



PROTOTYPING USING A PATTERN TECHNIQUE AND A CONTEXT-BASED BAYESIAN NETWORK IN MULTIMODAL SYSTEMS

Atef Zaguia^{1,3}, Chakib Tadj¹, and Amar Ramdane-Cherif²

1 MMS Laboratory, Université du Québec, École de technologie supérieure,
Montreal, Canada.

Emails: ctadj@ele.etsmtl.ca

2 LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines,
Paris, France.

Emails: rca@prism.uvsq.fr

3 Department Of Computer Science, College of computers and information technology
Taif University, Taif, Saudi Arabia

Emails : Zaguia.atf@tu.edu.sa

Submitted: May 3, 2015

Accepted: Jul. 21, 2015

Published: Sep. 1, 2015

Abstract- Today, technology allows us to produce extensive multimodal systems which are totally under human control. These systems are equipped with multimodal interfaces, which enable more natural and more efficient interaction between man and machine. End users can take advantage of natural modalities (e.g. audio, eye gaze, speech, gestures, etc.) to communicate or exchange information with applications. In this work, we assume that a number of these modalities are available to the user. In this paper, we present a prototype of a multimodal architecture, and show how modality selection and fission algorithms are implemented in such a system. We use a pattern technique to divide a complex command into elementary subtasks and select suitable modalities for each of them. We integrate a context-based method using a Bayesian network to resolve ambiguous or uncertain situations.

Index terms: Multimodality, Ontology, Bayesian Network, Pattern, User Interface, Multimodal Fission.

I. INTRODUCTION

Computer systems are born of scientific needs, but they owe their success to general public use. This has motivated researchers to develop systems that meet the needs of users and to target use of these systems on a large scale. Current technological advances are leading to the design of ever more powerful machines which are increasingly easy to use. These machines should be capable of interacting with users in a harmonious way, but this is only possible if they are able to understand human communication in many of its natural modalities, such as speech, gestures, eye gaze, and facial expressions. Inspired by these communication modalities, multimodal systems are developed to combine a number of them, depending on the task at hand, on user preferences, and on the user's intentions.

These systems represent a remarkable deviation from conventional systems with their standard human-machine interface modalities, such as windows icons, for example, as they provide the user with more natural means of interaction, which are both flexible and portable.

A multimodal system has two crucial components: the fusion process and the fission process. In fusion [1] two distinct data modalities are combined; for example, data modalities, such as a mouse and speech [2]. In fission, by contrast, a complex command is divided into elementary subtasks, which are presented as elementary output modalities.

In our work here, we focus on: 1) the services connected to the output, that is, multimodal fission; and 2) the creation of a multimodal interaction system. The objective is to develop a flexible multimodal system, capable of manipulating more than two modalities that can interact with more than one application. The approach consists of designing modules that detect the modalities involved, take into account the parameters of each modality, and perform the fission of the modalities in order to obtain the corresponding action to be undertaken. The rest of this paper is organized as follows. Section II discusses work that is related to ours. Section III describes our research issue. Section IV presents the various components of the proposed architecture. Section V discusses modality selection and the fission algorithm. Section VI presents the implementation of the prototype. The paper is concluded in section VII.

II. RELATED WORK

In the context of human-machine and human-human communication, modality refers to the path or channel by which human and machine interact. Multimodality improves the recognition and comprehension of the commands emanating from the environment (user, robot, etc.) by the machines. A system that combines many modalities dynamically in both input and output is called a multimodal system. “Multimodal systems are generally intended to deliver natural and efficient interaction, but it turns out that there are several specific advantages of multimodality” [3]. The first multimodal system was created in 1980 by Richard Bolt, “Put-That-There” [4]. This system, equipped with a microphone and a touchscreen, made it possible to move or change the display of objects on the screen, and accommodate voice commands accompanied by pointing on a touchscreen.

Since Bolt published his work, academia has provided prototypes and systems that offer a variety of multimodal interaction techniques. These systems are a particularly effective solution for users who can't use a keyboard or a mouse, visually impaired users, users equipped with mobile devices, handicapped users, etc.

Most current multimodal systems address a very specific technical problem, such as media synchronization [5], or they are dedicated to very specific modalities [6], [7], [8] and [9]. The system presented in [2] is a multimodal system designed for blind users, which allows them to read through Braille mathematical formulas. This system combines a keyboard and voice for input and Braille and audio for the output. The system presented in [27] use Kinect based multimodal gesture recognition, exploiting multiple audio and visual modalities.

In [10], the authors present an efficient multimodal system (RISCOM) which can be used in the case of a disaster. The system sends information on a mobile device in the form of maps to indicate the location of safe havens in a natural disaster. It uses display, audio, and text message modalities, is easy to use, and provides instant access to the information. By integrating multiple modalities, this system is a very effective way to deliver emergency services in critical situations, which can help save lives.

The prototype presented in [11] is equipped with a speaker, a video-camera, a microphone, a touchscreen, a graphical user interface, and a talking head. It is a multimodal kiosk with a user

interface which accommodates touch, natural speech input, and head and hand gestures, and can also be used by those who are physically challenged [11].

PIXELTONE [12] is a multimodal photo editing system. The user speaks to edit images, instead of hunting through menus.

However, most research in multimodal systems is focusing more on the fusion process [13] than the fission process [14] [15] [16], in spite of the fact that the fission module is a critical component of multimodal systems. As these authors put it, “There isn’t much research done on the fission of output modalities, because most applications use few different output modalities, therefore simple and direct output mechanisms are often used” [13]. Also supporting our viewpoint are [17]: “Multimodal fission is a research topic that is not often addressed in the scientific community.”. In the few cases where the fission process is presented, the systems are very simple [18] and the user is limited to some predefined instructions.

In this paper, we propose a new methodological solution in which an architecture is modeled that facilitates the work of the fission process. We do this by defining an ontology that contains different applicable scenarios and describes the environment in which the multimodal system exists. We then implement this architecture in a real application.

III. CHALLENGES AND PROPOSED SOLUTION

Our objective is to develop an expert system capable of providing services to different multimodal applications. The main goal is to create a multimodal fission system capable of understanding a complex command from the environment, particularly from the user. In order to achieve this, a complete semantic modeling of the environment is required. The task of this system is to divide a complex command into elementary subtasks and present them to the output modalities.

To achieve this goal, we list the main challenges that need to be addressed in developing our system, along with our proposed solutions:

1. What are the modules required to design the architecture of a multimodal fission system?
Here, we will specify, define, and develop all the necessary components of the system ;
2. How will we represent the multimodal information? We will model the environment semantically, and create a context-sensitive architecture that is able to: 1) manage multiple distributed modules; and 2) automatically adapt to the dynamic changes of the interaction context (user, environment, system) ;
3. How will we perform the fission process? We will introduce an algorithm that describes the fission mechanism, including the rules of fission and the rules for selecting the output modalities ;
4. How will the system manage uncertain or ambiguous data in the fission process? We will introduce a new, context-based method using a Bayesian network (BN) to resolve the uncertainty problem during the fission process in a multimodal system ;
5. What is the optimal representation of the environment in our architecture? The optimal representation is a solution based on an ontology, which we will adopt. The modalities, scenarios, and objects, along with their characteristics, are stored in the ontology that describes the relationship between them.

We present these challenges summary form only in this paper. For more details, please refer to [19], [20], and [21].

IV. COMPONENTS OF MULTIMODAL FISSION SYSTEM

The objective of multimodal fission is to move from an independent presentation of modalities to a coordinated and coherent multimodal presentation. A general schema of a fission process is presented in Figure 1. The process consists of three main modules: Modalities Selection, Fission, and Subtasks-Modalities Association. These modules are explained briefly in the next sections.

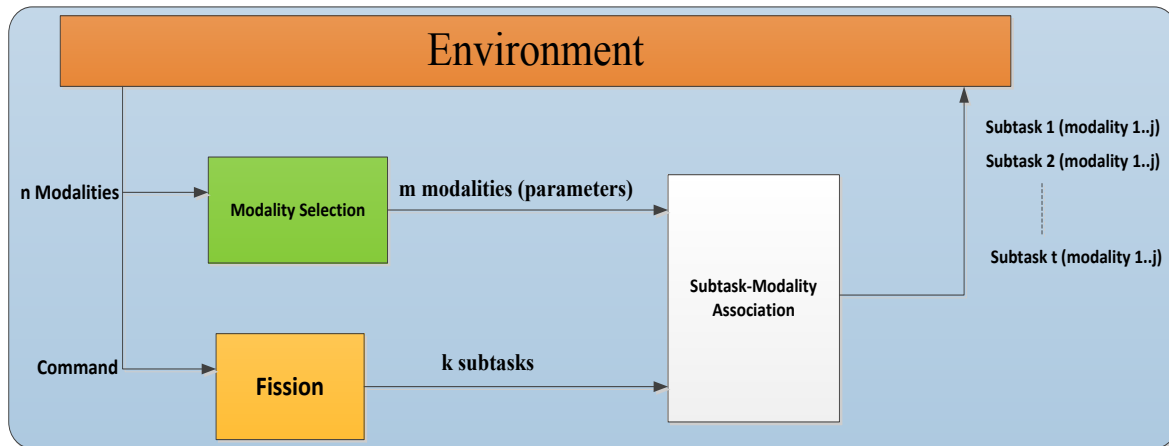


Figure 1. General view of the fission process.

a. MODALITIES SELECTION

This module plays an important role, as it is here that the appropriate modalities are selected according to the context. The context is defined by three essential components [19]:

User context: This component gives the user profile and its location. It makes it possible to determine the capability of the user to use certain modalities. For instance, if the user is blind, the display modality will be disabled.

Environmental context: This component obtains information from sensors installed in the user/machine environment. It detects changes in that environment and adjusts the selection of modalities based on these changes. For example, if the system detects that the environment is becoming too noisy, the audio modality is disabled.

System context: This component detects the computing device that the user is currently using, as well as the important parameters of the computing resource, such as the currently available bandwidth, the network to which the computer is connected, the computer's available memory, the specifications of the battery, and the type of processor and its activities.

b. FISSION

This module takes as input the complex command and as output the elementary subtasks. It is the crucial component in our architecture, with the role of dividing a complex command into elementary subtasks.

The fission process is based on the use of the pattern fission technique [20]. These patterns are generally defined as having two parts: problem and solution (Figure 2).

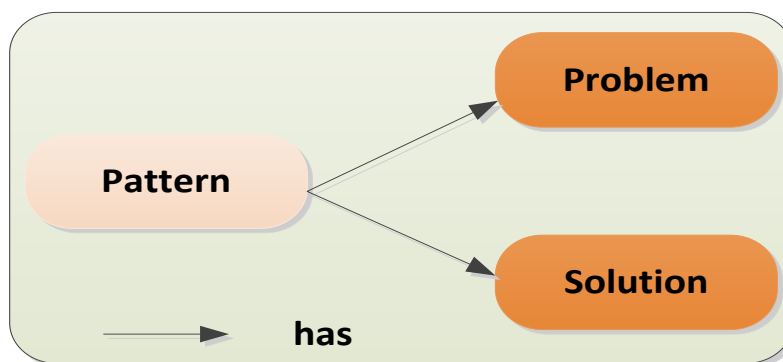


Fig. 2 Pattern definition.

These patterns are stored in an ontology. A simple example of a pattern is shown in Figure 3.

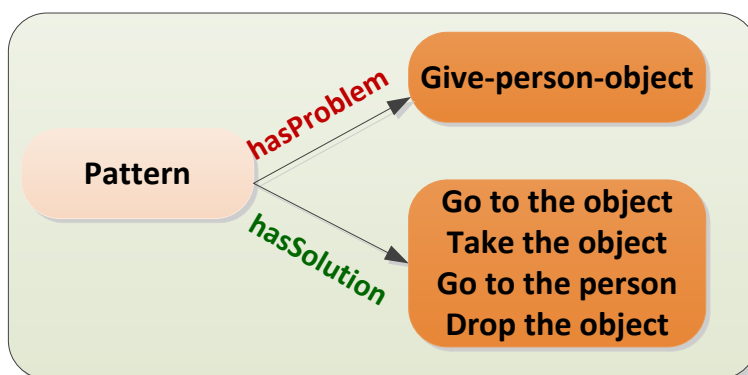


Fig. 3 Example of a pattern.

The system sends a query with the problem parameters to find the matching pattern fission in the ontology.

c. SUBTASKS-MODALITIES ASSOCIATION

This module takes as input the elementary subtasks and the available modalities, and as output a set of subtasks associated with the appropriate modality or modalities. The goal with this module is to associate each subtask generated by the fission module with the appropriate and available modality or modalities. We also use patterns in this part as predefined models that describe the selected modality or modalities. In our work, a modality pattern is defined by: a) a problem composed of the components application, parameter, priority, combination, scenario, and service; and b) a solution composed of the selected modality or modalities. For more details regarding scenario selection and modality selection, see [19].

d. BAYESIAN NETWORK

During the fission process, the system might face ambiguity or uncertainty. To overcome this problem, we use a context-based Bayesian network (BN). “A BN provides a mechanism for graphical representation of uncertain knowledge and for inferring high-level activities from the observed data. Specifically, a BN consists of nodes and arcs connected together forming a directed acyclic graph. Each node can be viewed as a domain variable that can take either a set of discrete values or a continuous value. An arc represents a probabilistic dependency between the parent node and the child node” [23].

We represent our adaptation of the BN with context information (time, user status, temperature, location, etc.) with the following equation:

$$Con_i \rightarrow (C_j, P_j), i = 1, \dots, n$$

where n and m represent the number of contexts and the number of ambiguous concepts respectively, Con = context, C= concept, and P = probability. The arrow represents the relation between context and concept. Each context is connected to one or more concepts with a corresponding probability. We choose the most probable concept by calculating the probabilities of each concept using the following equation:

$$P(C|Con) = \frac{P(Con|C)P(C)}{P(Con)}$$

$$Con = con_1, con_2 \dots, con_n$$

Where:

$P(C|Con)$: a posteriori probability;

$P(Con|C)$: likelihood;

$P(Con)$: evidence;

$P(C)$: a priori probability.

e. ONTOLOGY

We use an ontology [25] to model the environment. As shown in Figure 4, the environment is composed of the following classes [19]:

The Modality Class represents the possible modalities present in the environment. It contains the subclasses vocal, visual, gestural, tactile, and manual.

The Context Class represents the interaction context [26] containing the user context, environmental context, system context, and location.

The Event Class contains the four subclasses that can form a command, and their possible combinations:

- Action subclass: the verbs that the command can contain ;
- Location subclass: the locations that we can find in a command ;
- Object subclass: the various objects that we can use ;
- Person subclass: the relations that exist between individuals.

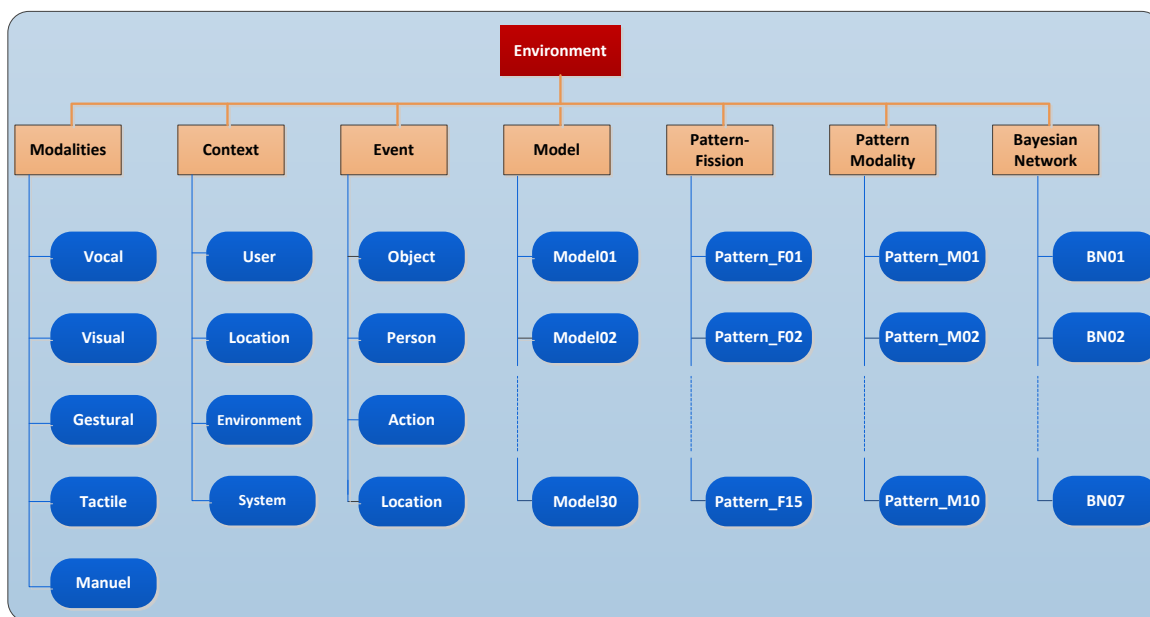


Figure 4. The classes of the ontology.

The Model Class contains 30 subclasses (Model01 to model30), and allows us to validate the meaning and the grammar of a command. We have defined several models, each of which includes two or more subclasses of the Event class in a predefined order. For instance, the model of the command, “Put the ball on the table,” is Action→Object→Location.

The Pattern Fission Class contains 15 subclasses (Pattern_F01 to Pattern_F15), and describes various scenarios that are saved as patterns. They are mainly composed of two parts: problem and solution, as described briefly in section 4.b and in detail in [20].

The Pattern Modality Class contains 10 subclasses: Pattern_M01 to Pattern_M10, and allows the selection of the appropriate modalities for a given subtask.

The Bayesian Network Class contains 7 subclasses (BN01 to BN07), and defines the possible BNs in the case of ambiguity or an uncertain situation. These subclasses are modeled in the ontology.

V. MODALITIES SELECTION AND FISSION ALGORITHMS

In this section, we present the algorithm for every module presented in Figure 1. Figure 5 shows modality selection algorithm. Once the environment has been modeled and a set of contexts defined, these contexts become the parameters that affect modality selection. There are four types of context: environment, user, location, and system. When an event is detected, the system receives information from sensors and looks for correspondence between the data received from the environment and those of the ontology. If contexts are verified, then the modality is enabled; otherwise it is disabled.

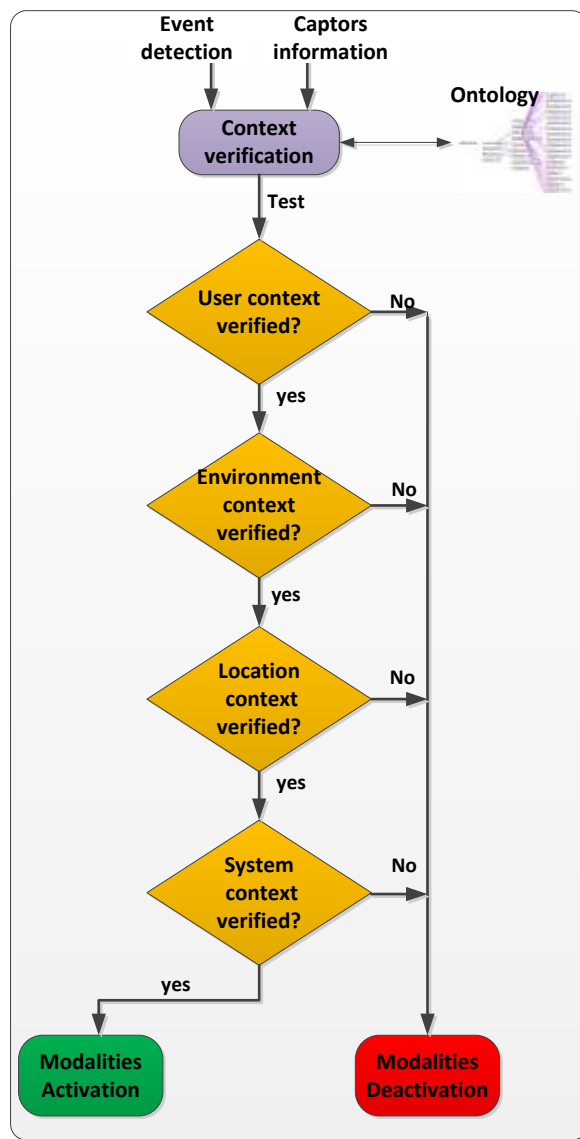


Figure 5. Modalities selection.

Figure 6 describes the steps involved in the fission process. In this diagram, a numbers of commands n serve as an input to the system. The steps undertaken are as follow:

Step-1: the system extracts every word from the command.

Step-2: for every word, the vocabulary stored in the Vocab ontology is checked.

Step-3: $vocab_i$ is extracted from each $word_i$. The extracted words are then concatenated in the same order as in the original command. In this way, the model of the command is obtained.

Step-4: a query is sent to the Grammar Model ontology to look for the model.

Step-5: if the model is found, we proceed with step 7 otherwise we proceed with step 6.

Step-6: the command is not valid and a feedback is sent to the user.

Step-7: a query is sent to find a matching pattern fission from predefined patterns stored in the Pattern Fission ontology. The system compares the query with all the pattern fission problems stored (in the ontology) until it finds a match. Then the pattern fission solution is returned.

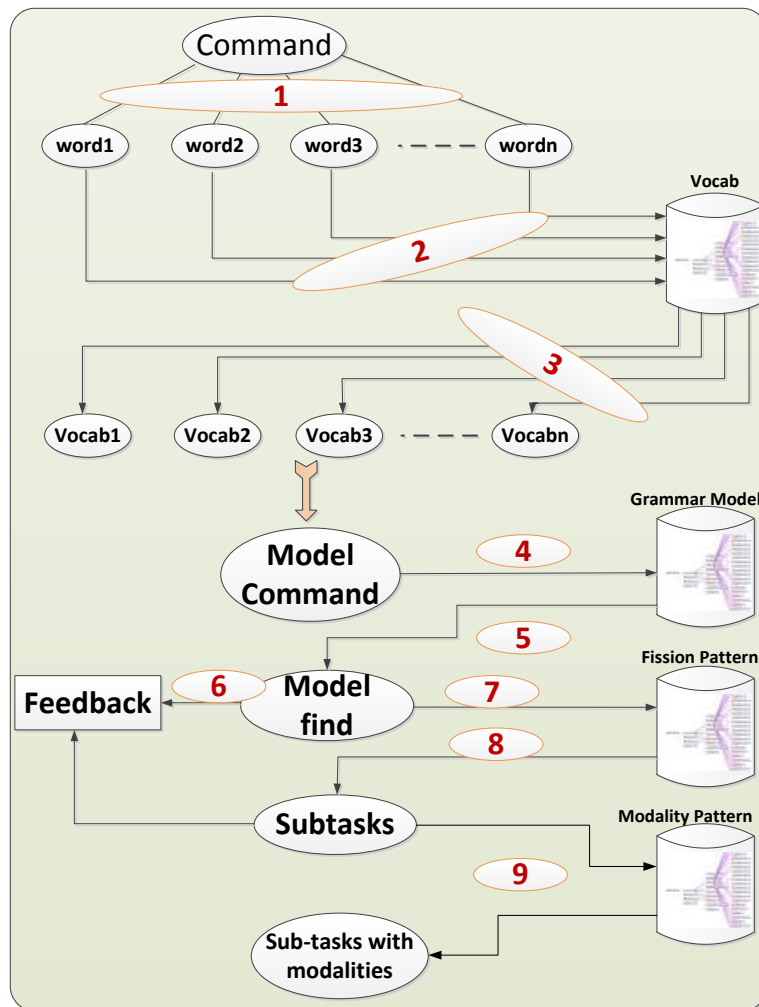


Figure 6. Stages of fission process.

Step-8: if no matching pattern is found, a feedback is sent to the user.

Step-9: every subtask is associated with the appropriate and available modality or modalities.

This is done by sending a query to find the matching pattern modality.

VI. PROTOTYPE'S IMPLEMENTATION

The prototype focuses on multimodal fission, the main idea behind this work. We are interested in the selection of modalities and the fission process, so that the system will be able to understand the commands sent by the user and adapt to any change in the environment.

We used the JAVA programming language to implement the four modules in our system (Figure 7), and we used four computers connected to the Internet to implement it. Table 1 describes each computer and the module it contains.

Table 1 Characteristics of the computers used.

Computer	specifications	Module
A	Operating System: XP Internet Connection: WIFI IP Address: 192.168.2.100 Processor: Xeon (TM) CPU 3.40 GHz	GPS Application
B	Operating System: Win7 Internet Connection: file IP Address: 192.168.2.42 Processor: Intel® Core (TM) 2 CPU 1.67 GHz	Robot Application
C	Operating System: Win7 Internet Connection: file IP Address: 192.168.2.42 Processor: Intel® Core (TM) 2 CPU 1.67 GHz	Fission
D	Operating System: XP Internet Connection: file IP Address: 192.168.2.68 Processor: Intel®Core (TM) 2CPU 1.67 GHz	Ontology

As shown in Figure 7, stage (1) consists of sending xml files from user 1 (computer A) or user 2 (computer B) to the computer that contains the fission module, in our case computer D. This module receives files – stage (2) – and extracts the complex command. In stages (3), (4), (5), and (6), the fission module interacts with the ontology (computer C) to divide the complex command into elementary subtasks associated with the appropriate modalities. Once the fission process is completed, the system sends the result to the appropriate application –stage (7) – and the result is presented to the user in stage (8).

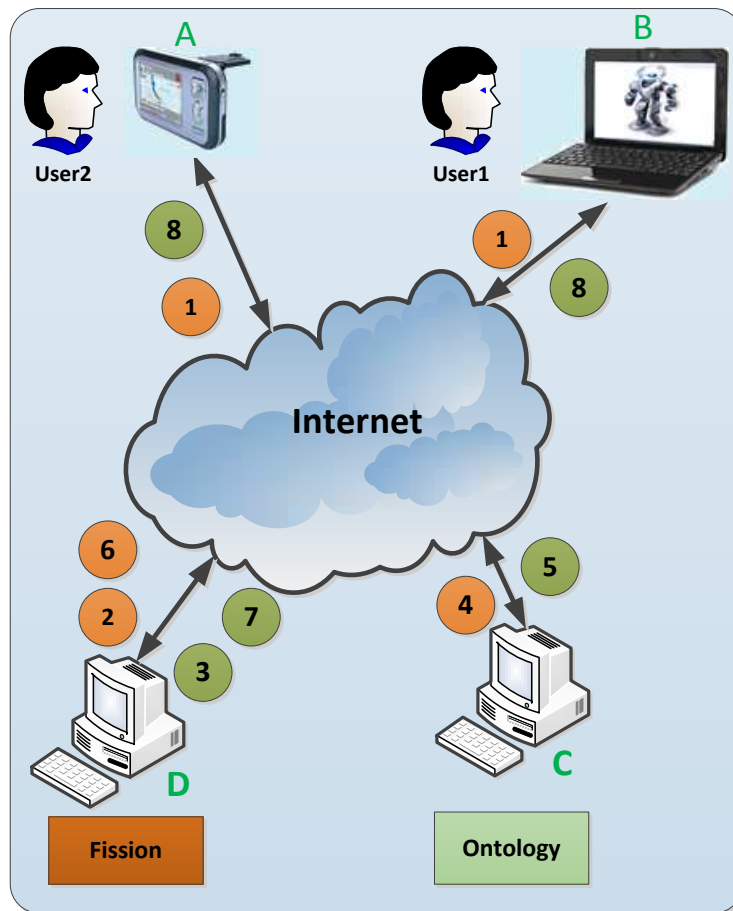


Figure 7 System's implementation.

The files are transferred from one module to another using the Socket class implemented in Java [20]. Two graphic interfaces are created for the implementation of the prototype: robot control interface (Figure 8, corresponding to B in Figure 7) and GPS interface (Figure 9, corresponding to A in Figure 7). The robot interface is used to validate our fission module and the GPS interface is used to validate the BN with the context in the case of ambiguity or uncertainty.

a. ROBOT CONTROL INTERFACE

This interface is made up of five components: modalities, context, command, elementary subtasks, and actual execution.

The component context affects the selection of the modalities. Normally, contextual information is captured from sensors installed in the environment. In our simulations, the system randomly generates different values (Figure 8). Suppose we have the following information:

- the level of the noise is high ;
- the level of brightness is good ;
- the temperature is 19 °C ;
- the bandwidth is good ;
- the user is not physically challenged.

As the level of noise is high, the context will affect only the audio modalities. In the command component, the command chosen by the user to be executed is “Move the circle to (205, 175).” When the user clicks on “Start”, an XML file containing the complex command is created and sent to the Fission module, which is located in computer D. The system follows the steps of the algorithm presented in section 5. The result of the command is presented in the elementary subtasks component: {Move to the circle at position (152, 58), Take the circle, Move the circle to position (205, 175), and Drop the circle}. Finally, the actual execution component represents the subtask that the system is in the process of executing and the appropriate modalities associated with the subtask. In our example, the results are:

- Subtask: move the circle (152, 58)
- Modalities: {mobility mechanism, Screen, printer}

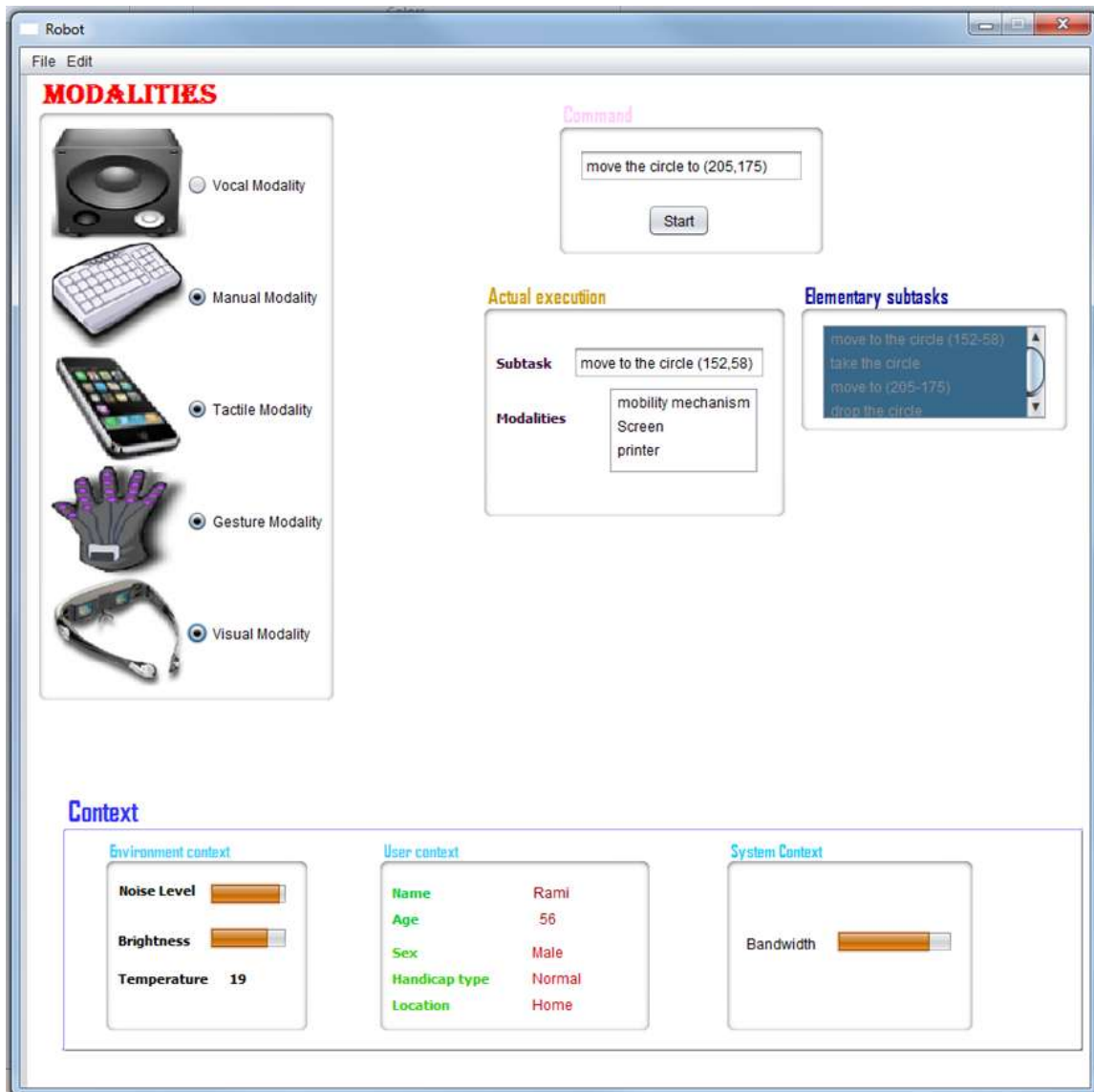


Fig. 8 Robot control interface

b. GPS INTERFACE

This interface is implemented to illustrate the use of BN with the context. As shown in Figure 9 (a), the main interface is composed of two components: Context and Command. For the Context component, we used time, actual location, user status, and temperature. The values of these contexts are chosen randomly by the system (Table 2).

Let us say that the command entered by the user is “I want to go to Montreal.” When the user clicks on “Start”, the fission process begins. In this case, “Montreal” has many meanings in the ontology. This puts us in an uncertain situation.

Table 2 Context's parameters.

Context	Value
Time	12:45
Actual location	Montreal
User status	Hungry
temperature	22

Figure 9 (b) shows the detection of the uncertain case {City, Restaurant, Street}. When we click on Decision Process, the system uses equation (2) and the algorithm presented in [19] to choose the best situation according to the context. The decision taken is shown in Figure 9 (c): Restaurant is the most probable request. The more contextual parameters we add, the more accurate the result will be.

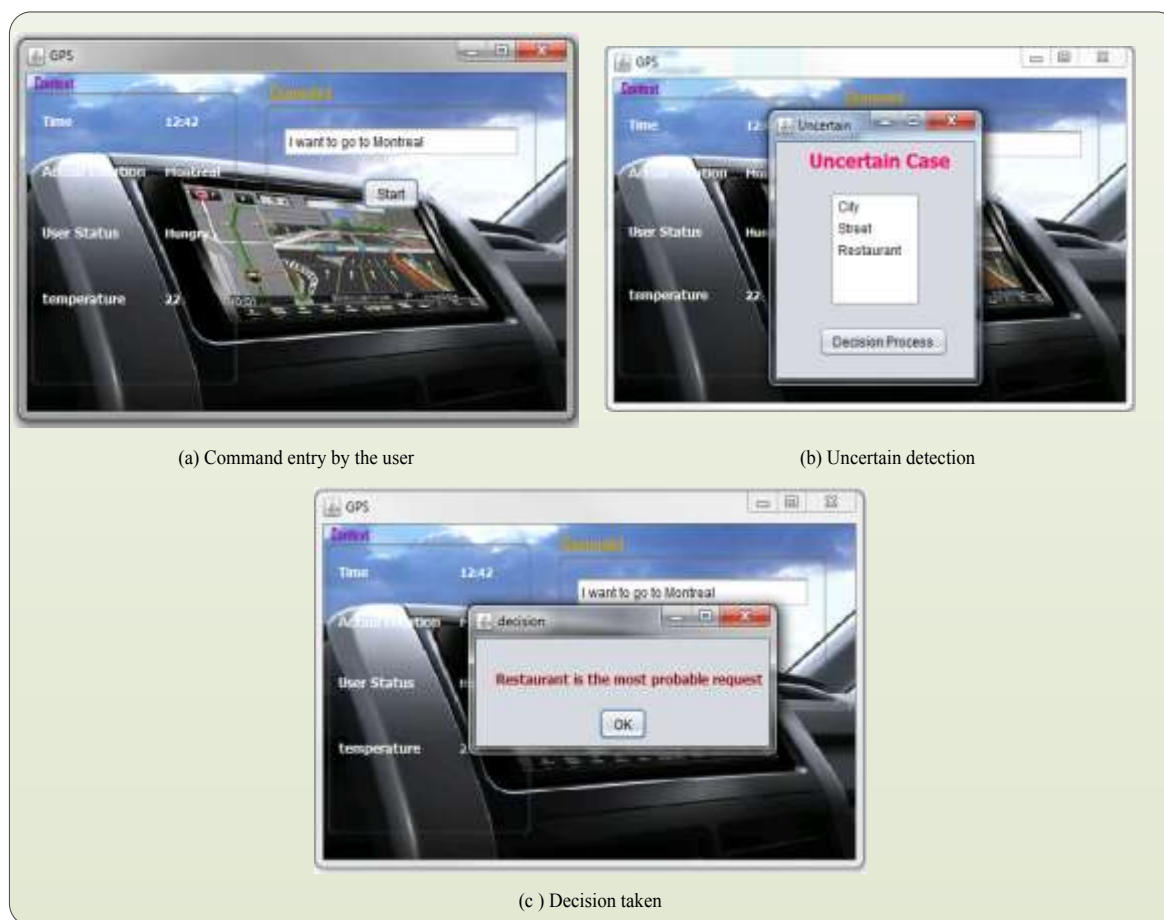


Figure 9. GPS interfaces.

VII. CONCLUSION

In this paper, we presented a very useful architecture for a fission system which allows multimodal interaction. In this interaction architecture, several natural input/output modalities (speech, pen, touch, hand gesture, eye gaze, and head and body movements) can be considered.

We have shown how abstract concepts, such as: i) a fission process using a pattern technique; and ii) a context-based method using a Bayesian network, can be used to develop applications.

We implemented two real applications to validate our approach, and showed that the proposed solution is applicable in a real environment. We presented two interfaces: 1) a robot control interface, which is implemented to validate the fission process using the pattern technique stored in the ontology, and 2) a GPS interface, which is implemented to validate the context in a case of uncertainty or ambiguity.

REFERENCES

- [1] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia Systems*, vol. 16, pp. 345-379, 2010.
- [2] H. Djenidi, S. Benarif, A. Ramdane-Cherif, C. Tadj, and N. Levy, "Generic multimedia multimodal agents paradigms and their dynamic reconfiguration at the architectural level," *EURASIP J. Appl. Signal Process.*, vol. 2004, pp. 1688-1707, 2004.
- [3] R. Bolt, "Put-that-there," *Voice and gesture at the graphics interface ACM SIGGRAPH Computer Graphics*, vol. 14, pp. 262-270, 1980.
- [5] Turk, Matthew. "Multimodal interaction: A review." *Pattern Recognition Letters* 36 (2014): 189-195.
- [6] T. D. C. Little, C. Y. R. Chen, C. S. Chang, and P. B. Berra, "Multimedia Synchronization," *EEE Data Eng. Bull.*, vol. 14, pp. 26-35, 1991.
- [7] S. Oviatt, P. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro, "Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions," in *Human-Computer Interaction*. vol. 15, ed: Lawrence Erlbaum Associates, 2000, pp. 263-322.

- [8] R. Raisamo, A. Hippula, S. Patomaki, E. Tuominen, V. Pasto, and M. Hasu, "Testing usability of multimodal applications with visually impaired children," *MultiMedia*, IEEE, vol. 13, pp. 70-76, 2006.
- [9] M. Debevc, P. Kosec, M. Rotovnik, and A. Holzinger, "Accessible Multimodal Web Pages with Sign Language Translations for Deaf and Hard of Hearing Users," in *Database and Expert Systems Application*, 2009. DEXA '09. 20th International Workshop on, 2009, pp. 279-283.
- [10] J. Lai, S. Mitchell, and C. Pavlovski, "Examining modality usage in a conversational multimodal application for mobile e-mail access," *International Journal of Speech Technology*, vol. 10, pp. 17-30, 2007/03/01 2007.
- [11] M. C. Caschera, A. D'Andrea, A. D'Ulizia, F. Ferri, P. Grifoni, and T. Guzzo, "ME: Multimodal Environment Based on Web Services Architecture," presented at the OTM 2009 Workshops, Vilamoura, Portugal, 2009. pp 504-512.
- [12] A. Karpov, A. Ronzhin, I. Kipyatkova, and L. Akarun, "Multimodal Human Computer Interaction with MIDAS Intelligent Infokiosk," in *Pattern Recognition (ICPR)*, 2010 20th International Conference on, 2010, pp. 3862-3865.
- [13] G. P. Laput, M. Dontcheva, G. Wilensky, W. Chang, A. Agarwala, J. Linder, and E. Adar, "PixelTone: a multimodal interface for image editing," presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Paris, France, 2013.
- [14] A. Zaguia, M. D. Hina, C. Tadj, and A. Ramdane-Cherif, "Using Multimodal Fusion in Accessing Web Services," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 1, pp. 121-138, 2010.
- [15] D. Costa and C. Duarte, "Adapting Multimodal Fission to User's Abilities," in *Universal Access in Human-Computer Interaction. Design for All and eInclusion*. vol. 6765, C. Stephanidis, Ed., ed: Springer Berlin Heidelberg, 2011, pp. 347-356.
- [16] D. Perroud, L. Angelini, O. Abou Khaled, and E. Mugellini, "Context-Based Generation of Multimodal Feedbacks for Natural Interaction in Smart Environments," in *AMBIENT 2012, The Second International Conference on Ambient Computing, Applications, Services and Technologies*, 2012, pp. 19-25.
- [17] Honold, Frank, Felix Schussel, and Michael Weber. "The Automated Interplay of Multimodal Fission and Fusion in Adaptive HCI." *Intelligent Environments (IE)*, 2014 *International Conference on*. IEEE, 2014.
- [18] Schnelle-Walka, Dirk, Stefan Radomski, and Max Mühlhäuser. "Multimodal Fusion and Fission within W3C Standards for Nonverbal Communication with Blind Persons." *Computers Helping People with Special Needs*. Springer International Publishing, 2014. 209-213.
- [19] A. Benoit, L. Bonnaud, A. Caplier, P. Ngo, L. Lawson, D. G. Trevisan, V. Levacic, C. Mancas, and G. Chanel, "Multimodal focus attention and stress detection and feedback in an augmented driver simulator," *Personal and Ubiquitous Computing*, vol. 13, 2009. pp 33-41.

- [20] A. Zaguia, A. Wahbi, M. Miraoui, C. Tadj, and A. Ramdane-Cherif, "Modeling Rules Fission and Modality Selection Using Ontology," *Journal of Software Engineering and Applications*, vol. 7, pp. 354-371, 2013.
- [21] A. Zaguia, A. Wahbi, C. Tadj, and A. Ramdane-Cherif, "Multimodal Fission For Interaction Architecture," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 4, February 2013 2013. pp. 152-166.
- [22] A. Zaguia, M. D. Hina, C. Tadj, and A. Ramdane-Cherif, "Interaction context-aware modalities and multimodal fusion for accessing web services," *Ubiquitous Computing and Communication Journal*, vol. 5, N 4, pp.1-15, 2010.
- [23] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *Vehicular Technology, IEEE Transactions on*, vol. 53, pp. 1052-1068, 2004.
- [24] Java. (2013). Java Socket. Available: docs.oracle.com/javase/tutorial/networking/sockets/definition.html
- [25] Zhang, Fei, Wuying Liu, and Yude Bi. "Review on Wordnet-based ontology construction in China." *International Journal on Smart Sensing and Intelligent Systems* 6.2 (2013). pp.630-647.
- [26] Guessoum, Djamel, Moeiz Miraoui, and Chakib Tadj. "SURVEY OF SEMANTIC SIMILARITY MEASURES IN PERVASIVE COMPUTING." *International Journal on Smart Sensing and Intelligent Systems* 8.1 (2015). pp 125 – 158.
- [27] Pavlakos, Georgios, Stavros Theodorakis, Vassilis Pitsikalis, S. Katsamanis, and Petros Maragos. "Kinect-based multimodal gesture recognition using a two-pass fusion scheme." In *Proc. Int'l Conf. on Image Processing*. 2014. pp. 1495 – 1499.