieces of software on the small devices that can exploit these bases and communicate with the participant either in a subordinate way or a proactive way.

We consider that the participants of a project may also think about it whenever they are not around the table. They may have such private work sessions, collaborative or not. They must have at their disposal reporting documents about the work already done. That is why we have developed a tool that automatically transforms brainstorming phase model into pdf documents. Also, the interactive system should provide different ways of entering in the textual content of the models, for example allowing the user to use speech-to-text

[1] FreeMind site: http://freemind.sourceforge.net

functionality to dictate text or uploading a list that the user has previously written on a smartphone.

4. Organization of the System

All the functionalities presented in the previous sections show that a good organization of the system can be fulfilled by an agent system, more exactly by two multi-agent systems in charge of two main groups of functionalities and collaborating to support all the activities that a preliminary design project requires.

The first group of functionalities concerns the work itself around the interactive surfaces (work during the sessions and management of the created objects). The second group of functionalities is composed of the participants' assistance facilities: voice interaction and suggestions about the project content. In order to leave open a good scalability of the future features that could be added to our system, we disconnect this second group of facilities to the first one.

The first group is managed by a JADE[2] [6] multi-agent system. It is highly compatible with the pieces of software implemented using the toolbox MT4J[3] (Multi-touch for Java) [7]. The second group is managed by an OMAS[4] multi-agent system [8]. The left part of figure 4 shows the first agent system. The main container of agents lies on the table. The right part of figure 4 shows the second agent system. It consists of personal assistant agents lying on small devices. The agent system design methodology for building these systems is based on GAIA [9], JADE [10] and PASSI [11] methodologies.

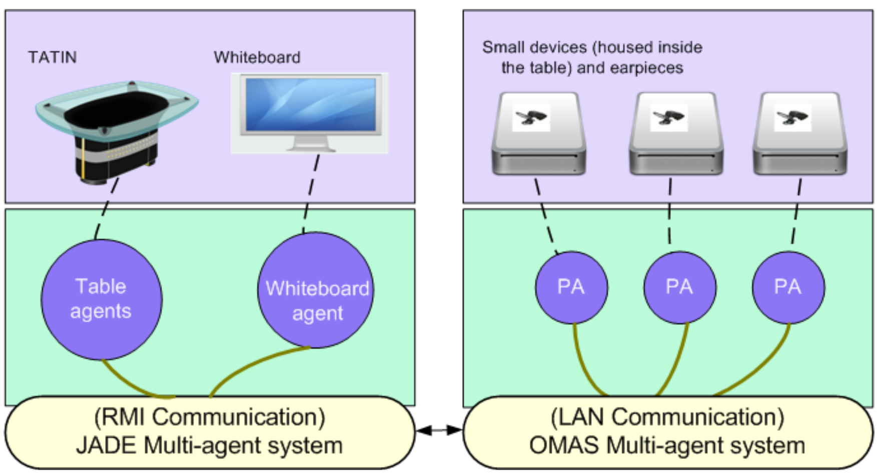

Figure 4: Agent Organization

4.1 JADE System

The JADE multi-agent system (see figure 5) is responsible for the management of the table and the whiteboard environments. Table agents are physically situated in the table station and the whiteboard agent in the whiteboard station. This decoupling of roles leaves it possible to use only the table interface during a session. Table agents are divided into three categories:

---

[2] JADE site: http://jade.tilab.com/
[3] MT4J site: http://www.mt4j.org/
[4] OMAS site: http://www.utc.fr/~barthes/OMAS/

- The table agent itself (TA): one agent for controlling table access.
- The phase agents (PhA): each one manages a specific phase of a project; they are five types of phases and thus five types of phase agents.
- The workbench agents (WbA): each one manages the elements created and manipulated by a participant; there is only one type of workbench agents.

During the bootstrap session on the table, the moderator chooses the project on which the participants will work and the platform agents is launched with the TA, the PhAs and the postman agent that will manage the exchange between the table and the PAs. The participants must login and the moderator after verification asks the system to register the participants to the open session. The WbAs are created just after the registration and their nicknames are the login that the participants have given.

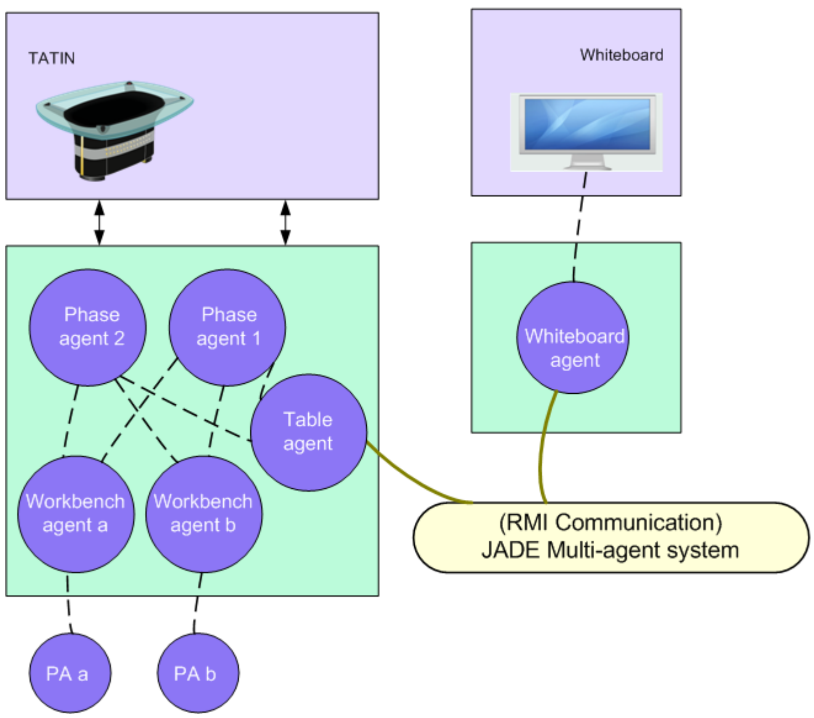

Figure 5: JADE Multi-agent System

The TA is in charge of the communication between the whiteboard and the table and transfers to the current phase agent, the requests of the whiteboard agent. Each phase is managed by a specific PhA. This agent is in charge of the phase model, i.e. the structure containing the elements manipulated during this phase and confers to the scene object (the graphical user interface) the role to display them on the table. The main communication role of such an agent is respond to the TA that asks for a specific view of a phase model. For example, a Brainstorming phase agent is able to send the brainstorming model under three forms:

- a serializable form: when the model of one phase has to be sent to the whiteboard agent or to a personal assistant agent (PA), it must be first serialized as a string. The receiver is in charge of exploiting this form for its own needs.

- a PDF document containing the content of the post-it notes elaborated during the corresponding phase. This PDF document is sent to the whiteboard agent that asks for the operating system of its station to display it. This can occur when the moderator, standing in front of the whiteboard wants the participants to focus on some aspects of the thinking that has been done. This PDF document helps the capitalization and reporting of the work done during a session.
- An XML format representing a mind map, compatible with the Freemind software.

A WbA is in charge of the elements manipulated by a specific participant. It is in relation with a PA. Voice commands concerning elements on the table and treated by a PA are transformed in texts and are transferred to the WbA that is in charge of their application on the table. For example, these commands could be for creating a post-it note, modifying the text on a post-it note. A WbA can also receive some suggestions that a PA could give to a participant to aid her with her tasks.

The JADE multi-agent system also contains a postman agent that operates the communication with the other multi-agent system. It waits for external agent messages on a specific socket and dispatches them to their intended receivers. A PA communicates with a WbA according to the names that have been given by the participant during the login step. It also sends answering messages to the outer agents. Messages exchanged between two different systems must respect a specific language (see section 4.3); however messages exchanged inside a platform respect the object format of this platform (ACL messages for JADE). The postman agent is responsible for transforming an outer message, which is a string, into an internal one and vice versa.

4.2 OMAS System

The OMAS multi-agent system has a twofold objective. Each participant to a project is supported by a PA and this system first links each personal assistant to a JADE Workbench agent. The second aspect is to link PAs between them and eventually to staff agents having specific skills that other agents could need. Some particular project topic may require specific knowledge bases and this multi-agent system is in charge of integrating the agents responsible for managing access to them.

The flow of information is the following: a participant asks to its PA to look for specific information using the voice recognition system. For example during a Brainstorming session, a participant may ask the PA to analyze the content of post-it notes, searching for duplicates of similar strings. The PA asks the corresponding WbA to send the model of the phase, receives it and filters it. It then can inform the participant of actions to do.

If the PA is not able to give an appropriate answer, it transfers the request to staff agents that may question their knowledge bases. The answer is sent back to the PA that informs the participant. If no answer is available after a while, the participant is also warned.

It is also possible to attach PAs which have skills not shared by the other PAs. In this case, they are not considered as staff agents but may answer to requests coming from any participant.

When a session starts, the participants are around the table, but they are not identified by any technological system or device. The session boot allows each participant to enter a login that is communicated to a workbench agent. Each participant must also be recognized by a PA running on a small device. Once the identification is done, a PA asks the TA for the workbench agent having the same login as itself. An identification error may be detected. This approach also allows a participant to choose whether or not to use a PA (some sessions may not require such a support) or to activate the agent at any moment of the work session. In fact, the environment supporting a session may have different levels of complexity. They correspond to the type of activities that the moderator intends to conduct during a session. It may consist of the table agents with or without personal agents, with or without the whiteboard agent. Though the whiteboard cannot be used alone, it can be activated at any moment of a session.

4.3 Exchange Format

The architecture described in the previous sections has some advantages. Each multi-agent system has its own role. The JADE system helps to manage the activities on the table and the whiteboard. The OMAS system manages the activities which directly assist the participants. Both multi-agent systems may exploit the information system that capitalizes the projects done with the multi-touch table. Internally, they use their own structure and dispatch the messages.

We have to consider messages exchanged inside one agent system and messages exchanged between both systems. Inside the JADE system, messages respect the corresponding format but sometimes the core of messages must contain the model of a phase. For example, it is necessary to serialize it when a phase agent sends its model to the whiteboard agent. The model is first transformed into a serializable object and then into a JSON string that is inserted in the message (see figure 6). Any JSON library is able to transform such an object into a JSON string.

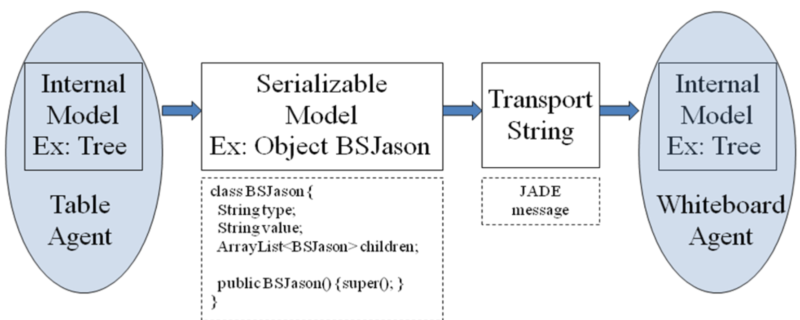

Figure 6: Transformation of Models

The main drawback of the architecture is the need to define an exchange message format for the communication between the multi-agent systems. All the actions that a workbench agent performs, either under the direction of the table, or to inform a PA of an event, or for answering a PA request, have to be described in detail and the exchange format has to define all parameters. However, a PA never directly controls an action to be done on the table. There are different possibilities: a participant may ask a PA to send a message to the workbench in order to perform an action; a PA may suggest to the participant some action to do using the voice recognition system and if the participant accepts, it sends the command to the workbench agent.

Complex questions can be asked to a PA in natural language (or near natural language) according to the strength of the agent analyzer. For now our analyzer is limited to the study of short sentences and a PA has not yet learning abilities like some meeting assistants can have [12]. As an answer, a PA can suggest to the participant to perform an action using a toolkit available on the table. For example, a participant can ask to a PA: “post-it notes of projects about microlight” or “post-it notes about microlight” and the PA can answer with the reference of the project that most closely matches the request. The participant can then open with the PDF reader tool the PDF file that is automatically created when a project is saved.

The exchange format follows the JSON[5] grammar and syntax. A message is considered as a JSON object whose properties are in accordance with FIPA[6] standard. These properties are: sender, receivers, profile of receivers, creation date, performative, language, IP address and port of the sender host, identifier of the message, task identifier and content. The main performatives used during exchange are cancel, answer, error (for failure and not understood), inform and request. The content is also a JSON object whose main properties are:

- action: a symbol corresponding to a skill or a behavior
- args: list of parameters useful when the agent performs the action
- contents: JSON structure corresponding to an answer
- error-contents: explanation when an error occurs

The following example shows the content of a message requesting for the creation of a group of post-it notes and for adding some post-it notes to this group.

```
{"action" : "create",
 "args" : {"category" : "group",
           "content" : "Structure",
           "ref" : ["p3-2-1-1"]}}
```

Parameters depend on the nature of the action requested. The content argument is the text to be inserted in the post-it note and the ref argument is the post-it note list to be added to the group created.

5. Conclusion

We have presented in this paper the main functionalities and the agent based architecture of an environment that aims to support the work of experts during the collaborative preliminary design phases of a project. The environment includes a multi-touch interactive table, an interactive whiteboard, and also smaller personal devices such as laptops, tablets or even PDAs. This environment allows the participants to a project to use different devices configurations and the system may accept different levels of complexity. To address this complexity we propose an agent-based architecture, relying on two muti-agent systems, JADE and OMAS. The JADE agents are in charge of the collaborative activities on the table and the OMAS agents are mainly personal agents supporting vocal interactions with the users and giving them some assistance. This architecture provides clarity and flexibility, with each system having its own role. The counterpart is the need to define an exchange message format for the communication between the multi-agent-systems. A first prototype of this multi-surface and multi-modal environment is already functional.

Our future works include the development of efficient personal assistant agents in order to give a real assistance to the participants. They also include the study of different gestures and modalities needed to ease the preliminary design process and the design of efficient personal agents. These gestures must be as natural as possible in order to avoid the participants to learn

[5] http://www.json.org/
[6] FIPA: http://www.fipa.org/

how to use the software. We have planned a large series of experiments to verify the efficiency of the system and in order to correct and enhance the usability of our system and to implement the necessary adaptations it requires.

We consider that the system is generic enough to be used for preliminary design in different applications domains. Experiments will also be conducted to verify this approach. The architecture based on multi-agent systems should also allow the interoperability with other applications and in particular with indexing systems.

Acknowledgments

The TATIN-PIC project is co-funded by the European Union and the French region of Picardy. Europe [7] is committed to the Picardy Region with "Le Fond Européen de Dévelopement Régional" (The European Fund for Regional Development). The authors would also like to thank Thierry Gidel and Atman Kendira for their active participation to the TATIN-PIC project.

References

[1] J. Rekimoto, and M. Saitoh, Augmented surfaces: a spatially continuous work space for hybrid computing environments, Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '99, 1999.

[2] T. Gjerlufsen, C. Klokmose, J. Eagan, C. Pillias, and M. Beaudouin-Lafon, Shared substance: Developing flexible Multi-Surface applications, ACM Conference on Human Factors in Computing, CHI 2011, 2011.

[3] S. Shiba, and D. Walden, Quality process improvement tools and techniques, Technical report, MIT, 2002.

[4] NUI Group. Multitouch Technologies, http://nuicode.com/projects/wiki-book/files, 2009.

[5] J. Schöning, P. Brandl, F. Daiber, F. Echtler, O. Hilliges, J. Hook, M. Löchtefeld, N. Motamedi, L. Muller, P. Olivier, T. Roth and U. Zadow, Multi-Touch Surfaces: A Technical Guide. *Technical University of Munich*, Munich, Germany, 2008.

[6] N. Spanoudakis, P. Moraitis, An Ambient Intelligence Application Integrating Agent and Service-Oriented Technologies, 27th SGAI International Conference on Artificial Intelligence (AI2007), Peterhouse College, Cambridge, UK, December 10-12, 2007.

[7] U. Laufs, C. Ruff and J. Zibuschka, MT4j – A Cross-platform Multi-touch Development Framework. Engineering Patterns for Multi-Touch Interfaces 2010, Workshop of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Berlin, Germany, 2010.

[8] J.-P. Barthès. OMAS - a flexible multi-agent environment for CSCWD, 13th International Conference Computer Supported Cooperative Work in Design (CSCWD), Santiago, Chile, April 21-25, pp. 258-263, 2009.

[9] M. Wooldridge, N. R. Jennings, D. Kinny, The Gaia Methodology for Agent-Oriented Analysis and Design, Journal of Autonomous Agents and Multi-Agent Systems, 3(3), pp. 285-312, 2000.

[10] M. Nikraz, G. Caire and P. A. Bahri, A Methodology for the Analysis and Design of Multi-Agent Systems using JADE, http://jade.tilab.com/doc/JADE_methodology_website.

[11] B. Henderson-Sellers, J. K. Debenham, Q.Numi Tran, M. Cossentino and G. Low, Identification of Reusable Method Fragments from the PASSI Agent-Oriented Methodology, Agent-Oriented Information Systems (A0IS), pp. 95-110, 2005.

[12] G. Tur,, A. Stolcke et al., The CALO Meeting Assistant System, IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, N°. 6, pp. 1601-1611, August 2010.

[7] The content of this paper is the sole responsibility of the authors and in no way represents the views of the European Commission or its services or the Picardy region.

[13] M. Micire, M. Desai, A. Courtemanche, K. M. Tsui, and H. A.Yanco, Analysis of natural gestures for controlling robot teams on multi-touch tabletop surfaces, ACM International Conference on Interactive Tabletops and Surfaces (ITS '09), Banff, Alberta, Canada, pp. 41-48, 2009.

[14] T. Gidel, A. Kendira, A. Jones, D. Lenne, J.-P.Barthès and C. Moulin, Conducting Preliminary Design around an Interactive Tabletop, international conference on Engineering Design, ICED 2011, Denmark, 15-18 August, 2011.

[15] A. Jones, A. Kendira, D. Lenne, T. Gidel and C. Moulin, The TATIN-PIC Project, a Multi-modal Collaborative Work Environment for Preliminary Design, 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2011), June 8-10, Lausanne, Switzerland, 2011.

[16] M. R. Morris, Benefits and Challenges of Tabletop Peripheral Displays, Adjunct Proceedings of the Sixth International Conference on Ubiquitous Computing (UbiComp 2004), September 7-10, 2004, Nottingham, England, 2004.

[17] F. Enembreck, and J.P. Barthès, Improving CSCW with Personal Assistant Agents, Journal of Integrated Design & Process Science, vol. 7, no. 2, pp. 3-19, 2003.

[18] F. Heidrich, M. Ziefle, C. Röcker and J. Borchers, Interacting with smart walls: a multi-dimensional analysis of input technologies for augmented environments, 2nd Augmented Human International Conference (AH 2011), Tokyo, Japan, pp. 1-8, 2011.

Authors

Claude Moulin

PhD in Computer Sciences since 1998, Claude Moulin has been working for four years in research centers in Italy. He is now Assistant Professor at the University of Technology of Compiègne (France) and member of the laboratory HeuDiaSyc (Information Knowledge and Interaction team). He has a background in Knowledge Base System, Knowledge Engineering and web service semantic description, ontology design and formalisms. His current research interests include Multi-Agent systems and interaction with multi-modal Interfaces.

Alistair Jones

After receiving his Bachelor and Masters in Computer Science from the Georgia Institute of Technology in 2008, Alistair Jones has been studying usability and human-computer interaction. He received a second Master's degree from the University of Technology of Compiegne (UTC), and he is currently a PhD candidate at the UTC researching interaction in multimodal and multi-display ecosystems for engineering design.

Dominique Lenne

Dominique Lenne is professor of computer science at Compiègne University of Technology (France). His research interests include knowledge modeling, human computer interaction and interactive learning environments. He is involved in several projects aiming at embedding knowledge in intelligent interactive systems. He has published many papers in journals and conferences in theses domains. He is also member of several related research communities.

Jean-Paul Barthès

Jean-Paul A. Barthès obtained his Engineering degree from Ecole Centrale de Paris (France) and his Ph.D. from Stanford University (1973). Currently, he is a professor in the department of Computer Science at the University of Technology of Compiègne in France. His main research interests are related to Distributed Artificial Intelligence and mixed societies of cognitive artificial and human agents. He is also the chairman of the IIIA Institute, a European program for helping large companies to cope with the problems of Knowledge Management.