MODELING ONTOLOGY FOR MULTIMODAL INTERACTION IN UBIQUITOUS COMPUTING SYSTEMS

Ahmad Wehbi
LISV Laboratory, University of Versailles-Saint-Quentin-en-Yvelines
10-12 avenue de l'Europe, 78140 Vélizy, France
Ahmad.wehbi@lisv.uvsq.fr

Amar Ramdane Cherif
LISV Laboratory, University of Versailles-Saint-Quentin-en-Yvelines
10-12 avenue de l'Europe, 78140 Vélizy, France
rca@prism.uvsq.fr

Chakib Tadj
MMS Research Group, University of Québec, École de Technologie Supérieure
1100, rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3
ctadj@ele.etsmtl.ca

ABSTRACT
People communicate with each other using different ways, such as words, gestures, etc. to give information about their status, emotions and intentions. But how may this information be described in a way that autonomous systems (e.g. Robots) can react with a human being in a given environment?
A multimodal interface allows a more flexible and natural interaction between a user and a computing system. This paper presents a methodological approach for designing an architecture that facilitates the work of a fusion engine. The selection of modalities and the fusion of events invoked by the fusion engine are based upon the definition of an ontology that describes the environment where a multimodal interaction system exists.

Author Keywords
Conference publication.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Multimodal system, multimodal fusion, modality, ontology, events, environment, Environmental and user context.

INTRODUCTION
Ubiquitous computing is a post-desktop model of human-computer interaction in which information processing has been thoroughly integrated into everyday objects and activities. An important type of ubiquitous computing systems are the multimodal systems, were a user communicates with a system, using different types of modalities (gestural, vocal, etc.) by sending commands, that are called events in this paper. The user is an object of the environment; the environment is the place where a user and a multimodal system exist, it could be indoor (home, office, etc.) or outdoor (road, train station, etc.). If a multimodal system is inside a room for example, it tries to identify: different objects existing in this room (chair, table, window, door, etc.), information about environmental context in the room (lightness, darkness, noise level, etc.) to choose the most suitable modality according to the context. A Multimodal system receives events (at least two) in order to take a decision and react (e.g. moving an object, give a ball to the user, etc.). That decision is affected by the way of understanding these events and the multimodal fusion will take place if all preconditions are respected.
Various multimodal applications [1-2] are conceived and are effective solutions for users who have constraints such as the impossibility of using a keyboard or a mouse [3], having visual handicap [4], being in move, using mobile devices [5], and being weak or disabled [6].
In multimodal interactive systems, multimodal fusion is a crucial step in merging various input modalities (e.g. speech, gesture, eye gaze, etc.). Since the work of Bolt [7], the academic world has provided models and designs offering multimodal interaction techniques using the fusion engines [8], [9], [10], [11], [12]. Currently, the engineering of these engines still remains complex and cumbersome, either in design, specification or validation.
This paper proposes a new methodological solution by modeling an architecture that facilitates the work of a fusion engine, by defining an ontology that contains different applicable scenarios and describes the environment where a multimodal system exists. The proposed architecture has three main characteristics:
- *Openness*: handling a large number of modalities that prevents the restriction in its application to specific domains.
- *Flexibility*: the use of ontology makes the description of an environment and its scenarios easier.
- *Consistency:* by the description of the most potential objects and scenarios of the environment.

This paper discusses these characteristics by explaining the architectural design of the proposed solution. The paper is organized as follows. In related work, we will review some

UbiComp' 12, Sep 5 – Sep 8, 2012, Pittsburgh, USA.

previous architectural designs and show their weaknesses. In the architectural design, we will present the architecture itself and describe its components. In the ontology definition, we will present the created ontology, in the modality selection, we will present the selection of modalities according to a context, the fusion algorithm, we will present the fusion algorithm of matching, in applicable scenario, a scenario is discussed, and finally, the conclusion.

RELATED WORK

Modality refers to the path or channel by that human and machine interact with each other. An impoverished traditional computing set-up uses mouse, keyboard and screen by which the user interacts with the computer and vice-versa. The concept of multimodalities allows the use of other means, or modality, in sending events to the machine as well as the machine sending output to the user. Such modalities include other gadgets and sensors, such as touch screen, stylus, etc. and man's natural modalities, such as speech, eye gaze and gestures. The invocation of multimodalities permits a more flexible interaction between user and machine as well as allowing users with temporary or permanent handicap to benefit from the advancement in technology in undertaking computing tasks. Multimodality allows the invocation of other means when some other modalities are not available or possible to use. For example, speech [13] may be a more effective input modality than a mouse or a keyboard if the user is on the go. Using multimodality in accessing user application is an effective way of accomplishing user's computing task. When two or more modalities are invoked at the same time (e.g. speech and clicking a mouse button), the user invokes complementarities of these modalities. This provides a rationale for the invocation of multimodal fusion and the fusion engine which is responsible for determining the meaning of such complementarities.

In the literature, several solutions were proposed to facilitate the work of fusion engines. To our point of view, the most significant works follow. Engel et al [14] proposed an approach for processing modalities in a system called *SmartKom* [15]. Its basic idea is to generate all meaningful combinations after considering all hypotheses and afterwards selects the *n* best results which are passed to the intention analyzer. Apart from its relatively easy implementation, its other advantage is that during the processing, no decisions based on incomplete context have to be made. The disadvantage of this approach is that under adverse circumstances, specifically in the case of recognition errors or unintended gestures done by the user, the generation of all meaningful combinations takes too much time. Sonntag et al [16] proposed an ontological solution for a system called *SmartWeb*. This system is based on question answering technology that combines different kinds of domain ontologies into an integrated and modular knowledge base. For this purpose, they defined an upper model ontology based on SUMO [17] and DOLCE [18] and integrated domain ontology on it. The main problem of this approach is the specification of its design. Since it is only for the application of answering questions, the solution presented by this work is very limited. The architecture of *HephaisTK* system developed by Dumas et al [19] is based on software agents that are dispatched to manage individual modality recognizers, receive and encapsulate data from the recognizers, and send them to an individual central agent named the "*postman*". This postman agent centralizes all data coming from the dispatched recognizers agents in the database, and distributes the data to other interested agents like the fusion manager. However, this architecture needs a configuration file to be specified for describing the human-machine multimodal dialog desired for the client application, and for the specification to which recognizers need to be used.

Multimodal interface tools are currently few in numbers and they address to a very specific technical problem such as media synchronization [20], or they are dedicated to very specific modalities. For example, the Artkit toolkit which is designed to support direct manipulation augmented with gesture only [21] or MATIS (Multimodal Airline Travel Information System) which allows a user to retrieve information about flight schedules using speech, direct manipulation, keyboard and mouse, or a combination of these techniques [22].

Having these weaknesses taken into account, we come up with a proposed architecture that addresses these issues. This is done through environment description that contains all possible modalities, objects and scenarios. The adoption of this architecture will facilitate the work of a fusion engine by giving it the most meaningful combinations of events.

ARCHITEURAL DESIGN

In this section, we will describe our proposed architectural design as a solution to the described problems of previous architectures.

General Approach

A general overview of the architectural design is shown in Figure 1. It illustrates the different components of the environment and show how a user or objects supplies events using different modalities leading to the fusion process by the fusion engine inside the multimodal system. These components are as follows:

- *Environment:* it is the place where the multimodal system exists; it could be indoor or outdoor.
- *Ontology:* is the knowledge base that describes every detail in the environment (objects and events, etc.).
- *Modalities selection*: is the part responsible of the selection of a modality according to a specific context. Modalities are affected by two types of contexts:
  - *Environmental context:* it describes the lightness, darkness, noise level, and temperature level in the place where the multimodal system exists.
  - *User Context:* it describes the user profile, especially if he has any handicaps.

• *Fusion:* is the part responsible of merging events coming from modalities according to a set of preconditions defined in the ontology.
• *Multimodal fusion system:* it is the system that contains the fusion engine, the ontology and the selection mechanism.
• *Decision:* The final meaningful result obtained after the events has been merged which is a command that have a meaning and could be executed.
• *Multimodal fission system:* it is the part responsible of executing the commands after understanding them in the fusion process. The command given by the fusion engine is used by the fission system to make the robot execute an order.
The multimodal fusion system will interpret the environment, merge the events coming from the input modalities and then produce the complex command. After, the multimodal fission system will interpret this command and will divide it into elementary tasks in order to present them to the available output modalities in the environment. The execution of these tasks requires the discovery and selection of appropriate services to fulfill these various tasks. In this paper, we will not deal with the fission system; we are only interested in the fusion that merges events coming from modalities.

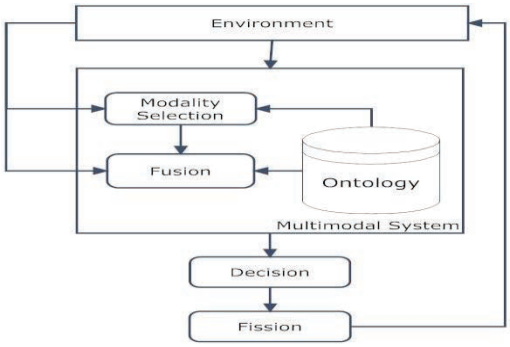

Figure 1: General approach of multimodal fusion system with the fission module

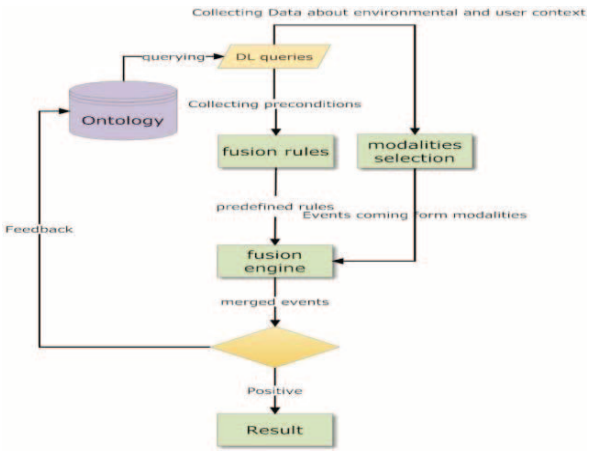

Figure 2: General architecture

General Architecture

The general approach is further refined (see Figure 2) showing the whole concept of the architecture. The remaining system components and the description of various steps involved in the fusion process are described below:

- **Ontology**: it describe in detail:
  - the environment in which a scenario takes place
  - events that may occur
  - the environmental context and the user context
  - modalities and their relation with the environmental and user contexts

The ontology will be created using a tool called PROTÉGÉ [23] , an authoring system for creating ontologies.

• **Modality selection rules**: to select the most appropriate modality according to a specific context. They will be described using the description logic queries of OWL semantic language (OWL-DL query) .These queries are used to collect all of the information about a concept, a property and an instance of the ontology. The closest modality to the needed one is selected using a simple DL query that takes into account the environmental and user context,

• **Fusion rules**: set of inferences rules, used by the fusion engine to verify the semantic between different events. They are described using DL queries and SWRL (semantic web rule language) [24].

Fusion engine: is a module responsible for merging different events according to fusion rules. The fusion process will give a decision, it could be positive, which mean that, the multimodal system has understood the environment, or negative and a feedback will occur.

ONTOLOGY CREATION

As mentioned before, the role of the ontology is to describe the environment surrounding the user and the multimodal system. It is described using a tool called PROTÉGÉ. This tool is based on Ontology web language (OWL). The OWL Web Ontology Language [25] is a language for defining and instantiating *Web ontologies*. *Ontology* is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related. OWL *ontology* may include descriptions of *classes*, *properties* and their instances. Given such ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but *entailed* by the semantics. **Figure 3** represents the main concepts needed to describe the environment for any scenario. Each scenario must have:

• **User:** sends commands to a multimodal system.
• **Events:** that can be sent by a user using modalities and also by the environment.
• **Object**: which must be recognized by the system, it could be an object to move from a place to another for example.

• **Place:** it describes the place where the scenario is happening. For example it could be in a room inside a home.
• **Modality:** it contains the different modalities used by a scenario
• **User Context:** it describes the user profile, especially if he has a handicap or no, because this will affect the choice of modality
• **Environmental Context:** it describes the lightness, darkness, noise level, etc. inside an environment.

Figure 4 represents the two concepts *Event* and *Modality.* As shown, *Event* has a subclass called *EventType*, that specify the type of an event; it could be *Talking, Pointing*, etc. and another sub class called *Time* that indicates the *StartTime* and the *EndTime* of an event. *Modality* class has two sub classes, which are types of modalities that can be used like *GestualModality*, *VocalModality.* (*It* may contain other modalities but we are showing two only). A relation exists between *Modality* and *Event*, called *HasEvent,* because each modality has its own set of events, for example talking is an event of a vocal modality, but it could not be an event of a gestural modality. This type of restrictions is defined using a function inside PROTÉGÉ called *disjoint classes* which is an OWL build in function

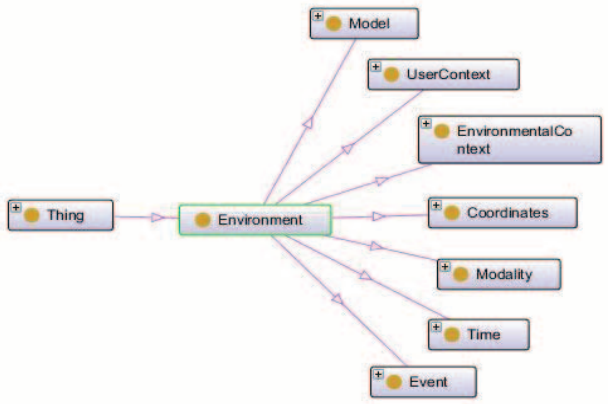

Figure 3: The main concepts of the ontology

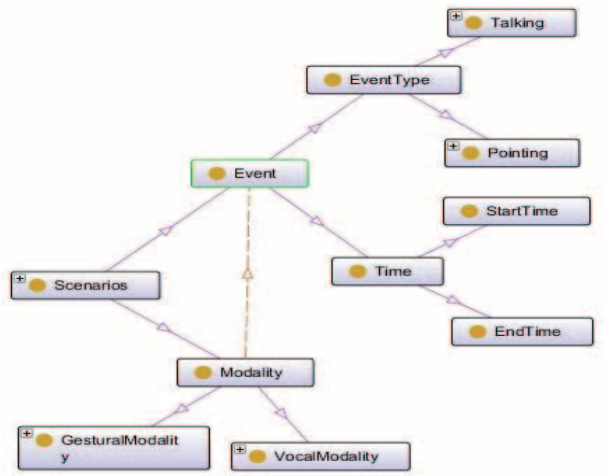

Figure 4: Event and Modality Concepts

Figure 5 represents the *Event* class and the relation of its subclasses with *Modality* class. *Event* has the subclasses *Time* and *EventType. Time* has subclasses *StartTime and EndTime* which are the start and end time of an event, *EventType* has the subclasees *Talking* and *Pointing (*another type of events can be defined but we are showing only these two events). An object property called HasEvent is defined between *VocalModality* class and *Talking* class, and between *GesturalModality* class and *Pointing* class to tell that a vocal modality has an event called talking, and a gestural modality has an event called pointing respectively.

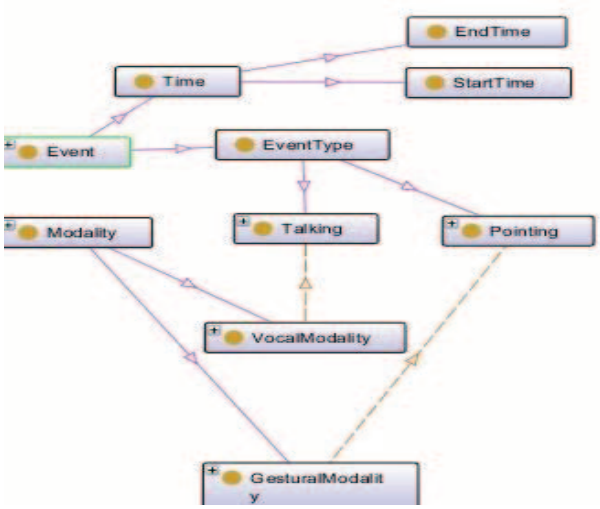

Figure 5: Events and Modality Concepts

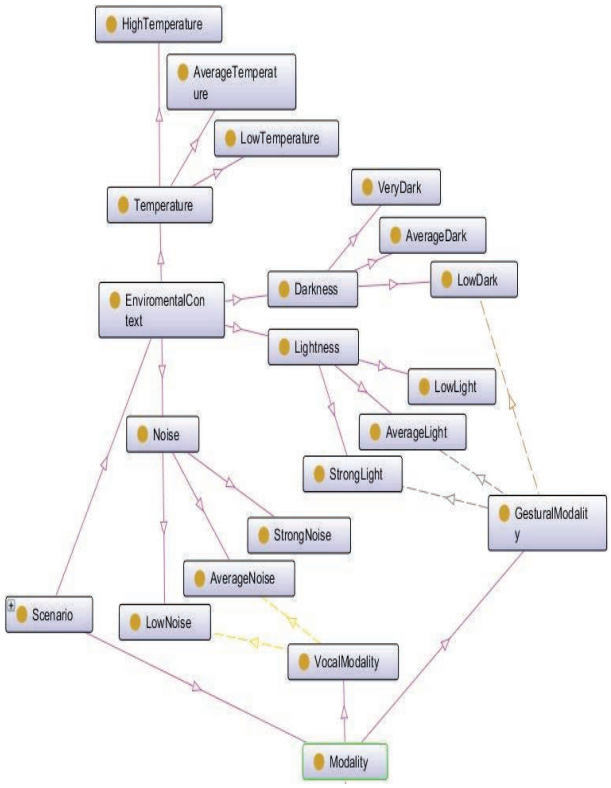

Figure 6: Environmental Context and Modality concepts

Figure 6 represents the *EnvironmentalContext* class and the relation of its subclasses with the *Modality* class. *EnvironmentalContext* has subclasses *Temperature, Darkness, Lightness, and Noise.* These subclasses describe the environmental context. As we said before, a system could be indoor or outdoor. *Temperature* has the subclasses *HighTemperature, AverageTemperature, LowTemperature*. The same thing for *Darkness*, *Lightness* and *Noise* classes, each of them has three subclasses, *VeryDark, AverageDark, LowDark, LowLight, AverageLight, StrongLight, StrongNoise, AverageNoise, LowNoise* respectively. Each of them has two object properties called *HasMaximumLevel* and *HasMinimumLevel.* They took an interval of instances that identify if the temperature, darkness, lightness and the noise of the place where the multimodal system exist is in the high, average or low range. The choice of modalities is affected by the environmental context, that's why, we have create restrictions as object properties to identify which modality can be active in specific context. As shown, there is a relation between *GesturalModality* and *strongLight* and *AverageLight* called *hasLightnessLevel* (in dashed black). The main goal of this relation is to tell that gestural modality can be active if the level of light is strong or average, but it will be deactivated if it is low. Also there is a relation between *GesturalModality* and *LowDark,* called *HasDarknessLevel* (in dashed red) which mean that this modality can be active if the level of darkness is low. Another relation called *HasNoiseLevel* (in dashed yellow) existing between *VocalModality* and *LowNoise* and *AverageNoise,* its goal is to tell that, a vocal modality can be active if the level of noise is low or average but not strong.

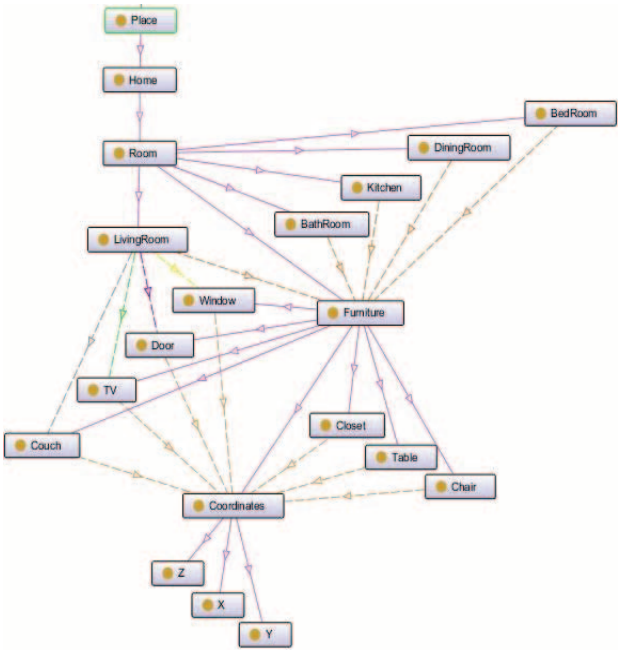

Figure 7: Place concept and its subclasses

In Figure 7, *Place* class has *Home* as subclass, *Room* has subclasses *BedRoom, BathRoom, DinningRoom, Kitchen, LivingRoom and Furtinure.* Each room of the home has furniture; this is described using an object property called *HasFurniture* (in dashed red). *Furniture* has subclasses *Chair, Table, Closet, TV, Couch, Door and Window* (we can add more furniture). Each one of these subclasses is related to the *Coordinates* class by an object property called *HasCoordinates* (in dashed orange). *Coordinates* has *X, Y* and *Z.* In LivingRoom class we identified that it contains a window, door, TV and a couch using objects properties called, *HasTV, HasDoor, HasWindow, HasCouch* (in dashed green, violet, yellow and black respectively)

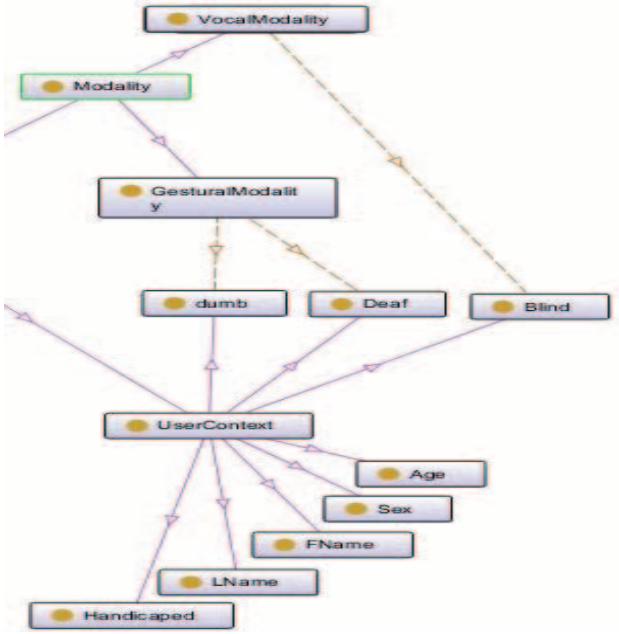

Figure 8: UserContext and Modality Concepts

Figure 8 represents the relation between a user context and modalities. *UserConetxt* Class has subclasses *Handicaped, LName, FName, Sex, Age, Dumb, Deaf* and *Blind.* A modality is affected by the user context, for example if a user is blind, he can't use the gestural modality and if he is deaf, he can't use the vocal modality, this is described using object property called *HasUserConext* (in dashed red).

MODALITIES SELECTION

As stated, *modality*, in this work, refers to the logical structure of man-machine interaction, specifically the mode by which data is entered and presented as a result between a user and computer. Using natural language processing as basis for categorization, we classify modalities into 5 different groups:

- **Tactile Modality (T_m)** – the user uses the sense of touch to input data.
- **Vocal Modality (VO_m)** – voice or sound is captured and becomes input data.
- **Manual Modality (M_m)** – data entry is done using hand manipulation or stroke.

• **Visual Modality (V_m)** – movement of human eyes are interpreted and considered as data input.
• **Gestural Modality (G_m)** – human gesture is captured and considered as data input.

For our intended application, we take into account parameters that specify whether a modality is suitable or not. If there is another type of modalities, they can be added to the ontology.

User Context:
• **User handicap** – it affects the user's capacity to use a particular modality. We note four handicaps, namely Manual handicap, Muteness, Deafness, and Visual impairment.
• **User location** – we differentiate between fixed/stationary locations, such as being at home or at work where user is in a controlled environment to that of a mobile location (on the go) where user generally has no control of what is going on in the environment.
Environmental Context
• **Noise level** – the noise definitely affects our ability to use audio as data input or receiving audio data as output.
• **Brightness of workplace** – The brightness or darkness of the place (i.e. to the point that it is hard to see things) also affects our ability to use manual input and modalities.
• **Darkness of workplace** – The darkness of the place also affects our ability to use manual input and modalities.
To summarize, a modality is appropriate to a given instance of interaction context if it is found to be suitable to every parameter of the user context and the environmental context. This is shown be a series of relationships given below:

$$T_m = (user \neq manual\ handicp) \wedge (location \neq road) \wedge (workspace = very\ dark)$$

$$VO_m = (user \neq mute) \wedge (location \neq road) \wedge (noise\ level \neq strong\ noise)$$

$$M_m = (user \neq manually\ handicaped) \wedge (workspace \neq average\ dark \vee worksapce \neq very\ dark)$$

$$VI_m = (user \neq blind) \wedge (location \neq road)$$

$$G_m = (user \neq manual\ handicap \vee user \neq blind) \wedge (workplace \neq average\ dark \vee workplace \neq very\ dark \vee workplace\ low\ light)$$

Once different modalities and their relations with the environmental and user context, (e.g. *GesturalModality HaslightnessLevel AverageLight*) are defined inside the ontology, we will be able to select the most appropriate modality according to a context using DL queries.

The rule below asks the ontology to select modalities that work in an average or high level of lightness and noise. According to specified conditions, GesturalModality and *VocalModality* are obtained as result.

"Modality and HasNoiseLevel some LowNoise or AverageNoise or HasLightnessLevel some StrongLight or AverageLight"

The formal representation of the general rule responsible of the selection of any type of modalities is as follows:

$$Q_s = C_M \wedge (P_0 \exists (C_1 \vee C_2 \vee \ldots \vee C_i) \vee P_1 \exists (C_1 \vee C_2 \vee \ldots \vee C_i) \vee \ldots \vee P_i \exists (C_1 \vee C_2 \vee \ldots \vee C_i))$$

Where C_M represents the *Modalities* concept inside the ontology, $P_{0\ldots i}$, represent the object properties (relations) between a modality and the contexts (e.g. *HasLightnessLevel, HasUserContex, ect.*) and $C_{1\ldots i}$ represent the different contexts of the environmental and user (e.g. *LowLight, StrongNoise, Deaf, Blind, etc.*).

FUSION ALGORITHM

Fusion rule is a set of preconditions and constraints needed for multimodal fusion. We must take into account several conditions, these conditions are:
• **Vocabulary Checking**: to verify if an event exist in our defined vocabulary or not.
• **Events order**: to verify if events sent from modalities are respecting the expected order defined in different models of the class *Model* in the ontology.
• **Temporal conditions:** to verify the maximum of time allowed between two active modalities and the time of a command itself.
Theses preconditions are very important for the fusion engine, because, first: the meaning of an event must be identified and defined in the ontology which contain all comprehensible words by the multimodal system so the system could understand what is receiving. Second, the order is an important precondition especially in the fission process where the fission system must execute a well formed command and third, the temporal aspect is also important, because the system must knows if the received events are probably for the same command or not. for example if a user say "put" and he wait for 40 sec, the system must ask the user to continue his command or the events will be removed.
Once we have specified the preconditions needed, we must define an algorithm that checks the preconditions inside the ontology.

1	*Declaring Command as string;*
2	*String [] instance=words of the command;*
3	*For (i=0, i<=instance.lenght, i++){*
4	*getInstanceOf (Instace [i]); // get the instances of the command from the ontology*
5	*//check if events of the commands exist in the instances vocabulary of the ontology*
6	*If (instance [i] = instances in the ontology) {*
7	*Memorize instance [i];*
8	*Check insatance [i+1]; }*
9	*//check maximal command time;*
10	*GetStartTime of event1;*

11	*GetEndTime of last event;*
12	*If (endTime – startTime) <=MaxCommandTime){*
13	*//check time between modalities*
14	*ΔT= startTime of Mi+1 – endTime of Mi;*
15	*If (ΔT < MaxActiveModalityTime){*
16	*//Match a model of ontology with the command*
17	*getClassesOfInstance;*
18	*// check their order if exists in a model defined inside the ontology*
19	*If (Matching ok);*
20	*Result of fusion;}}*
21	*Else*
22	*No fusion;*
23	*Else*
24	*No fusion;*
25	*Else*
26	*No fusion;*
27	*Else*
28	*No fusion;}*

Table 1: fusion algorithm

APPLICABLE SCENARIO

To understand the mechanism of fusion in Table 1, we will explain it by elaborating a simple known scenario "put that there". In this scenario, we are assuming that the multimodal system presented by a robot must move an object from its initial position to another location. The user uses 2 types of inputs modes, his voice by saying "put that here" and his hand by pointing to the object on the initial location and the new location. The multimodal system is inside a room with normal environmental context which mean that, there is light and no noise in the room.

Multimodal fusion system must realize the *fusion* process of three events coming from two different input modalities. These 3 events are:

- the spoken words "put that there"
- the pointing to an object in its initial location
- The pointing to the new location where the object to be moved

N	event	Start Time	End Time	Vocabu-lary	Modlituy
1	Put	0.5	1.5	AFMO	Vocal
2	That	1.5	2.5	IO	Vocal
3	Point	2	3	(x,y,z)	Gestural
4	There	3.1	4	IL	Vocal
5	point	3.8	4.5	(x',y',z')	Gestural

AFMO: Action for movable object
IO: Intended object
IL: Intended location

Table 2: Preconditions used by the fusion

Table 2 shows the order that must be followed by each event (N°), the start and end time (in seconds) of each event sent by the user to the multimodal system using modalities (Modality) and the vocabulary of events as defined in the ontology. These data must be checked according to the algorithm so that the fusion can occur. First, the vocabulary is checked to identify if words exist as instances in the ontology, if true, the temporal aspect is checked by verifying if the maximum time of command (*MaxCommandTime*) and the time between the events that came from different modalities (M*axTimeActiveModality*), if true, the order of events must be checked by matching the order of events with a model defined in the ontology, in this example the used model is AFMO→IO→IL. If these entire preconditions are respected, fusion occurs and if one of these conditions is false, in this case there will be no fusion.

CONCLUSION

In this paper, we presented an architecture that is very useful in a multimodal system. We developed ontology for designing multimodal fusion architecture that allows multimodal natural interaction. In this interaction system several natural input modes (speech, pen, touch, hand gestures, eye movement, head and body movements) can be investigated. They are ultimately aiming the intelligent systems that are aware of the context and user needs. In this solution, the selection of appropriate modalities is based on a mechanism of rules selection and the fusion engine is based on mechanism of fusion rules. This architecture is implemented by defining a set of concepts, object properties and instances inside ontology, taking into account special environmental and user contexts associated with a modality, events and preconditions associated to fusion rules. We defined a fusion algorithm that take into account the order of events and the temporal aspect for each of them. The creation of the ontology made the description of the environment, the choice of modalities easier by declaring all constraints and relations between different concepts using semantic relations, in the other hand, DL queries are very useful in the selection of modalities and the recuperation of instances for preconditions needed for fusion rules. The semantic rules are described using SWRL (semantic web rule language). The ontology and the description languages are based on W3C standards, which make it formal and logically verified.

In this paper we adopted an approach that addresses the weaknesses of previous designs in multimodal systems. This is done through environment description that contains all possible modalities, objects and scenarios. The adoption of this architecture will facilitate the work of a fusion engine by giving it the most meaningful combinations of events.

ACKNOWLEDGEMENT

This work has been made possible by the funding awarded by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

1. Yuen, P.C., Y.Y. Tang, and P.S.-p. Wang, *Multimodal Interface for Human-Machine Communication*. Machine Perception Artificial

Intelligence. Vol. 48. 2002, Singapore: World Scientific Publishing Co., Pte. Ltd.

2. Oviatt, S.L. and P.R. Cohen, *Multimodal Interfaces that Process What Comes Naturally.* Communications of the ACM, 2000. 43(3): p. 45 - 53.

3. Shin, B.-S., H. Ahn, and G.Y. Kim, *Wearable multimodal interface for helping visually handicapped persons*, in *16th international conference on artificial reality and telexistence*. 2006, LNCS vol. 4282: Hangzhou, China. p. 989-988.

4. Raisamo, R., et al., *Testing usability of multimodal applications with visually impaired children.* IEE, Institute of Electrical and Electronics Engineers Computer Society, 2006. 13(3): p. 70-76.

5. Lai, J., S. Mitchell, and C. Pavlovski, *Examining modality usage in a conversational multimodal application for mobile e-mail access.* International Journal of Speech Technology, 2007. 10(1): p. 17-30.

6. Debevc, M., et al., *Accessible multimodal Web pages with sign language translations for deaf and hard of hearing users*, in *DEXA 2009, 20th International Workshop on Database and Expert Systems Application*. 2009, IEEE: Linz, Austria. p. 279-283.

7. Bolt., *"Put-that-there": Voice and gesture at the graphics interface ACMSIGGRAPH Computer Graphics, v.14 n.3, p.262-270, July 1980*

8. Holzapfel, H., Nickel, K., and Stiefelhagen, R. I., *mplementation and evaluation of a constraint-based multimodal fusion system for speech and 3D pointing gestures. In Proceedings of the 6th international Conference on Multimodal interfaces (State College, PA, USA. ICMI '04. ACM, New York, NY, 175-182.* 2004.

9. Duarte, C.a.C., L. , *A conceptual framework for developing adaptive multimodal applications. In Proceedings of the 11th international Conference on intelligent User interfaces (Sydney, Australia, January 29 - February 01, 2006). IUI '06. ACM, New York, NY, 132-139.* 2006.

10. Latoschik, M.E., *Designing transition networks for multimodal VR-interactions using a markup language. Multimodal Interfaces, 2002. In Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces. IEEE, 411-416.* 2002.

11. Schlomer, T., Poppinga B., Henze N. & Boll S. , *Gesture Recognition with a Wii Controller, Proceedings of the 2nd international conference on Tangible and embedded interaction, ACM, 2008.* 2008.

12. Jobs, S.P.e.a., *Touch Screen Device, Method, and Graphical User Interface for Determining Commands by Applying Heuristics. United States Patent Application 20080122796. Kind Code A, May 29, 2008.* 2008.

13. Schroeter, J., et al. *Multimodal Speech Synthesis*. 2000. New York, NY.

14. Engel, R.P., Norbert. , *Modality Fusion.SmartKom: Foundations of Multimodal Dialogue Systems. Springer, Berlin.* 2006.

15. Norbert Reithinger, J.A., Tilman Becker, Anselm Blocher, Ralf Engel, Markus Löckelt, Jochen Müller, Norbert Pfleger, Peter Poller, Michael Streit, Valentin Tschernomas, *SmartKom - Adaptive and Flexible Multimodal Access to Multiple Applications, DFKI GmbH – German Research Center for Artificial Intelligence*2003.

16. Daniel Sonntag, M.R., *A Multimodal Result Ontology for Integrated Semantic Web Dialogue Applications Daniel Sonntag_, Massimo Romanelli DFKI GmbH . German Research Center for Arti_cial Intelligence Stuhlsatzenhausweg 3, d-66123* 2006.

17. www.ontologyportal.org.

18. www.loa-cnr.it/DOLCE.html.

19. Dumas, B., D. Lalanne, and R. Ingold, *Description languages for multimodal interaction: a set of guidelines and its illustration with SMUIML.* Journal on Multimodal User Interfaces, 2010. 3(3): p. 237-247.

20. Little, T.D.C., Ghafoor, A., Chen, C.Y.R., Chang, C.S., Berra, P.B. , *Multimedia Synchronization.* IEEE Data Engineering Bulletin, 1991. 14: p. 26-35.

21. Henry, T.R., Hudson, S.E., Newell, G.L.. , *Integrating Gesture and Snapping into a User Interface Toolkit, in Proc*, in *Symposium on User Interface Software and Technology* 1990, ACM Press. p. 112-121.

22. Nigay, L., Coutaz, J., Salber, D. MATIS, *A multimodal airline travel information system*, E.B.A. SM/WP10, Editor. 1993.

23. http://tt.stanford.edu/.

24. http://www.w3.org/Submission/SWRL/.

25. http://www.w3.org/TR/owl-guide/.