Adaptive Multimodal Architectures

A. RAMDANE-CHERIF(1), H. DJENIDI(1)(2), C. TADJ(2)
(1)PRISM, Université de Versailles Saint-Quentin,
45, Avenue des États-Unis, 78035 Versailles Cedex, France
{rca}@prism.uvsq.fr
(2)Electrical Engineering Department, École de Technologie Supérieure,
1100, Notre-Dame Ouest, H3C 1K3, Montréal, Québec, Canada.

Abstract

In this paper, intelligent expert agent-based architectures for multimedia multimodal dialog protocols are proposed. The full architecture is implemented to help disabled people to access Web services and common computer equipment. The generic components of the application are then monitored by an expert agent, which can then perform dynamic changes in reconfiguration, adaptation and evolution at the architectural level. The expert agent's behavior modeled by Petri nets permits a maintainability of software quality via a scenario-based methodology

1. Introduction

Information Technology (IT) can offer much to people with disabilities who may also be socially isolated and equipment deprived [1, 2, 3]. However, the results of research highlight difficulties faced by the technology when attempting to offer to these people the use of standard technology (for using a computer or accessing Internet for examples). In this context, multimodal application projects aim to facilitate the process of matching disabled people to appropriate technology and methods for accessing common computer equipment [4, 5, 6, 7]. They should professionally assess and document a customer's physical skills and success in using the computer, like using task specific assessment appropriate to the Web, rather than traditional "desk and paper-based" perceptual, language, memory and physical assessment, for example.

This paper describes a multimodal multimedia (MM) software environment where speech, eye-gaze, wireless control, virtual keyboard and/or tactual screen are used as input modalities and where, display on monitor screen and speech synthesis are the output ones. This application is dedicated to paralytics and allows them to navigate in the Web and to use the Windows environment. We develop some HLTCPN scenarios to show how the monitoring works and how we identify and improve the usability quality

2. A generic multi-agent MM architecture

In the architecture depicted in Figure 1 each input modality must be associated with a language agent (LA). The LA embeds an interpreter component (ICo) as shown in Figure 2. For basic modalities like manual pointing or mouse-clicking, the complexity of the LA is sharply reduced. It is the ICo that checks whether or not the fragment is known or not and or no longer necessary. The LA embeds a 'Sentence Generation Component' which is also reduced to a simple event thread whereon another external control agent could possibly make parallel fusions. In such a case, the external agent could handle 'Redundancy' and 'Time' information, with two corresponding components. These two components are agents which check redundancies and the time neighborhood of the fragments respectively during their sequential regrouping. A 'Serialization Component' processes this regrouping. Thus, depending on the input modality type, the LA could be assimilated into an expert system or into a simple thread component. Two or more LAs can communicate directly for early parallel fusions [8] (Figure 1 left) or, through another central agent, for late ones [9] as shown in Figure 1 right. This central agent is called a Parallel Control Agent.

In the first case, the 'Grammar Component' of one of the LAs must carry extra-semantic knowledge for the purpose of parallel/serial fusion/fission made by the semantic component (SCo). This knowledge could also be distributed between LA's 'Grammar Components'. Several Serializing Components share their common information until one of them gives the sequential parallel fusion output. In the other case, a 'Parallel Control Agent' (PCA) handles and centralizes the parallel fusions of different LA information. For this purpose, the PCA has two intelligent components, for redundancy and time management respectively. These agents exchange information with other components to make the decision. Then, generated authorizations are sent to a component that performs semantic parallel/serial fusion/fission. Based on these agreements, this component will carry out the steps of the semantic fusion/fission process. The Redundancy and Time Management components receive the redundancy and time information via the Semantic Fusion Component or directly from the LA, depending on the complexity of the architecture and on designer choices. When an architectural choice is done, the model can be successively refined (with the same tool of specification: HLTCPN) in a top-down way until the fusion/fission dialog level [8] or lower level.

3. InterAct Software

To support the novel aspect of the approach, this section describes the three main characteristics of the proposed architecture through a multimodal interface software application called InterAct Software 1.0 (IAS). IAS is dedicated to use by disabled individuals (particularly those who are paralyzed, like hemiplegics, quadriplegics, etc.), and is based on late fusion architecture modeled with a design CPN, as shown in the previous sections.

IAS assists the disabled person during the interaction with the computer. It plays the role of intelligent intermediate between the user and the computer. It offers to the disabled user certain applications like Internet navigator and permits him/her to use all Windows applications with great freedom. IAS also offers to the disabled users the possibility to control the computer in Windows environment via different modalities. During the communication with the computer, IAS can suggest autonomously to the user some modalities by taking into account his impairments [6].

IAS is based on components using multithreading technology. The disabled users can use several modalities to interact with the computer (see the HLTCPN-modeled dialog architecture on Figure 2). We implement a mouse emulator (wireless control mouse), a virtual keyboard (keyboard showing on the screen), an eye gaze System (permits to the user to move mouse with its eyes), a haptic screen, a speech recognition system and a classical mouse and keyboard. In Figure 2 the Interpreter Components (ICos) are special components designed to translate the signals sent by the input devices. A first instance of the 'Semantic Component' (SCo_1) filters messages received from the Interpreter components.

The intervals allowed for fusion, defusing and fission are chosen in the HLTCPN-model by the architect in agreement with the nature of the processed data and managed by the 'Time Management Component' (TMCo) via temporal windows. This component verifies if the time constraints are respected at each step of the multimodal process. The TMCo resets or allows the SCo2 (the second instance of the Semantic Component) to generate gradually a semantic multimodal sentence if the time constraints are respected. This is done following a multimodal grammar. The multimodal sentences are saved and therefore commands are merged to the output modalities via three components. The Visual, Audio and Commands components are linked to the output modalities and managed by the SCo2.

4. Improving quality attributes of MM architecture via an expert agent

IAS lets disabled users act via several modalities. But more than that, the developed software is reactive to the user's requirements and able to response autonomously to support demands. It embeds an implementation of an expert agent which supervises the MM architecture. This agent processes (Figure 3) in runtime mode: i) monitoring of the software qualities; ii) architectural reconfiguration to maintain desired quality tradeoff.

The development of MM application requires establishing high functionalities and quality attributes [9, 10, 11] to make it easier and more convivial to use for the disabled persons. We define the software quality as the degree to which software possesses a desired combination of attributes [12].

The introduction of the expert agent on the MM architecture provides many advantages:

- IAS responses in runtime mode to the users needs in term of required quality.
- The expert agent improves the modifiability of the MM architecture by dynamic reconfigurations in order to recover some qualities.
- The expert agent is able to maintain a tradeoff between required qualities in runtime mode by continuous monitoring on the architecture.
- The expert agent manages the input and output modalities and suggests to the disabled user the right modalities at the right time.
- The expert agent includes default scenarios (suggested by some interviewed disabled users and regarding to our specifications) or new scenarios obtained via completed forms (in runtime mode).
- The expert agent produces a MM report, supporting graphics, values, text and/or voice. The report classifies the quality attributes, contains information about the scenario presented (under priority constraints) and gives the risk points, the tradeoff points for this scenario's choice. This information is saved and can be checked any time by the user.

In the following we show one examples of scenarios targeting one characteristic of the attribute usability. To achieve the developed scenarios the architectural configuration must change at variable points already identified in the architecture. Several modalities can simultaneously be used in InterAct (Figure 4) and this synergic use of input modalities could drive up errors. Different modalities introduced in InterAct simulate the mouse and the keyboard events. For example, the haptic screen and Eye Gaze system perform the mouse functionalities. A dilemma occurs when the users attempt to use two devices simulating the same physical device. To avoid these devices' inconsistency we impose an input modalities' activation rule. For this purpose, the input modalities are prearranged in three groups:

- Group 1 (Mouse Simulator): mouse, haptic screen, Eye Gaze System, Wireless control mouse.
- Group 2 (Keyboard Simulator): keyboard, virtual keyboard.
- Group 3 (Vocal Commands): Speech recognition.

The chosen rule is simple: the user can't activate two input modalities belonging to the same group (for example, eye gaze system and the mouse). But one input modality can be used with other modalities belonging to the other groups (for example, the haptic screen and speech recognition).

The management of the input modalities is expressed by a rule and included into a scenario. The stimulus (Figures 5) of the scenario is the activation of the new input modality. The expert agent uses the scenario in order to manage the input modalities autonomously and avoid errors' generation during the disabled user's dialog with the system. If a new input modality is activated, its agent's reactive layer gets the event and sends it to its reasoning layer through the linking layer (Figure 5).

The expert agent 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 ICo and its connection with the SCo_1. Automatically, the input modality becomes inoperative.

A sequence of the scenario process performed by the expert agent is described in Figure 6. When the manager agent receives the scenario (Figure 6), 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 architecture knowledge base" of the reasoning layer.

The tasks containing the monitoring processes are sent to the adequate linking agent ("linking Agent 1" in Figure 6), 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 the ICo6. 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).These actions are: i) Action A (1, 2): deactivating the modality and ii) Action A (1, 3): sending message to the user. 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 changed 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. Note that the network in Figure8 doesn't describe the creation of agents. If the scenario implicates the creation of agents, the expert agent should engage a creation's process (not showed in Figure 6).

6. Conclusion

We proposed a generic MM architecture modeled with HLTPCN. The architecture is monitored and reconfigured by an expert agent via scenarios and with respect to software quality attributes. The scenarios are included or generated in the expert agent. These scenarios identify the variable points of the MM architecture and get different decisions on changes. Because of the HLTCPN modeling of the MM architecture and its dynamic reconfigurations scenarios, the MM dialog and interaction's proprieties were formally checked within time and stochastic embedded constraints. The expert agent (modeled and implemented for the software evaluation and reconfiguration) is able to: i) provide a report of evaluation and ii) autonomously maintain the qualities required by monitoring the architecture.

7. References

[1] Christophe Jacquet, Yolaine Bourda and Yacine Bellik, "A Context-Aware Locomotion Assistance Device for the Blind". People and Computers XVIII -- Design for Life, 2004. pp. 315-328. Sally Fincher, Panos Markopoulos, David Moore and Roy Ruddle, editors. Springer-Verlag (London).

[2] Marcela Fejtová, Jan Fejt and Lenka Lhotská: Controlling a PC by Eye Movements: The MEMREC Project. Computers Helping People with Special Needs, 9th International Conference, ICCHP 2004, Paris, France, July 7-9, 2004, Proceedings. Lecture Notes in Computer Science 3118 Springer 2004, p.770-773.

[3] Malek Adjouadi, Anaelis Sesin, Melvin Ayala, Mercedes Cabrerizo: Remote Eye Gaze Tracking System as a Computer Interface

for Persons with Severe Motor Disability. ICCHP 2004, Paris, France, July 7-9, 2004, Proceedings. Lecture Notes in Computer Science 3118 Springer 2004, p. 761-769.

[4] Mario Mühlehner, Klaus Miesenberger: HeadControl+: A Multi-modal Input Device. ICCHP 2004, Paris, France, July 7-9, 2004, Proceedings. Lecture Notes in Computer Science 3118 Springer 2004, p.774-781.

[5] Cyril Rousseau, Yacine Bellik, Frédéric Vernier, Didier Bazalgette, "A Framework for the Intelligent Multimodal Presentation of Information", Signal Processing journal, Special issue on Multimodal Interfaces, Elsevier, 2006.

[6] H. Djenidi, S. Benarif, A. Ramdane-Cherif, C. Tadj and N. Levy. InterAct Software 1.0: An Eye-Gaze-Speech Multimodal Environment Dedicated To Disabled People. In ISPACS 2003: International Symposium on Intelligent Signal Processing and Communication Systems. Awaji Island, Japan, December 7-10, 2003.

[7] H. Djenidi, S. Benarif, A. Ramdane-Cherif, C. Tadj and N. Levy. Generic Multimedia Multimodal Agents Paradigms and their Dynamic Based Agent Reconfiguration at the Architectural Level. In EURASIP: European journal of Applied Signal Processing, November 2004. Pages 1688-1707. DOI: 10.1155/S1110865704402212.

[8] H. Djenidi, A. Ramdane-Cherif, C. Tadj and N. Levy. Generic Multi-Agents Architecture for Multimedia Multimodal Software Environment, In JOT'2004: International Journal of Object Technology. Vol. 3, no. 8, September-October 2004, pp. 147-168.

[9] Bass, L.; Clements, P. & Kazman, R. Software Architecture in Practice. Reading, MA: Addison Wesley Longman, 1998.

[10] Bass, L; Klein, M. & Bachmann, F. Quality Attribute Design Primitives (CMU/SEI-2000-TN-2000-017). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.

[11] Barbacci, M. Klein, M. Longstaff, T & Weinstock, C. Quality attributes (CMU/SEI-95-TR-21 ADA307888). Pittsburg, PA: Software Engineering Institute, Carnegie Mellon University, 1995.

[12] Bass, L. Klein, M. Moreno, G. Applicability of general scenarios to the architecture tradeoff analysis method. October 2001. Technical report.

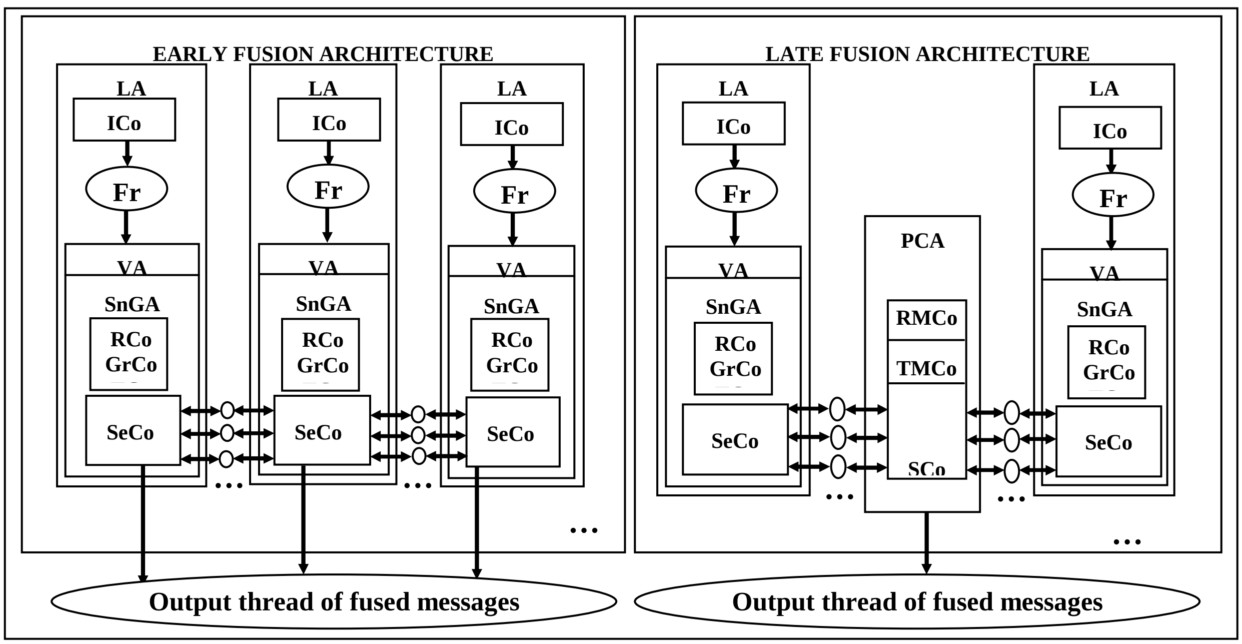

Fig. 1. Principles of early and late fusion architectures (A: agent, C: control, Co: component, Fr: fragments of signal, G: generation, Gr: grammar, I: interpreter, L: language, M: management, P: parallel, R: redundancy, S: semantic, Se: serialization, Sn: sentence, T: time and V: vocabulary). More connections (arrows that indicate the data flow) could be added or removed by the agents to gather fusion information

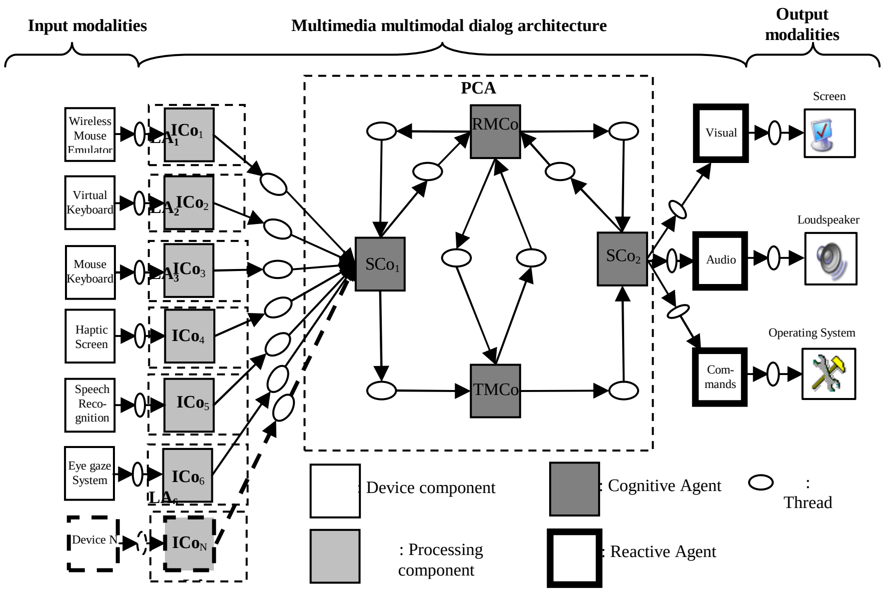

*Fig. 2. Multimodal multimedia dialog architecture (A: agent, C: control, Co: component, G: generation, Gr: grammar, L: language, M: management, P: parallel, R: redundancy, S: semantic, T: time and V: vocabulary)***.**

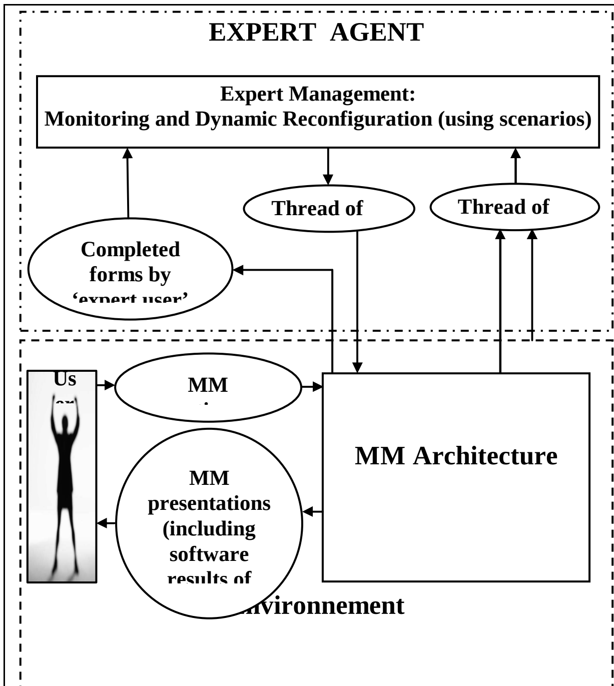

Fig. 3 Global view of monitoring and dynamic reconfiguration process

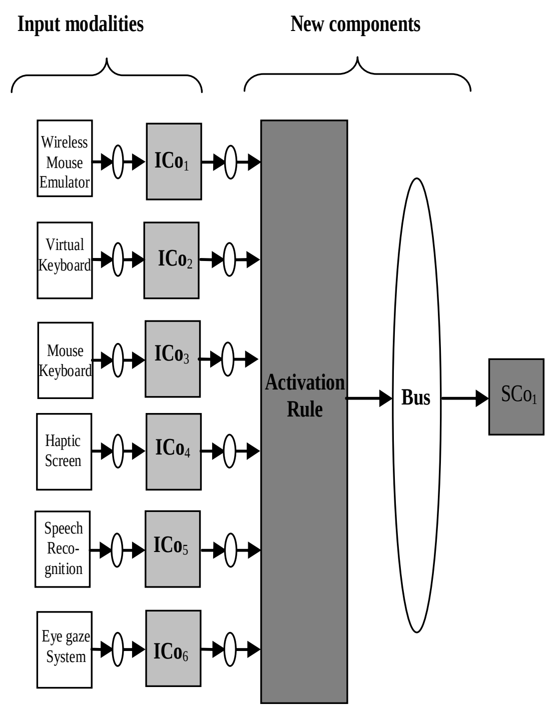

Fig. 4- Activation Rule in InterAct.

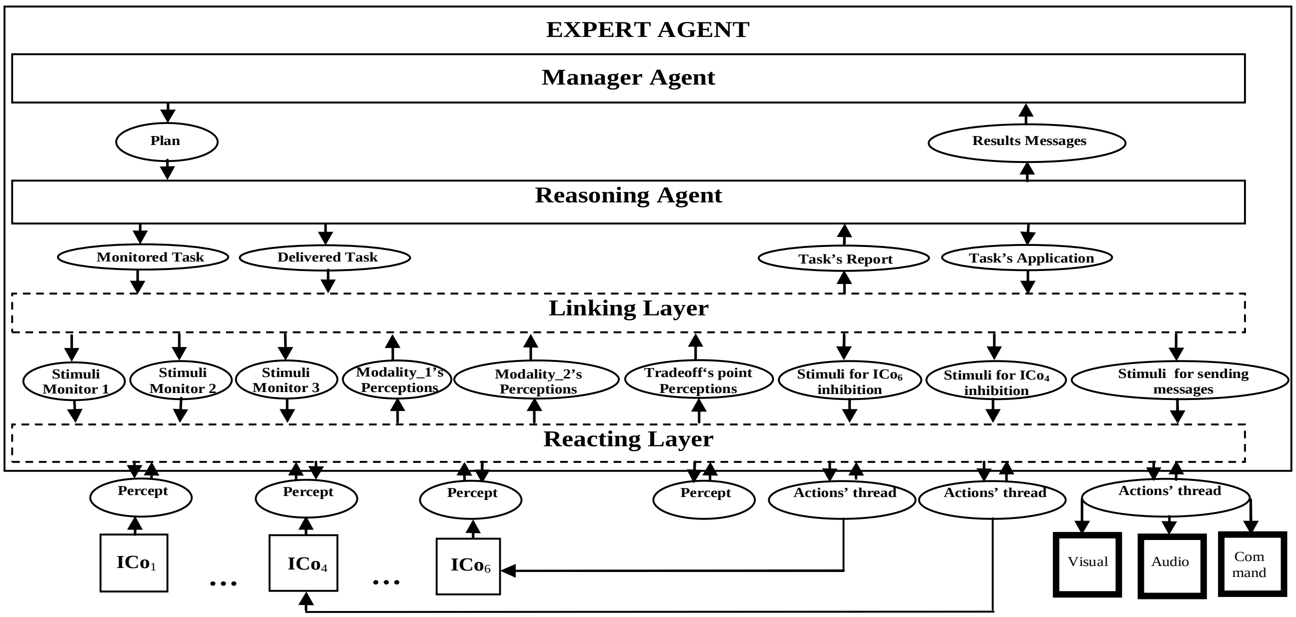

Fig.5. Expert Agent.

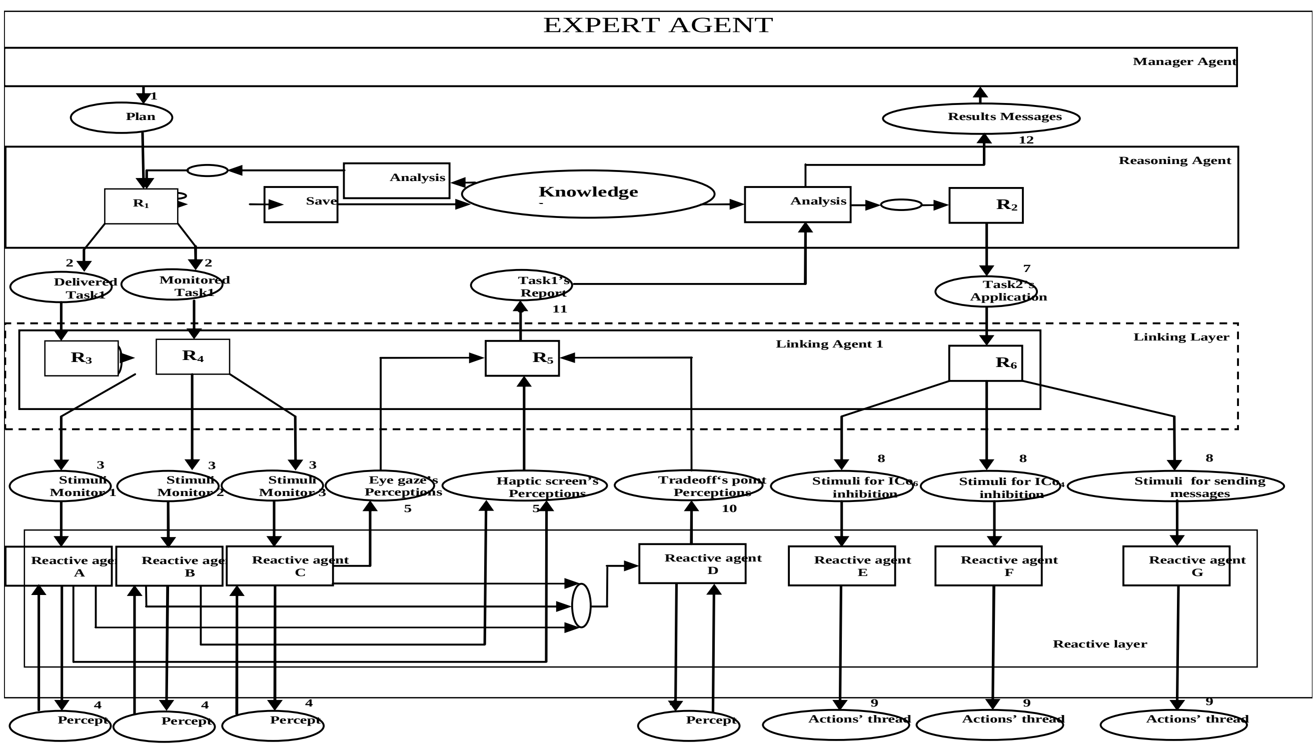

Fig. 6. Usability Scenario application process example inside the expert agent.