ADAPTATIVE MULTIMODAL ARCHITECTURES MANAGING SOFTWARE QUALITIES

Hicham Djenidi[1], Amar Ramdane-Cherif[1,2] and Nicole Lévy[1]
[1]*PRISM,* [2]*LISVI, Université de Versailles Saint-Quentin*
45, Avenue des États-Unis, 78035 Versailles Cedex, France

Keywords: Multimodal multi-agent architecture, Scenario of reconfiguration, Petri net models, Software quality.

Abstract: Multimodal interfaces for natural human-computer interaction involve complex architectures that should facilitate the process of matching IT to people. These architectures should react to events occurring simultaneously, and possibly redundantly, from different input media. In this paper, intelligent expert agent-based architecture for multimedia multimodal dialog protocols are proposed. The generic components of the multimodal architecture are monitored by an expert agent, which can perform dynamic changes in reconfiguration, adaptation and evolution at the architectural level. Software performance and usability are maintained by the expert agent via a scenario-based methodology. The expert agent's behavior modeled by Petri nets permits a software quality tradeoff between attributes of usability and other software attributes like system's performance.

1 INTRODUCTION

The use of multi-agent architecture (A), such as the Open Agent A (Martin et al., 1998), in which dedicated agents (Ag) communicate with each other is a common practice in multimodal (M) systems. However, the A in (Martin et al, 1998) is dedicated to predefined types of modality and does not constitute a methodology of formal and generic specification but offers rather the possibility of re-use in certain cases the already coded blocks and methods to connect them. Another example of MA, is the A of (Nigay and Coutaz, 1995) which supports the mechanism of generic fusion named melting pot. The core component in this A is the Dialog Controller - a set of cooperating Ag who realizes parallel data processing at several levels of abstraction. However, as showed by the author in (Bellik, 1995), this approach does not define clearly the roles of the Ag. Besides, its specification by does not allow an automatic validation of the scenarios of dialog before the development and the implementation phases of the prototype, as would allow it a timed colored Petri net (TCPN) model. QuickSet is a group of tools based on a multi-Ag A where co-workers act in a wireless M environment allowing several users to realize and to check military simulations by the use of voice, gesture and by attributing to users, behaviors and roles. This application, offers a centralized A (based on the blackboard concept) of colleagues Ag (Cohen et al. 1997). However, being a military application, the authors don't give enough information about their MA and about the way the dynamic dialog is specified in this MA. We notice, according to these examples, that, for MA, the literature offers, either architectural solutions dedicated to precise cases, or generic multi-Ag A, but at high levels of abstraction or without making a relevant identification of the Ag, detailed description of their roles and their dynamic interactions which they maintain. Or still, if they describe in a detailed way the M interaction, their A does not state the 'dynamic architectural reconfiguration' (in the case of the use or the renunciation of a modality for example.) Finally, for some of them, there is no formal specification of all the important aspects for the generic M management of the events. We think, that it is recommended to use a formal specification of the M interaction by a formal model, if we wish to allow the A of a M dialog, to take into account at the same time temporal and grammatical constraints and if we wish to prove formally the behavioral properties of the interaction before the stage of coding. We think, also, that an A has to allow the automatic validation of scenarios of dialog, before the stage of coding, to

Djenidi H., Ramdane-Cherif A. and Lévy N. (2009).
ADAPTATIVE MULTIMODAL ARCHITECTURES MANAGING SOFTWARE QUALITIES.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 349-352
DOI: 10.5220/0001665303490352

minimize development costs. On the other hand, it seems important that a MA offers characteristics relative to the dynamic reconfiguration (dependent on the semantic and\or temporal constraints of two or several events) to allow the most natural possible language. So, we choose to model a MA with a timed colored Petri nets TCPN because of its suitability to MA (Djenidi et al., 2004), (Navare et al., 2005) and dynamic concurrent objects (Robert, 2006).

Our paper presents an architectural model of an expert Ag (EAg) that monitors system and user performance in a MA and can dynamically adjust input and output modalities. The dynamic adaptation provides an interesting mechanism for enhancing the services given by the system based scenarios. The core of our proposal consists of: i) M software based on multi-queued components A modeled by a (TCPN). ii) An expert agent (EAg) to improve some software performance and usability (section 2), also modeled by a TCPN. The M software application is used like an example of the A on which we implement our EAg to increase the M software qualities. We develop some TCPN scenarios to show how the monitoring works and how we identify and improve the usability and performance qualities.

2 DYNAMIC RECONFIGURATION

2.1 Reconfiguration Scenario

Several modalities can simultaneously be used and this synergic use of input modalities could drive up errors. Different modalities introduced in our MA simulate the mouse and the keyboard events. For example, the haptic screen and the eye gaze system perform the mouse functionalities. To avoid these devices' inconsistency we impose an input modalities' activation rule. For this purpose, the input modalities are prearranged in three groups matching the use of different devices: i) Group 1: mouse, haptic screen, eye gaze system, wireless control mouse; ii) Group 2: keyboard, virtual keyboard; iii) Group 3: Speech recognition. The chosen rule is simple: the user can't activate two input modalities belonging to the same group. This rule is included into a scenario. The stimulus of the scenario is the activation of the new input modality. If a new input modality is activated, reactive agent's layer gets the event and sends it to its reasoning layer through the linking layer (Figure 1).

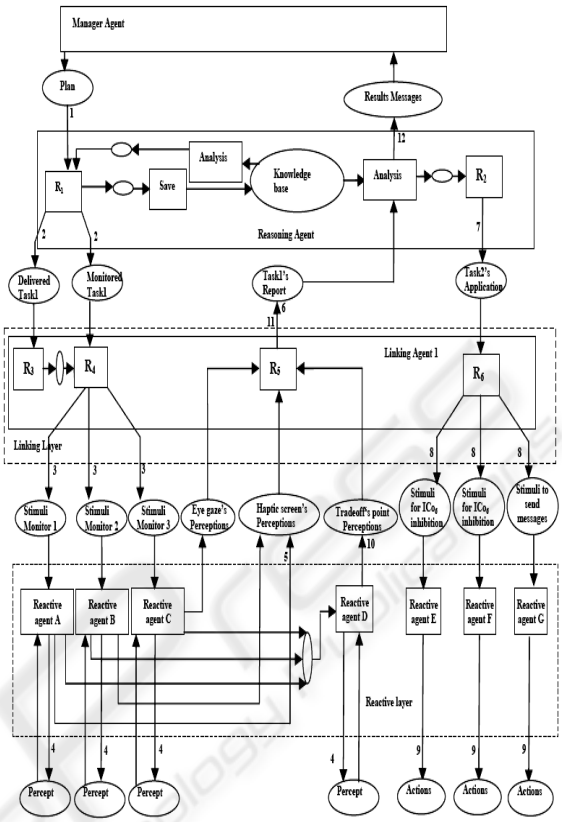

Figure 1: Principle of usability scenario application process: an example inside the expert agent.

The EAg tests if the modality is used at the same time with another device belonging at the same group (application of the rule). If the new modality causes conflict, the reasoning layer establishes a plan based on the scenario. The plan consists on deactivation of the new modality. The reactive layer deactivates the concerned component and its connection with the MA. Then, the input modality becomes inoperative. A sequence of the scenario process performed by the expert agent is symbolically described in Figure 1. When the manager agent receives the scenario, it analyses it and sends it as a plan to the reasoning agent (step 1). The plan only contains applicative requests, which are saved into the shared "Knowledge base" of the reasoning layer. The tasks containing the monitoring processes are sent to the adequate linking agent ("linking Agent 1" in Figure 1), in order to listen to the stimulus event and monitor the specified component (step 2). The actions of monitoring are distributed to the specific reactive agents ("Reactive agents A, B and C) (step 3). After that, the monitoring actions are applied in step 4. The feedback information about the perception actions is

sent to the "Linking Agent 1" (step 5). For example, the reactive agent C monitors an eye gaze component. The reports about the application of the task are sent to the reasoning agent in order to be analyzed (step 6). The reasoning agent contains the rule on the modalities. If the condition, on the modalities, is not satisfied, the stimulus condition is activated and the application of the role is started, by consequence, the new task "task 2" is sent to the linking agent (step 7). The linking agent applies the "task 2": primitives actions are sent to the reactive agent (step 8). The actions are accomplished by the reactive agents (Agents E, F and G) (step 9). The feedback of the last actions is accomplished by the reactive agent D (step 10). The information is charged to reports and sent up to the reasoning agent through the linking agent (step 11). Finally, the sum of the reports is sent to the manager agent in order to be transformed to documentations. Safety of the MA and priority of the others scenarios are continually checked during the gradual process of reconfiguration.

2.2 Example of Quality Tradeoff

In real situations, we don't know in which hardware the M application will be installed (the CPU performance, memory performance, etc. are unknown). The EAg must maintain the software performances regarding to user's impairment and hardware's performances of the computer, for example. In this scenario, the EAg employs the modalities' activations to drive up the application in a resource-less way. Clearly, certain modalities consume more resources than others. The scenario 1 specifies to the EAg to choose the modalities that consume less resources in order to maintain a high degree of performance in the M application. Another scenario (named scenario 2) identifies a major goal of one characteristic of usability that the expert agent can maintain. But in certain cases one characteristic is not sufficient to assume acceptable work of the application. Priority of the scenarios is then used and parameterized by the users (also a default parameterization is available in the EAg). In our example case, the Scenario 2 has priority 8/10, whereas the scenario 1 (performance) has priority 10/10. The comparison between the two qualities is occurred in a section "quality trade-off" defined in each scenario. Automatically, the EAg executes the scenario 2 at detriment of the scenario 1 (if the condition of the trade-off quality is satisfied). In the scenario 1, the expert agent monitors the consumed resources, if the CPU charges exceed the 60% the scenario 1 is applied and scenario 2 is stopped. For example, if we use the computer including Pentium III with 850 MHz CPU frequency, and if the user some modalities, the scenario 1 is executed (because the consumed resources are 70%). The Scenario 1 starts procedures to activate the modality belonging to the same group of the eye gaze system like haptic screen to satisfy the requirement for the performance of the application. In this case, the M application consumes 30% of resources. The EAg was modeled in CPN-tools for the two scenarios. The results of the simulation of the nets are presented at Figure 2. A reasoning agent 2 (like the one in Figure 1 but for scenario 1) randomly inhibits one of the three modalities involved in the process described by the scenario 1, when the CPU charge exceeds 100%, with respect to the constraints of the M interaction. The network is conceived for always guaranteeing that the CPU resources are not saturated when the M events occur (in parallel or sequential) and doesn't disturb the running interaction in the MA. It thus fulfils its function of monitoring and control in an intelligent and automatic way. The intelligence comes from the fact that we know that the agent will carry out its role, even if we are not able to affirm, in advance, at every moment, in a procedural way, which of the modalities will be inhibited by the agent. The automatism comes from the behavioral cyclic invariants in the Petri nets which allow to control and to monitor the MA. While the scenario 1 is performed, the EAg supervises and controls the input modalities and applies the scenario 2. The part of the EAg, in charge of the application of the scenario 2, uses the beginning instant of the events captured on the input modalities and their respective durations for performing activation and/or inhibition rules on the antagonist modalities. The A of the network is an automaton which applies the scenario 2 with respect to cyclic invariants. When inhibition actions are applied by the EAg to modalities, the cognitive agent sends messages to the user. During the step by step simulation of the networks with CPN-Tools (Jensen et al. 2008), we observe the correct behavior of the EAg performing the scenario 1 and the scenario 2. The graphs on Figure 2 show, during time, how the EAg reacts to the variation of the input signals. Those signals are the input modalities' ones. The inhibitor signals in Figure 2 are generated by the EAg, with respect to the MA's reliability, to apply the scenarios. CPN-ML inscriptions on the various arcs, in the codes (performed by the transitions), of conditions (checked by the transitions) and also the dynamic links (arcs) between the various instances of agents

(transitions) constitute the specifications of the actions of reasoning of the EAg CPN model according to its own rules which it works out. We made with the CPN-tools the automatic analysis of the behavioral properties of the nets: these networks do not comprise any blocking during the execution of the scenarios. This analysis (via CPN-Tools) gives a report of the behavioral properties checked. The report shows liveness properties of the networks which models the EAg.

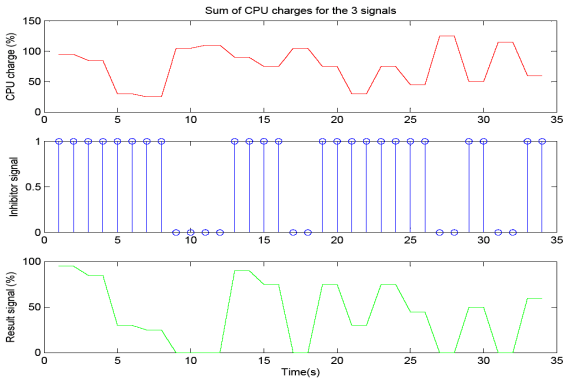

Figure 2: Result of control by reasoning agent CPU charges for 3 modalities during time.

3 CONCLUSIONS

The design of the multi-Ag systems offers a systematic methodology to produce formal specification of multi-agent multimodal architectural model. Modeling by TCPN takes all its importance here because it offers a formal framework whose advantages remain identical to those already quoted for the architectural paradigms of the M interaction The lack of commercial tools to specify and simulate multi-agent multimodal architecture before the stage of coding (in development life cycle's prototype) is a problem which we solve by our Petri net model approach with CPN-Tools. We described a model of the organization of the EAg and the problems of software qualities of the MA via an example. The model of the EAg in TPCN constitutes a strategic representation because it takes account of the various elements of the architectural structure of the EAg (various agents, various requirements, various data and their types, etc) and its behavior for the execution of the selected quality profile. Lastly, this model enables us to simulate the structure of the EAg for a profile of quality chosen before the phase of coding of the application, in addition to being able to make an automatic checking of the behavioral properties of the EAg grafted on MA, before this same phase of development. It also puts in value the various cyclic behaviors and the protocols of communications in the model. The Petri integration of the EAg provides important information on the course of the scenarios.
The innovation with our approach resides in explicit associations between quality requirements, automated scenarios and their behavioral validations in a TCPN model of a multi-Ag A.

REFERENCES

Martin, J. C., Julia, L., Cheyer, A., 1998, 'A Theoretical Framework for Multimodal User Studies', *International Conference on Cooperative Multimodal Communication, Theory and Applications'*, Tilburg, The Netherlands.

Nigay, L., Coutaz, J., 1995, 'A generic platform for addressing the multimodal challenge', *International Conference on Computer-Human Interaction*, ACM, pp. 98-105, Denver (CO).

Bellik, Y., 1995, *Interfaces multimodales : concepts modèles architectures*, PhD thesis. Paris, Paris XI.

Jensen, K., Kristensen, L., M, Wells, L., 2008, CPN Tools University of Aarhus, Denmark. Available from: <http://wiki.daimi.au.dk/cpntools-help/_home.wiki>
6 october 2008].

Cohen, P., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., Clow, J., 1997, 'QuickSet: Multimodal interaction for distributed applications', *Fifth ACM International Multimedia Conference MULTIMEDIA'97*, pp. 31-40.

Djenidi, H., Benarif, S., Ramdane-Cherif, A., Tadj, C., Levy, N., 2004, 'Generic Multimedia Multimodal Agents Paradigms and their Dynamic Based Agent Reconfiguration at the Architectural Level', *European journal of Applied Signal Processing*, pp. 1688-1707.

Navarre, D., Palanque, P., Schyn, A., Nedel, L., Freitas, C., Winckler, M 2005, '*A Formal Description of Multimodal Interaction Techniques for Immersive Virtual Reality Applications*', Interact 2005, IFIP. Rome, Italy, pp. 12-16.

Robert, G., Pettit, I., V., Gomaa, H., 2006, 'Modeling Behavioral Patterns of Concurrent Objects Using Petri Nets', *Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, pp. 303-312.