Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design

Personal Assistant Agents and Multi-agent Middleware for CSCW

Alistair Jones, Claude Moulin, Jean-Paul Barthès, Dominique Lenne
Heudiasyc laboratory, UMR - CNRS 6599
University of Technology of Compiegne
Centre de Recherche de Royallieu
60205 COMPIEGNE Cedex
Email: [alistair.jones | jean-paul.barthes
| claude.moulin | dominique.lenne] @ hds.utc.fr

Atman Kendira, Thierry Gidel
COSTECH laboratory, EA 2223
University of Technology of Compiegne
Pierre Guillaumat
60203 COMPIEGNE Cedex
Email: [atman.kendira | thierry.gidel] @ utc.fr

***Abstract*—In recent years, there has been an increased interest for research on computer-supported cooperative work performed in collaborative interactive spaces. The TATIN-PIC project envisions a true multi-surface collaborative work environment with an interactive tabletop, an interactive board display, tablet PCs, and smartphones. In this paper, we first present the middleware based on a multi-agent architecture which use in our implementation, and then we detail how voice-controlled personal assistant agents can be implemented to provide unique interactions within a multi-surface environment.**

I. Introduction

Work environments supporting collocated collaboration have become an increasingly popular field of research in recent years. Their application is often focused on performing conceptual design work, such as brainstorming or project planning. The first goal of such systems is to provide a method for saving the content of collaborative work and distributing the data throughout the team. This goal is often directly at odds with the secondary goal of the system: engendering communication and collaboration. Indeed, the devices for capturing input data from the team are same ones which inhibit the productivity of the group. For example, several meeting systems have been proposed to facilitate brainstorming by having users at separate desktop workstations with users focusing primarily on their computer workstation. These systems diminish group awareness and collaboration because the system naturally discourages simple face-to-face communication. Ultimately, creativity and productivity are negatively impacted.

In response to the limitations introduced by the traditional hardware repurposed for collocated groupware, researchers are capitalizing upon the increased availability of vertical and horizontal interactive surfaces to fashion interactive boards and tabletops capable of supporting a social framework more advantageous for group collaboration. The TATIN-PIC (from the french: TAble Tactile INteractive - Plateforme Intelligente de Conception) project at the University of Technology of Compiegne continues with this approach in order to build a collaborative multi-display working environment for design and engineering teams (Fig. 1). To address the problems typically associated with multi-display ecosystems, we provide in this paper a detailed description of the middleware we have design for the system. The middleware is a multi-agent system which provides ample extensibility and stability to programmers when building applications for our interactive environment. Because we use a multi-agent architecture, we can easily integrate voice-controlled personal assistant agents which aid the users in performing tasks on the interactive tabletop.

Fig. 1. The hardware of the TATIN-PIC system: An interactive tabletop, an interactive board, tablet-PCs, smartphone and headsets. (Not pictured: MiniPCs housed inside the interactive tabletop for the personal assistant agents)

The following section will present the TATIN-PIC platform, and how we were influenced by preliminary design and related work for the design choices of system. Next, we describe TATIN-PIC's implementation of a multi-agent system for effective middleware in a multi-display work environment. With a multi-agent system in place, we propose new experimental interactions with personal assistant agents and demonstrate how new technology can be integrated within our system with minimal overhead.

II. Preliminary design in TATIN-PIC

Though we have previously written on the subject of the preliminary design phase in engineering projects [1], we summarize the results here. The preliminary design phase is based on recent developments in design methodologies and allows an engineering team to make more cost-effective decisions earlier in the project. Over the course of a preliminary design phase, the engineering team moves from a problem statement given from a client or a director, and produces a detailed project plan which will lead to the development of a solution to the problem. Preliminary design teams are typically a multi-disciplinary team with members from different departments inside an organization (engineering, human resources, direction, research, sales) and can involve participants with various levels of computer proficiency. Preliminary design uses creativity tools which emphasize collaboration and consensus and which focus on concepts and ideas rather than detailed drawings and plans (e.g. AutoCAD designs). The four tools prioritized in the TATIN-PIC project are detailed below.

1) Brainstorming: A brainstorming session begins when a moderator gives a central problem or question to the participants. The moderator will then lead the participants through three steps. First, the participants generate as many ideas as possible around the central problem or question, writing each idea separately on individual Post-it notes. Then, participants each take turns presenting and explaining their ideas to the group. Finally, participants collaboratively group related ideas, and create labels for each group. Participants are encouraged to discuss the groupings and try several different possible categorizations. The resulting data of a brainstorming can be modeled as a hierarchical three-level tree, with the central problem or idea at the root, the group labels at the 2nd level, and the individual ideas at the 3rd level. This exercise is designed to promote divergent thinking by the participants and encourages the exploration of the solution space of a problem.

2) Causal analysis: A causal analysis begins when a moderator presents a problem to the participants. The participants must generate possible causes and effects to answer why this problem occurs or what the consequences of this problem are. The participants use the causes and effects (again each written on different Post-it notes) to collaboratively build a bow tie diagram. Different causes and different effects are linked together in order to show the chain of causality. Once participants have agreed upon a final bow tie diagram, the team can discuss strategies for eliminating a problem, either by removing one of its causes or nullifying its effects.

3) Risk analysis: At the beginning of a risk analysis, the moderator will identify a set of critical points which concern the project. From these critical points, the teams generates possible risks to the project. After all the risks are generated they are listed row-by-row in a spreadsheet. The columns of the spreadsheet are labelled ad hoc. For example, the columns frequently contain such entries as "Probability of occurrence of risk," "Probability of non-detection of risk," and "Seriousness of risk," which are then used to calculate the "Criticality" of each risk. After the spreadsheet is complete, the team now can discuss additional steps or functions that must be put in place during the course of the project to treat each risk.

4) Project planning: At the beginning of a project planning phase, each member of the team will generate the tasks that they (or their department) must complete over the course of the project. These tasks are manipulated in flow-chart form, where the group must collaboratively decide upon the chronology of the tasks over the course of the project. After the entire flowchart of the project is built, the group decides upon the duration of each task. With chronology and the duration of all the tasks established, the group is able to view their project planning in gantt chart form.

From the descriptions of the four tools described above, two reflections can be made.

- The tools each begin with a form of content generation (ideas, causes, effects, risks, and tasks) which emphasizes divergent thinking.
- The tools each force consensus and convergent thinking, as the team members must finalize a coherent diagram (hierarchal tree, bow tie diagram, spreadsheet, and flowchart, respectively) which they all agree upon.

These reflections have identified two design considerations for the TATIN-PIC platform. First, we have generalized the procedure of the brainstorming session in order for it to be the first step in a casual analysis, risk analysis, and project planning. Under our proposed approach, a brainstorming session can be used to generate the causes, effects, risks, and tasks, which can be transferred into bow tie diagrams, spreadsheets and flowcharts to facilitate the construction of these information models. Second, the physical design of our interactive space must be able to handle both divergent thinking and convergent thinking in order for the team to experience the proper dynamics of collaboration required by these tools. The physical design must also provide a universal and democratic way for participants to interact with the system, which can allow a heterogeneous design team to communicate ideas and concepts amongst themselves without inhibition. These design consideration are addressed in the following section.

III. Design of TATIN-PIC platform

A. Interactive collaborative surfaces

The recent popularity of interactive *tabletops* and *board displays* for groupware can be attributed to the advantages they provide to face-to-face meeting as opposed to groupware on desktops, laptops, and other devices. Rogers and Lindley [2] explain the differences between such horizontal displays and vertical displays for groupware: the tabletop surfaces usher face-to-face collaboration, prompting opportunities for the sharing and discussion of new ideas, while the interactive board promotes shoulder-to-shoulder collaboration, where participants focus and reflect upon the content. Coupling the displays with a data connection allows designers to build a multi-surface work environment. The tabletop and board respect human-to-human interaction in this space: their large shared public screen space allows users to use computing systems in meetings without obstructing natural verbal and visual communication as opposed to desktop and laptops. These surfaces frequently use multi-touch technology for direct interaction with the diagrams; this unencumbered interaction allows for spontaneous interaction by all members of the preliminary design session. Because the gestures are a much more visible method of interaction, new users can easily copy and understand interaction techniques performed by expert users.

Following this research, at the core of the TATIN-PIC platform, is a multiuser interactive tabletop surface with a data connection to a multitouch interactive board display. The interactive tabletop is the platform on which the participants generate and initially organize data in brainstorming sessions because it promotes divergent thinking. The interactive board provides a complementary role; the form factors at play here can nurture both shoulder-to-shoulder collaboration, which encourages convergent thinking. In related work, there exists two examples of brainstorming applications which couple an interactive vertical display to a tabletop surface to promote "group awareness" [3] and be a "shared reflection space" [4]. Our strategy for the interactive board is different; after the brainstorming is completed on the tabletop, the categorized Post-it notes from the brainstorming session are transferred to the interactive board and work continues in either the project planning, risk analysis, or causal analysis phase. Because the content is the same from the brainstorming to the next phase, we hope this shift will still promote group awareness and shared reflection space, but our main goal is to move from face-to-face to shoulder-to-shoulder style interaction to help converge the collaborative effort into the production of coherent diagrams. Some information, such as the content of the Post-it note, or metadata concerning a task, can still be modified in real time from the tabletop. Selecting a set of multitouch gestures can be problematic for such large multitouch surfaces, therefor many of our design choices in this area are influenced by the set of user-defined gestures for interactive tabletops as specified by Wobbrock et al. [5].

B. Multimodality and personal assistant agents

Though, multitouch interaction alone will be enough to collaboratively construct diagrams on the interactive tabletop surface and the interactive board, tactile large-surface interfaces suffer from certain common user interface problems. Firstly, the greater the number of gestures, the more complicated and intricate they become. This is especially the case with offline gestures, when a predetermined path is drawn with a finger to trigger a specific action, and the processing of the gesture occurs after the gesture is finished. The user quickly becomes overloaded with a large number of gestures, with little visual aid to help them recall such gestures. Secondly, touch-based user interfaces suffer from clutter [6]. The reasons for this can generally be attributed to a lack of contextual menus attached to UI widgets for hiding context-specific options (as typically seen in traditional WIMP interfaces). Exacerbating this problem, is that buttons that are usually placed on the UI widgets are slightly larger in size than on traditional WIMP interfaces (to be easily touched with a finger). This means there is a tendency to have more buttons and larger widgets which occupy more screen real estate. We hope to lessen the burden carried by touch interfaces by providing multimodal (touch and voice) interactions with headsets.

Adding the modality of voice can provide a large range of functionality without burdening the visual user interface. The modality of voice alone can be used to perform certain commands such as creating a new Post-it note, when the user speaks "Create a new Post-it". However, by intertwining the modality of voice with multitouch interaction, we can create a much more powerful and expressive set of multimodal interaction techniques. In this case, the design strategy of TATIN-PIC is to design our multimodal interface with a complementary strategy, allowing each modality to perform the task at which it is better at expressing. More specifically, if we look at the speech-augmented multitouch interaction patterns presented by Schnelle-Walka and Dowling [7] , we are hoping to use such patterns as:

- *Auditory Mode Switching*: When performing an action the user can vocally specify more information to effect a different outcome. For example, a user can move a Post-it note by dragging it across the screen. A user can copy a Post-it note by performing the same action, but saying, "Copy my note" at the same time.
- *Select by Touch, Operate By Voice*: This allows a user to specify the target, and an action using two modes which complement each other. For example, a user can select a Post-it note on the tabletop by tapping on it and saying "Change color to blue."
- *Voice as Text Input*: For example, a user can perform actions on a Post-it note by saying "Edit the text of this Post-it <select by touch> with the content 'Update market plan.'"

In order crystalize our design strategy for multimodal interaction, we have embodied the above functionality inside *personal assistant agents*, which can interact with the user with speech-to-text and text-to speech modules in the headset. The added benefits of personal assistant agents are numerous. They give the user a virtual entity that they may address, allowing the user to give the impression that the are talking to someone. Also, they are able to give feedback and initiate a dialogue in order to ask for missing information in the command. In related work, Tse et al. [8] also demonstrated multimodality with a multi-user brainstorming tabletop prototype. This research gives credence to the idea and we hope to further contribute to this area of research by implementing personal assistant agents in multi-surface collaborative space.

C. Secondary surfaces

The last components of our interactive space are tablet PCs and smartphones. There are three main motivations for including such device in our collaborative space. First, recall that preliminary design teams often consist of experts from different domains, who bring to a meeting resources and references which apply directly to their fields (e.g. patents, industry standards, research papers, etc.). We are currently exploring interaction techniques to allow the dynamic transfer of this information from the user's personal tablet PC and smartphones to the interactive tabletop (the current TATIN-PIC prototype only allows user to transfer and display PDF documents by USB key on the interactive tabletop). The second motivation for including tablet PCs and smartphones concerns their virtual keyboards. We have seen in our own previous evaluations that users have difficulty entering text into the virtual keyboard on the tabletop surface. They are, however, well accustomed to the virtual keyboards on their smartphones and tablet PCs, which take advantage of personalized corrective text prediction to facilitate text entry [9]. By accepting text entry from users' tablet PCs and smartphones, we are allowing users to bring their own personal keyboard to these meetings. Our current prototype on tablet PCs allows participants to create and modify virtual Post-it notes. This functionality is also related to our third motivation for including these displays. The content generating step of a brainstorming can suffer if the form factors of the environment contribute to production blocking of ideas. Certain users may feel shy or inhibited when writing ideas out on a large collaborative display. Including tablets and smartphones can give a more private surface on which users can write out their ideas before transferring them to the collaborative surface.

With the addition of tablet PCs and smartphones, our proposed solution is classified as a multi-display eco-system [10], and is a veritable realization of Mark Weiser's ubiquitous computer vision of "tabs, pads, and boards" [11]. We mention the tablet PCs and smartphones here, not only because they impact the design and collaborative nature of our environment, but also because their inclusion has been taken into the middleware of TATIN-PIC. This middleware is built using a multi-agent architecture, to facilitate the inclusion of such heterogenous devices in that TATIN-PIC system, and to allow us to comfortably integrate our personal assistant agents alongside.

IV. Multi-agent Architecture for Collaborative Spaces

We have described a system where multiple users interact with a variety of surfaces to construct different data models in different phases of preliminary design. This has certain implications for the design of our system's architecture. Specifically, we require a middleware capable of supporting multiple users, heterogeneous devices, heterogeneous data models, and robust run-time connectivity between components. The TATIN-PIC project uses a multi-agent system (MAS) to handle communication between different aspects of the graphical user interface and interactive surfaces. Here, we introduce certain elements of our user interface, and then describe how our middleware handles such components.

Fig. 2. The workbench agents are tied to each user, and are responsible for the widgets the user creates and interacts with. Here, these widgets include the circular menu, from which the user can open PDF files on the system and virtual keyboards to create virtual Post-it notes.

A. I/O for interactive surfaces in TATIN-PIC

The interactive tabletop surface uses infrared Laser Light Plane technology [12]; when a user places a finger on the tabletop, he breaks the laser light plane and the infrared light in captured by cameras connected to a computer underneath the table. The input from the cameras is treated with an image processing and object tracking application called Community Core Vision. CCV sends touch events as described by the TUIO protocol to a specified port on the computer. The graphical user interface of the tabletop is built using the MT4j toolkit [13], a multitouch library designed in part for human-computer interaction research. An MT4j application has global input processors and gesture processors which listen for TUIO touch events on an open socket and transform them into touch events in Java so that they may be used by user interface widgets. Core UI widgets (Fig. 2) on the interactive tabletop include:

- Post-it notes which may be grouped and labelled. They are the basic building of a preliminary design session.
- Virtual keyboards from which users may create and modify virtual Post-it notes.
- Circular menus from which users may open their keyboards and PDFs. The moderator's circular menu has extra functionality, e.g. sending the brainstorming from interactive tabletop to the interactive board, opening and closing different phases. The users login at the start of the meeting to have their own personal menus.

The I/O infrastructure is similar for the TATIN-PIC interactive board. An infrared LED frame is overlaid on an LCD screen and connected to its own dedicated computer station. The driver for the LED frame sends TUIO events to an MT4j application running on the board, and this allows users to interact with on-screen widgets by touch. The interactive board in TATIN-PIC will be used by one user at time, typically the moderator of the preliminary design session. There is one circular menu on the board which is responsible for opening new phases (either risk analysis, causal analysis, or project planning) and pulling content from the brainstorming session on the tabletop to the interactive board.

B. MAS Middleware in TATIN-PIC

Our TATIN-PIC platform uses two different multi-agent toolkits, JADE (Java Agent DEvelopment Framework) [14] and OMAS (Open Multi-Agent System) [15]. The JADE agents easily integrate directly into the MT4j applications on the tabletop and interactive board (since all libraries are java-based). The OMAS toolkit allows us to integrate personal assistant agents and is discussed in more detail in the following section.

One of the primary functions of the JADE multi-agent system in TATIN-PIC is to handle the communication between the tabletop and the interactive board. Both the tabletop and the board have their own dedicated computers, from which a moderator will launch the TATIN-PIC software independently. This gives the participants the freedom to use the different surfaces as necessary. For example, after a brainstorming is completed on the tabletop, the moderator may decide to display the data in a different form on the board. The moderator can then turn on the interactive board and launch the TATIN-PIC application to send the data to the board. There are four different agents that will be implicated in such an action:

- A **table agent (TA)** for sending and receiving messages to and from the tabletop computer.
- One **phase agent (PhA)** per phase on the tabletop computer for managing the information models associated with each phase. There are four types of phases (brainstorming, causal analysis, risk analysis and project planning) and thus four types of phase agents.
- One **workbench agent (WbA)** per user on the tabletop computer for managing the elements created and manipulated by a participant, such as virtual Post-it notes, as well as connections from outside devices such as tablet PCs and headsets.
- A **board agent (BA)** on the interactive board for sending and receiving messages to and from the tabletop computer.

When the TATIN-PIC application is launched on the tabletop computer, the participants of the meeting are greeted with an initial login screen and the table agent is activated. After each user logs in, their workbench agents are all immediately activated. Once the moderator of the session creates a new brainstorming phase, the phase agent is activated. These agents are pictured in Fig. 3. All messages from the BA are transmitted directly to the TA, who is in turn responsible for transmitting the message to PhA associated with the phase currently open and active on the tabletop. The PhA has access to the corresponding model of each phase (e.g. the tree of ideas from the brainstorming) and can send this when requested. For example, a brainstorming PhA is able to send the brainstorming model under three forms:

- a serialized object: When the model of the brainstorming phase has to be sent to the board agent, it must be first serialized as a string. The board agent can then use the object to display on the interactive board.
- 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. This can be opened directly by the software similar to the example of the PDF.

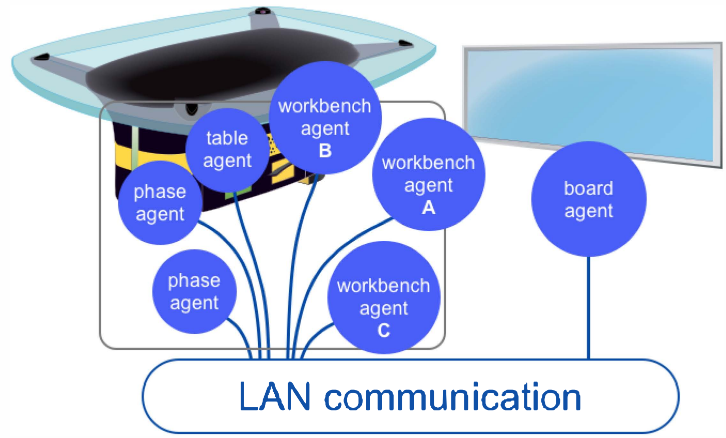

Fig. 3. JADE agents of the TATIN-PIC system: one Table Agent for the tabletop, one Workbench Agent for each of the users A, B, and C, two Phase Agents for brainstorming and project planning, and finally one interactive board agent.

This means that after a brainstorming is completed by the team, the moderator can turn to the interactive board and request the model of the brainstorming to be displayed in the form of a project planning flowchart. In this case, the BA sends a request to the TA for the current brainstorming model. The TA transfers this request to the corresponding PhA, who serialized his model and send it directly to the BA. The BA then displays the tasks on the screen, and opens the file it has received. Now with one set of Post-it notes on the tabletop in a brainstorming session, and the identical set of Post-it notes copied to the board in a project planning flowchart, the users can edit the metadata of Post-it notes (i.e task duration) on the tabletop and the metadata will be immediately changed on the interactive board. In this case, the PhA agent on the interactive tabletop detects a change has been made to the model and sends an updated model to the TA, who transfers it to the BA.

C. Discussion

Our middleware takes a novel approached for multi-surface collaborative environments. The design we propose for our architecture is a multi-agent system [16] developed using

JADE and OMAS agents. This message driven architecture has proven to be adequate for our development-time and run-time user-experience requirements. At development time, MAS systems are advantageous for their highly extendable infrastructure; since agents can transfer information by broadcasting messages to the other agents connected on the same network, adding support to the environment for an input device is a matter of adding an agent responsible for the device. This allowed us to easily incorporate the tablet PCs in our implementation. JADE and OMAS agents can broadcast messages, which have no fixed recipient, and every agent hears every message. Agents only respond to a request by determining if they can treat the request, and if so, they perform the associate task and send back a response. In this regard, multi-agent systems can be designed for robustness: no agent is responsible for the complete control over the entire functionality of the system.

A properly designed system allows for the devices (and their corresponding agents) to be connected and disconnected will little cost to the performance of the system. An agent will broadcast messages and wait for a specified amount of time to hear responses. If a device is turned off in the middle of a session, the agent representing the device will no longer be connected to the network. Other agents wishing to address this agent with a broadcast message could either have their message handled by another agent, or timeout waiting for a response. Either way, the sending agent will still be functional and will still be able to adjust accordingly to the situation.

Gjerlufsen et al. [17] provide a summary of existing middleware systems for interactive environments. The advantage that multi-agent system will provide over existing systems is that agents can merge seamlessly with the application layer of our system architecture as well. For example, our interactive tabletop can have agents representing each user; each phase, each data model, and these agents will be able to communicate directly with other agents on other devices as if they were fully integrated within the same architecture. This brings homogeneity to our distributed architecture.

V. TATIN-PIC Personal Assistant Agents

In this section, we describe how personal assistant agents from the OMAS platform are joined with the JADE agents to form a multi-agent middleware capable of managing multimodal input.

A. OMAS personal assistant agents in TATIN-PIC

OMAS is a Lisp toolkit which is designed for the rapid prototyping of intelligent agents. For example, OMAS agents have a special dialogue engine and associated task library. Each task has several associated keywords ranked by importance. These keywords are weighted numerically from -1 to 1. When text is received as input into their dialogue engine, an n-gram analysis is performed on each of the words, and the most closely associated task is triggered. These tasks can have their own finite-state-machine sub-dialogues.

Personal assistant agents (PA) in OMAS are agents capable of natural language dialogue with users. These agents run on their own dedicated miniPC which is housed underneath the tabletop. Each user has a wireless bluetooth monaural headset capable of recording the user's voice and playing sounds into the user's ear. When a user speaks into his microphone, the audio input is immediately analyzed using text-to-speech software (Nuance's Dragon NaturallySpeaking). A user can activate their microphone by saying "Wake up" and deactivate it by saying "Go to sleep". When the user has his microphone activated, the resulting text is then passed to the personal assistant agent in OMAS. For example, a user can manually activate his headset and, speaking into his microphone, say "Create a new Post-it period." The text-to-speech software translates these spoken words into the string "Create a new Post-it." The PA detects the end of phrase indicator, which is the period punctuation, and performs an n-gram keyword analysis of the sentence. Because the sentence contains the keywords "create", "new", and "Post-it", the personal agent performs his new-postit task. This task begin by triggering the sub-dialogue associated with Post-it creation. The response of the personal agent is "What is the content of the Post-it?". This string is passed to a text-to-speech program written using the Microsoft Speech API, and the agent speaks directly into the ear of the user. The user then replies "Detailed work plan period," to which the agents can extract "Detailed work plan." The personal agent now has all the information required to create the content of the Post-it.

OMAS agents can freely exchange messages between themselves even on different platforms. The PA can also delegate tasks to its own staff agents, who have more specialized abilities. Because the PA relies on his staff to do most of the work, the PA acts as a "digital butler," a term introduced by Negroponte [18]. In our system, each PA has a dedicated staff agent for managing Post-its. Other supporting staff agents could include specialized agents which manage access to specific knowledge bases. For example, during a Brainstorming session, a participant may ask the PA to analyze the content of all the Post-it notes on the table, searching for any duplicates. A PA's staff agent can ask the corresponding WbA to send the model of the phase, and upon receiving the model he may scan it for duplicates and transfer the results back to the PA agent who informs the participant of which actions to do. If the service agent is not able to give an appropriate answer, it transfers the request to other staff agents that may question their knowledge bases. If the PA reaches a time-out limit while waiting for a response, he can warn the participant of the error.

B. OMAS assistant agents and the JADE middleware

The PA of each participant is also linked to the corresponding workbench agent in the JADE platform. When a session starts, the participants are gathered around the tabletop, but they are not identified by any technological system or device. The initial login screen allows each participant to enter a username, which is then associated to an activated WbA agent. Our PAs are custom built with the usernames of the participants, and once the identification is done on the tabletop, a PA asks the TA for the workbench agent having the same login as itself, and a link is created between the two agents. On the interactive tabletop, a user can tap a Post-it note to select it, making it the subject of all future vocal commands. The user's WbA on the tabletop informs the PA agent of the selection action and the PA stores the IDs of the Post-it note in a salient features queue. This allows users to perform multimodal interaction techniques, such as "Delete this <select by touch> Post-it note" and "Modify the text of this <select by touch> Post-it to "Business plan'". Users can also use their circular menu to open a window which shows the results of the voice recognition algorithm. This echo window allows the users to confirm that the agent has correctly understand them. In this case, a personal assistant agent continuously informs a workbench agent of input.

OMAS agents can exchange messages with the agents of the JADE system. 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 board agent. The model is first transformed into a serializable object and then into a JSON string that is inserted into the message. Any JSON library is able to transform such an object into a JSON string, whether the library be written in Java or in Lisp. The JADE multi-agent system also contains a *Transfer Agent* (or Postman Agent) that operates the transformation of one message written for OMAS agents to a message written for JADE. 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; however, messages exchanged inside a platform respect the object format of this platform (ACL messages for JADE). The OMAS transfer agent is located on the same machine as the JADE agents.

The exchange format follows the JSON grammar and syntax. A message is considered as a JSON object whose properties are in accordance with FIPA 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" : ["p4-3-2-1"]}} 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.

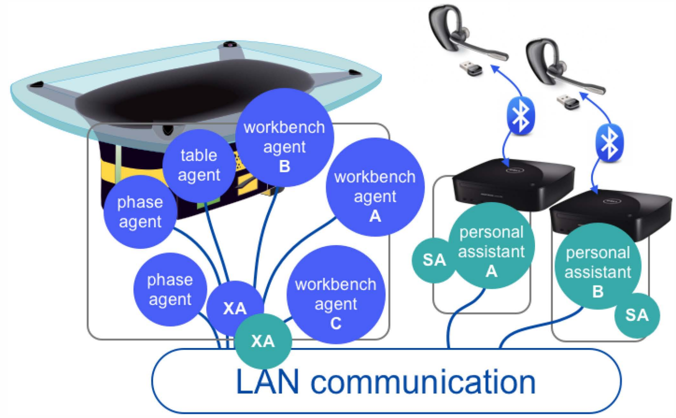

Fig. 4. OMAS Personal assistant agents have dedicated staff agent located one their machine. An OMAS exchange agent is located on the computer of the interactive tabletop to transfer message between the OMAS and JADE systems. (JADE agents are in blue; OMAS agents are in green.)

We will continue the example from above, where the user spoke the command "Create a new Post-it period." After the sub-dialogue the PA has all the information required to make a Post-it. He sends an internal OMAS message to a staff agent especially dedicated to managing Post-its. The staff agent uses the information he received from the personal agent concerning the content of the Post-it to send a message to the WbA on the table. Within the message sent by the staff agent to the workbench agent, he specifies that he would like the ID of the Post-it in the reply (i.e. the message is a FIPA "request" and not a FIPA "inform). The staff agent then sends the message to the WbA. When the message is broadcast to all of the OMAS agents on the LAN, it is intercepted by the transfer agent when he recognizes that the name of the receiver isn't an OMAS agent but a JADE agent. He encodes the message in a JSON format, and routes it toward an open socket, where a JADE transfer agent is listening for incoming packets. The JADE transfer agent receives the message, decodes the JSON format and sends an internal JADE message to the user's workbench. The user's workbench creates the post-it on the tabletop interface and then replies back to the OMAS Post-it staff agent (via the transfer agents) with the ID of the post-it that was created. Finally, the staff agent transfer this back to the PA, who communications the action was successful to the user.

C. Discussion

The architecture described in the previous sections has some advantages. Each multi-agent platform has its own role. The JADE system helps to manage the activities on the table and the board. The OMAS system manages the activities which directly assist the participants. Both multi-agent systems may exploit the information models which are developed on the tabletop. Internally, they use their own structure and dispatch their own messages.

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 tabletop, 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 synthesis 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, such as in the case of the meeting assistant CALO [19].

VI. Conclusion

The TATIN-PIC platform is a multi-display groupware for preliminary design. While the majority of the collaboration performed by the participants happens on our multi-touch tabletop, we have integrated an interactive board to support both convergent thinking and divergent thinking. Personal assistant agents to provide a natural user interface for our system, allowing user to perform multimodal tasks. We have described the architecture for our interactive TATIN-PIC platform and a middleware which serves as the cohesion between different heterogenous hardware devices and software information models. We have shown how personal assistant agents utilize the multi-agent framework to perform a rich set of tasks.

Though our project is focused on collocated CSCW, we have seen prior research in distributed CSCW with interactive surfaces, such as with the Tele-board [20]. One of the advantages of using a multi-agent system is that this middleware could apply to distribute infrastructures as well, with relatively little overhead for the programmers. There are several avenues for future work, including adding communication between different user's OMAS personal assistant agents to help users exchange information between themselves, and adding intelligent service agents to the personal assistant agents that can perform reasoning to help contribute more actively to the design process. We are also looking to add agents to the tablets and smartphones to provide more ways to interact with our system. User evaluations will be essential to uncovering new interaction techniques with in our collaborative environment.

Acknowledgment

Our work is supported and funded by the region of Picardy and the European Union. Europe in engaged in the region Picardy through FEDER.

References

[1] A. Jones, A. Kendira, D. Lenne, T. Gidel, and C. Moulin, "The tatin-pic project: A multi-modal collaborative work environment for preliminary design," in *Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD'11)*, June 2011, pp. 154–161.

[2] Y. Rogers and S. E. Lindley, "Collaborating around vertical and horizontal large interactive displays: which way is best?" *Interacting with Computers*, vol. 16, no. 6, pp. 1133–1152, 2004.

[3] O. Hilliges, L. Terrenghi, S. Boring, D. Kim, H. Richter, and A. Butz, "Designing for collaborative creative problem solving," in *Proceedings of the 6th ACM SIGCHI conference on Creativity cognition CC 07*, 2007, p. 137.

[4] F. Geyer, U. Pfeil, A. Hochtl, J. Budzinski, and H. Reiterer, "Designing reality-based interfaces for creative group work," in *Proceedings of the 8th ACM conference on Creativity and Cognition*, ser. CC '11. New York, NY, USA: ACM, 2011, pp. 165–174.

[5] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-defined gestures for surface computing," in *Proceedings of the 27th international conference on Human factors in computing systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 1083–1092.

[6] B. Hartmann, M. R. Morris, and A. Cassanego, "Reducing clutter on tabletop groupware systems with tangible drawers," in *Adjunct Proceedings of UbiComp 2006*, 2006.

[7] D. Schnelle-Walka and S. Döwling, "Speech augmented multitouch interaction patterns," in *Proceedings of the European Conference on Pattern Languages of Programs*, 2011.

[8] E. Tse, S. Greenberg, and C. Shen, "Exploring interaction with multi user speech and whole handed gestures on a digital table," in *Proceedings of the 8th international conference on Multimodal interfaces - ICMI '06*, Banff, Alberta, Canada, 2006, p. 76.

[9] S. Ko, K. Kim, T. Kulkarni, and N. Elmqvist, "Applying mobile device soft keyboards to collaborative multitouch tabletop displays: design and evaluation," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '11. New York, NY, USA: ACM, 2011, pp. 130–139.

[10] L. Terrenghi, A. Quigley, and A. Dix, "A taxonomy for and analysis of multi-person-display ecosystems," *Personal Ubiquitous Computing*, vol. 13, no. 8, pp. 583–598, Nov 2009.

[11] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, 1991.

[12] J. Schöning, P. Brandl, F. Daiber, F. Echtler, O. Hilliges, J. Hook, M. Löchtefeld, N. Motamedi, L. Muller, and P. Olivier, "Multi-touch surfaces: A technical guide," *Technical University of Munich*, 2008.

[13] U. Laufs, C. Ruff, and J. Zibuschka, "MT4j - a cross-platform multi-touch development framework," *ACM EICS 2010, Workshop: Engineering patterns for multi-touch interfaces*, 2010.

[14] N. Spanoudakis and P. Moraitis, "An ambient intelligence application integrating agent and Service-Oriented technologies," in *Research and Development in Intelligent Systems XXIV*, M. Bramer, F. Coenen, and M. Petridis, Eds. Springer London, 2008, pp. 393–398.

[15] J. P. Barthes, "OMAS - a flexible multi-agent environment for CSCWD," in *Computer Supported Cooperative Work in Design, 2009. CSCWD 2009. 13th International Conference on*, Apr. 2009, pp. 258–263.

[16] Y. Shoham, "Agent-oriented programming," *Artificial Intelligence*, vol. 60, no. 1, pp. 51–92, Mar. 1993.

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

[18] Negroponte, *Being Digital*. Vintage Publishing, 1995.

[19] G. Tur, A. Stolcke, L. Voss, S. Peters, D. Hakkani-Tur, J. Dowding, B. Favre, R. Fernandez, M. Frampton, M. Frandsen, C. Frederickson, M. Graciarena, D. Kintzing, K. Leveque, S. Mason, J. Niekrasz, M. Purver, K. Riedhammer, E. Shriberg, J. Tien, D. Vergyri, and F. Yang, "The CALO meeting assistant system," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1601 –1611, Aug. 2010.

[20] R. Gumienny, L. Gericke, M. Quasthoff, C. Willems, and C. Meinel, "Tele-board : Enabling efficient collaboration in digital design spaces," in *Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD'11)*, June 2011, pp. 47–54.