# INTERACTION CONTEXT-AWARE MODALITIES AND MULTIMODAL FUSION FOR ACCESSING WEB SERVICES

**Atef Zaguia[1], Manolo Dulva Hina[1,2], Chakib Tadj[1], Amar Ramdane-Cherif[2,3]**
[1]LATIS Laboratory, Université du Québec, École de technologie supérieure
1100, rue Notre-Dame Ouest, Montréal, Québec, H3C 1K3 Canada
{atef.zaguia.1@ens.etsmtl.ca, manolo-dulva.hina.1@ens.etsmtl.ca, ctadj@ele.etsmtl.ca}
[2]PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines
45, avenue des États-Unis, 78035 Versailles Cedex, France
[3]LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines
45, avenue des États-Unis, 78035 Versailles Cedex, France
rca@prism.uvsq.fr

**ABSTRACT**

These days, a human-controlled multimodal system equipped with multimodal interfaces is possible, allowing for a more natural and more efficient interaction between man and machine. In such a system, users can take advantage of the modalities to communicate or exchange information with applications. The use of multimodal applications, integrated with natural modalities, is an effective solution for users who would like to access ubiquitous applications such as web services. The novelty of this work is that all modalities that are made available to the user to access web services are already found to be suitable to the user's current situation. By suitablity, we mean these are optimal modalities – found to be suitable to the user's interaction context (i.e. the combined context of the user, his environment and his computing system) and media devices are available to support them. These modalities can be invoked for the data input/output by the user to access web service using a semantic combination of modalities, called "multimodal fusion". While current state-of-the-art uses two (on rare cases, three) predefined modalities, our approach allows an unlimited number of concurrent modalities. This approach gives user some flexibility to use the modalities that he sees fit for his situation and comfortable with it. The description of the detection of optimal modality as well as the fusion process, together with sample application are presented in this paper.

**Keywords:** multimodal fusion, web service, interaction context, multimodality

## 1 INTRODUCTION

As always, one of the biggest challenges in informatics has always been the creation of systems that allow transparent and flexible human-machine interaction [1, 2]. Researchers always aim to satisfy the needs of users and come up with systems that are intelligent, more natural and easier to use. Various efforts were directed towards the creation of systems that facilitate communication between man and machine [3] and allow a user to use his media devices invoking natural modalities (eye gaze, speech, gesture, etc.) in communicating or exchanging information with applications. These systems receive inputs from sensors or gadgets (e.g. camera, microphone, etc.) and make an interpretation and comprehension out of these inputs; this is multimodality [4-7]. A well-known sample of these systems is that of Bolt's "Put that there" [8] where he used gesture and speech to move objects.

In our days, various multimodal applications [3, 9] have been developed and are found to be effective solutions for users who cannot use a keyboard or a mouse [10], on users who have visual handicap [11], on mobile users equipped with wireless telephone/mobile devices [12], on weakened users [13], etc. The common weakness of these systems is they already have predefined the modalities that are associated with them and that the flexibility to use another modality other than those that have been defined is not existent. The modalities in these applications are therefore not adapted to the realities of the user's actual situation. Furthermore, given that the context of the user evolves as he undertakes a computing task, then any fixed modality that is assigned to the application is not adaptive to the evolution of the user's situation.

In this regard, we propose a work in which the modalities invoked by an application are not predefined. Furthermore, we take into account the

user's context – actually a much bigger context called "*interaction context*" – in determining which modalities are suitable to the user's situation. An interaction context is the collective context of the user, of his environment and his computing system from the time he starts undertaking a computing task up to its completion. In this paper, we consider the parameters that are important to the user in undertaking web services, and the constraints each of these parameters impose on the suitability of a modality. In a ubiquitous computing environment [14], context-awareness [15] and the system's adaptation to it are basic requirements.

Our work is a contribution on context-based modality activation for web services. The novelty of this approach is that we are sure that the modalities that are invoked in our work are indeed suited to the user's actual situation. Web services can be made accessible from another application (a client, a server or another web services) within the Internet network using the available transport protocols [16]. This application service can be implemented as an autonomous application or a set of applications.

The aim of this research work is to develop a flexible system based on application, capable of manipulating more than two modalities. The approach consists of modules that detect suitable modalities, take into account each modality's parameters and perform the fusion of the modalities in order to obtain the corresponding action to be undertaken within the application. This approach is more flexible than current state-of-the art systems that run on predefined two modalities.

The rest of this paper is organized as follows. Section II takes note of related research works. Section III discusses the modalities and media devices, section IV is about finding the appropriate modalities to a given interaction context, sections V and VI are about the discussion on multimodal fusion and the system's components, and sample application. The paper is concluded in section VII.

## 2 RELATED WORK

*Modality* refers to the mode of interaction for data input and output between a user and a machine. In an impoverished, traditional computing set-up, the human-machine interaction is limited to the use of mouse, keyboard and screen. Hence, multimodality is a solution that enriches the communication bandwidth between man and machine. Some media devices supporting modalities include 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 and is beneficial to users with temporary or permanent handicap, allowing them to benefit from the advancement in technology in undertaking computing tasks. In multimodality, other modes of interaction are invoked whenever some modalities are found to be not available or not possible to use. For example, speech [17] is a more effective input modality than a mouse or a keyboard for a mobile user. Using multimodality in accessing user application is an effective way of accomplishing user's computing task.

Current research works demonstrate that the modalities that are invoked for use are those that are suitable to the user's interaction context. Some of those works involve [18, 19] and [20]. Context [21] is a subjective issue, based on different definitions and implications each researcher associates to the term. Some related research work on context include [22], [23] and [24]. The rationale on using interaction context, rather than plain context, is we would like to come up with a more inclusive notion of context by considering not only the context of the user but also of his environment and his computing system, hence the notion of interaction context.

Web service [25] is a software component that represents an application function (or application service). It is a technology that allows applications to interact remotely via Internet, independent of platforms and languages on which they are based. The service can be accessed from another application (a client, server or another Web service) through Internet using transport protocols. Web services are based on a set of standardizing protocols, namely: the transport layer, the XML messages, the description of services and the search service.

Some works in accessing web services using various modalities include the work of [26] which presents an effective web-based multimodal system that can be used in case of disasters, such as earthquake. The work of [27] demonstrates the concepts of discovery and invocation of services. Here, a user (i.e. a passenger) can use his cellular phone to know the available services in the airport, and using voice and touch, the user can browse and select desired services. In [28], the author presents a system commonly used in house construction, such as a bathroom design. The multimodal system interface spontaneously integrates speech and stylus inputs. The output comes in the form of voice, graphic or facial expressions of a talking head displayed on screen. The work in [29] presents a case of a human-robot multimodal interaction. Here, the two-armed robot receives vocal and non-verbal orders to make or remove objects. The use of such robots with remote control can be very beneficial especially in places where access is dangerous for human beings. In [30], the authors proposed a multimodal system that helps children learn the Chinese language through stylus and voice.

The above-mentioned multimodal systems are important and make tasks earlier for humans. However, the very fact that they are based on only

two (and on rare occasion, three) modalities provides constraints on the part of the users. This leads us to the conceptualization of a multimodal system with an unlimited number of modalities, providing easier interface access and simpler usage for the users.

## 3 MODALITY AND ITS MULTIMEDIA SYSTEM REQUIREMENTS

As stated, *modality*, in this work, refers to the logical structure of human-machine interaction, specifically the mode on how data is entered and presented as output between a user and computer. Using natural language processing as categorization basis, we classify modalities into 8 different groups:

1. **Tactile Input ($T_{in}$)** – the user uses the sense of touch to input data.
2. **Vocal Input ($VO_{in}$)** – voice or sound is captured and becomes input data.
3. **Manual Input ($M_{in}$)** – data entry is done using hand manipulation or stroke.
4. **Visual Input ($VI_{in}$)** – movement of human eyes are interpreted and considered as data input.
5. **Gestural Input ($G_{in}$)** – human gesture is captured and considered as data input.
6. **Vocal Output ($VO_{out}$)** – sound is produced as data output; the user obtains the output by listening to it.
7. **Manual Output ($M_{out}$)** – the data output is presented in such a way that the user would use his hands to grasp the meaning of the presented output. This modality is commonly used in interaction with visually-impaired users.
8. **Visual Output ($VI_{out}$)** – data are produced and presented in a way that the user read them.

To realize multimodality, there should be at least one modality for data input and at least one modality for data output that can be implemented. In this work, we define multimedia as electronic media devices used to store and experience multimedia content (i.e. text, audio, images, animation, video, interactivity context forms). Not being an exhaustive list, we list below some electronic media devices that support modalities:

1. **Tactile Input Media (TIM)** – touch screen.
2. **Vocal Input Media (VOIM)** – microphone and speech recognition system.
3. **Manual Input Media (MIM)** – keyboard, mouse, stylus, Braille.
4. **Visual Input Media (VIIM)** – eye gaze.
5. **Gestural Input Media (GIM)** – electronic gloves.
6. **Vocal Output Media (VOOM)**– speaker, headset, speech synthesis system.
7. **Manual Output Media (MOM)** – Braille, overlay keyboard.
8. **Visual Output (VIOM)** – screen, printer, projector.

Clearly, there is a relationship that exists between the modality and media devices. To represent this relationship, let there be a function $\mathbf{g_1}$ that maps a modality to a media group, given by $\mathbf{g_1}$**: Modality → Media Group**. The elements of function $g_1$ are given beow:

$\mathbf{g_1}$ = {($T_{in}$, TIM), ($VO_{in}$, VOIM), ($M_{in}$, MIM),($VI_{in}$, VIIM), ($G_{in}$, GIM), $VO_{out}$, VOOM), ($M_{out}$, MOM), ($VI_{out}$, VIOM)}

Given a modality set $\mathbf{M = \{T_{in}, VO_{in}, M_{in}, VI_{in}, G_{in}, VO_{out}, M_{out}, VI_{out}\}}$ then modality is possible under the following condition:

$$Modality\ Possible = (T_{in} \vee VO_{in} \vee M_{in} \vee VI_{in} \vee G_{in}) \wedge (VO_{out} \vee M_{out} \vee VI_{out}) \quad (1)$$

Hence, failure of modality can be specified by the following relationship:

$$Modality\ Failure = ((T_{in} = Failed) \wedge (VO_{in} = Failed) \wedge (M_{in} = Failed) \wedge (VI_{in} = Failed) \wedge (G_{in} = Failed)) \vee ((VO_{out} = Failed) \wedge (M_{out} = Failed) \wedge (VI_{out} = Failed)) \quad (2)$$

where the symbols $\wedge$ and $\vee$ denote logical AND and OR, respectively.

Given the non-exhaustive media devices listed above, it is possible to denote each modality in terms of its supporting media devices, as given below:

$$T_{in} = touch\ screen \quad (3)$$

$$VO_{in} = (Microphone \wedge Speech\ recognition) \quad (4)$$

$$M_{in} = ((Keyboard \vee (Mouse \wedge stylus)) \wedge Braille \quad (5)$$

$$VI_{in} = eye\ gaze \quad (6)$$

$$G_{in} = electronic\ gloves \quad (7)$$

$$VO_{out} = ((Speech\ synthesis \vee (Speaker \wedge Headset)) \quad (8)$$

$$M_{out} = Braille\ Terminal \vee Overlay\ Keyboard \quad (9)$$

$$VI_{out} = screen \vee printer \vee projector \quad (10)$$

Here, our proposed system detects all media

devices available to the user and accordingly produces result indicating the appropriate modalities.

## 4 FINDING APPROPRIATE MODALITIES TO A GIVEN INTERACTION CONTEXT

Let interaction context, $\mathbf{IC} = \{\mathbf{IC}_1, \mathbf{IC}_2, \ldots, \mathbf{IC}_{max}\}$, be a set of all parameters that describe the status of the user, his environment and his computing system as he undertakes a computing task. At any given time, a user has a specific interaction context $i$ denoted $\mathbf{IC}_i$, $1 \leq i \leq$ max. Formally, an interaction context is a tuple composed of a specific user context (**UC**), environment context (**EC**) and system context (**SC**). An instance of **IC** may be written as:

$$IC_i = UC_k \otimes EC_l \otimes SC_m \tag{11}$$

where $1 \leq k \leq max_k$, $1 \leq l \leq max_l$, and $1 \leq m \leq max_m$, and $max_k$ = maximum number of possible user context, $max_l$ = maximum number of possible environment context, and $max_m$ = maximum number of possible system context. The Cartesian product (symbol: $\otimes$ ) means that at any given time, **IC** yields a specific combination of **UC**, **EC** and **SC**.

The user context **UC** is made up of parameters that describe the state of the user during the conduct of his activity. Any specific user context $k$ is given by:

$$UC_k = \bigotimes_{x=1}^{max_k} ICParam_{kx} \tag{12}$$

where $\mathbf{ICParam}_{kv}$ = parameter of $\mathbf{UC}_k$ where $k$ is the number of **UC** parameters. Similarly, any environment context $\mathbf{EC}_l$ and system context $\mathbf{SC}_m$ are given as follows:

$$EC_l = \bigotimes_{y=1}^{max_l} ICParam_{ly} \tag{13}$$

$$SC_m = \bigotimes_{z=1}^{max_m} ICParam_{mz} \tag{14}$$

For our intended application – web services – we take into account the IC parameters that factors in whether a modality is suitable or not. The following is a summary of these factors:

(a) **User Context**:

1. **User handicap** – it affects the user's capacity to use a particular modality. We note four handicaps, namely (1) manual handicap, (2) muteness, (3) deafness, and (4) visual impairment. See Table 1.
2. **User location** – we differentiate between a fixed/stationary location, 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. See Table 2.

(b) **Environmental Context**

1. **Noise level** – the noise definitely affects our ability to use audio as data input or receiving audio data as output. See Table 3.
2. **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. See Table 4.

(c) **System Context**

1. **Computing device** – the capacity of the type of computer we use is a factor that limits which modality we can activate. See Table 5.

**Table 1.** User handicap/profile and its suitability to modalities .

| Modalities | Regular User | Deaf | Mute | Manually Handicapped | Visually Impaired |
|---|---|---|---|---|---|
| Tactile Input ($T_{in}$) | √ | √ | √ | x | √ |
| Vocal Input ($VO_{in}$) | √ | √ | x | √ | √ |
| Manual Input ($M_{in}$) | √ | √ | √ | x | √ |
| Visual Input ($VI_{in}$) | √ | √ | √ | √ | x |
| Gestural Input ($G_{in}$) | √ | √ | √ | √ | √ |
| Vocal Output ($VO_{out}$) | √ | x | √ | √ | √ |
| Manual Output ($M_{out}$) | √ | √ | √ | x | √ |
| Visual Output ($VI_{out}$) | √ | √ | √ | √ | x |

(Note: symbols √ and × are used to denote suitability and non-suitability, respectively)

**Table 2.** User location and its suitability to modalities.

| Modalities | At Home | At Work | On the go |
|---|---|---|---|
| Tactile Input ($T_{in}$) | √ | √ | × |
| Vocal Input ($VO_{in}$) | √ | √ | × |
| Manual Input ($M_{in}$) | √ | √ | √ |
| Visual Input ($VI_{in}$) | √ | √ | × |
| Gestural Input ($G_{in}$) | √ | √ | √ |
| Vocal Output ($VO_{out}$) | √ | × | √ |
| Manual Output ($M_{out}$) | √ | √ | × |
| Visual Output ($VI_{out}$) | √ | √ | √ |

**Table 3.** Noise level and its suitability to modalities.

| Modalities | Quiet | Noisy |
|---|---|---|
| Tactile Input ($T_{in}$) | √ | √ |
| Vocal Input ($VO_{in}$) | √ | × |
| Manual Input ($M_{in}$) | √ | √ |
| Visual Input ($VI_{in}$) | √ | √ |
| Gestural Input ($G_{in}$) | √ | √ |
| Vocal Output ($VO_{out}$) | √ | × |
| Manual Output ($M_{out}$) | √ | √ |
| Visual Output ($VI_{out}$) | √ | √ |

**Table 4.** Brightness or darkness of the workplace and its effect on selection of appropriate modalities.

| Modalities | Workplace Bright | Workplace Dark | Workplace Very Dark |
|---|---|---|---|
| Tactile Input ($T_{in}$) | √ | √ | x |
| Vocal Input ($VO_{in}$) | √ | √ | √ |
| Manual Input ($M_{in}$) | √ | √ | √ |
| Visual Input ($VI_{in}$) | √ | x | x |
| Gestural Input ($G_{in}$) | √ | x | x |
| Vocal Output ($VO_{out}$) | √ | √ | √ |
| Manual Output ($M_{out}$) | √ | x | x |
| Visual Output ($VI_{out}$) | √ | √ | √ |

**Table 5.** The type of computing device and how it affects the selection of appropriate modalities.

| Modalities | PC/Laptop | Ipad | Cellphone/PDA |
|---|---|---|---|
| Tactile Input ($T_{in}$) | √ | √ | x |
| Vocal Input ($VO_{in}$) | √ | √ | √ |
| Manual Input ($M_{in}$) | √ | √ | √ |
| Visual Input ($VI_{in}$) | √ | x | x |
| Gestural Input ($G_{in}$) | √ | x | x |
| Vocal Output ($VO_{out}$) | √ | √ | √ |
| Manual Output ($M_{out}$) | √ | x | x |
| Visual Output ($VI_{out}$) | √ | √ | √ |

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, the environmental context and the system context. The suitability of a specific modality is shown by a series of relationships given below:

$$T_{in} = (user \neq manually\ handicapped) \wedge (location \neq on\ the\ go) \wedge (workplace \neq very\ dark\ ) \wedge (computer \neq cellphone\ /\ PDA\ ) \quad (15)$$

$$VO_{in} = (user \neq mute) \wedge (location \neq on\ the\ go) \wedge (noise\ level \neq noisy) \quad (16)$$

$$M_{in} = (user \neq manually\ handicapped) \wedge (workplace \neq dark \vee workplace \neq very\ dark) \quad (17)$$

$$VI_{in} = (user \neq visually\ impaired) \wedge location \neq on\ the\ go) \wedge (computer \neq Cellphone/PDA \vee computer \neq iPad) \quad (18)$$

$$G_{in} = (computer \neq iPad \vee computer \neq cellphone/PDA) \quad (19)$$

$$VO_{out} = (user \neq deaf) \wedge (location \neq at\ work) \quad (20)$$

$$M_{out} = (user \neq manually\ handicapped) \wedge (location \neq on\ the\ go) \wedge (computer \neq cellphone/PDA \vee computer \neq iPad) \quad (21)$$

$$VI_{out} = (user \neq visually\ impaired) \wedge (workplace \neq dark \vee (workplace \neq very\ dark) \quad (22)$$

In our work, the proposed system detects the values of related interaction context parameters and accordingly produces result indicating appropriate modalities.

Finally, the **optimal modality** that will be selected by the system is that modality that is found in the intersection of (1) appropriate modalities based on available media devices, and (2) appropriate modalities based on the given interaction context. For example, for a tactile input modality to be selected as an optimal modality Equation (3) and Equation (15) must hold, otherwise such modality is not appropriate for use and implementation. The same concept holds true for all other remaining modalities.

A particular modality is said to be **optimally chosen** if it satisfies both requirements stated above. Hence, the optimality of each modality for our target application (web services) is given below:

$$T_{in} = (available\ media = touch\ screen) \wedge (user \neq manually\ handicapped) \wedge (location \neq on\ the\ go) \wedge (workplace \neq very\ dark\ ) \wedge (computer \neq cellphone/\ PDA\ ) \quad (23)$$

$$VO_{in} = (available\ media = Microphone \wedge Speech\ recognition) \wedge (user \neq mute) \wedge (location \neq on\ the\ go) \wedge (noise\ level \neq noisy) \quad (24)$$

$$M_{in} = (\text{available media} = ((\text{Keyboard} \vee (\text{Mouse} \wedge \text{stylus})) \wedge \text{Braille}) \wedge (\text{user} \neq \text{manually handicapped}) \wedge (\text{workplace} \neq \text{dark} \vee \text{workplace} \neq \text{very dark}) \quad (25)$$

$$VI_{in} = (\text{available media} = \text{eye gaze}) \wedge (\text{user} \neq \text{visually impaired}) \wedge (\text{location} \neq \text{on the go}) \wedge (\text{computer} \neq \text{Cellphone/PDA} \vee \text{computer} \neq \text{iPad}) \quad (26)$$

$$G_{in} = (\text{available media} = \text{electronic gloves}) \wedge (\text{computer} \neq \text{iPad} \vee \text{computer} \neq \text{cellphone/PDA}) \quad (27)$$

$$VO_{out} = (\text{available media} = ((\text{Speech synthesis} \vee (\text{Speaker} \wedge \text{Headset}))) \wedge (\text{user} \neq \text{deaf}) \wedge (\text{location} \neq \text{at work}) \quad (28)$$

$$M_{out} = (\text{available media} = \text{Braille Terminal} \vee \text{Overlay Keyboard}) \wedge (\text{user} \neq \text{manually handicapped}) \wedge (\text{location} \neq \text{on the go}) \wedge (\text{computer} \neq \text{cellphone/PDA} \vee \text{computer} \neq \text{iPad}) \quad (30)$$

$$VI_{out} = (\text{available media} = \text{screen} \vee \text{printer} \vee \text{projector}) \wedge (\text{user} \neq \text{visually impaired}) \wedge (\text{workplace} \neq \text{dark} \vee (\text{workplace} \neq \text{very dark}) \quad (31)$$

## 5 AN INTERACTION CONTEXT-AWARE MULTIMODAL FUSION SYSTEM

In this section, we will describe the multimodal fusion system. Here, it is already assumed that all modalities in consideration are already taken as the optimal choices for the given user's situation.

### 5.1 Architectural Framework

Accessing a web service involves the use of four web-service modules. These modules need to be loaded on a computer, on a robot, or any machine that can communicate via Internet or social network. The architectural framework of our proposed system is shown in Figure 1.

As shown in the diagram, the multimodal fusion system consists of the following elements:

- *Context Information Agent* – this component detects the current instance of user's interaction context. See Figure 2 for further details.
- *Parser* – it takes an XML file as input, extracts information from it and yields an output indicating the concerned modality and its associated parameters
- *Parameter Extractor* – the output from the parser serves as input to this module, it then extracts the parameters of each involved modality
- *Fusion and Multimodality* – based on the given interaction context, this component selects the optimal modality and the parameters involved in each selected modality as well as the time in consideration; it decides if fusion is possible.
- *Internet/Social Network* – serves as the network by which the user and the concerned machine/computer communicate.
- *Computing Machine/Robot/Telephone* – this is the entity with which the user communicates.

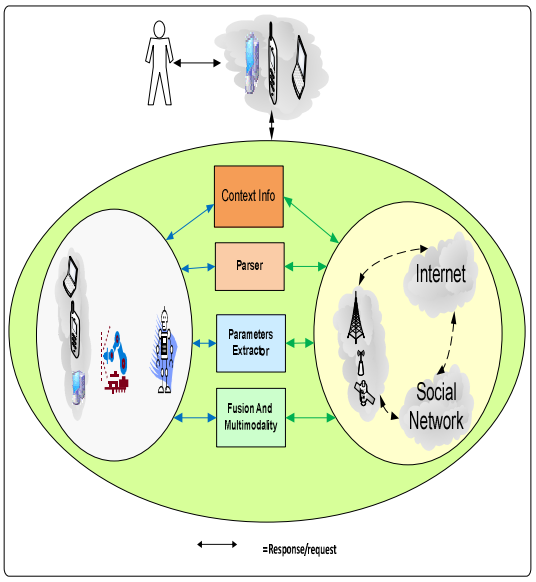


**Figure 1:** Architecture of multimodal fusion system for accessing web services

As stated, the **Context Information Agent** (see Figure 2) detects the current instance of user's interaction context. The values of the environmental context parameters are sensed using sensors and interpreted accordingly. The user's context is based upon the user profile as well as the user's location which is detected through the use of a sensor (i.e. GPS). The system context is detected using the computing device that the user is currently using as well as necessary computing resources parameters such as the current available bandwidth, the network by which the computer is connected, the computer's available memory, battery and processor and its activities.

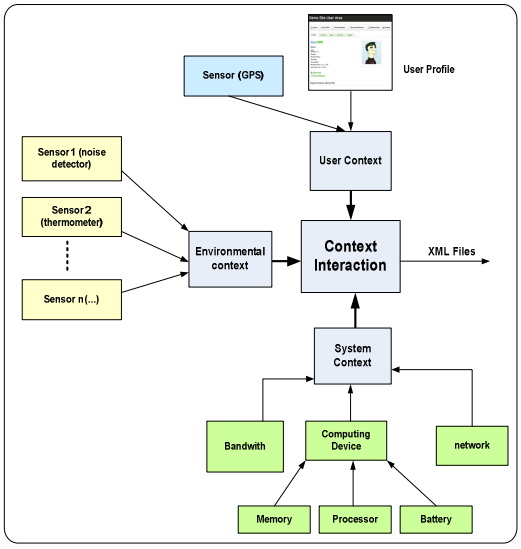


**Figure 2:** The parameters taken into account by the Context Information Agent.

### 5.2 Multimodal Fusion

*Fusion* [28, 31, 32] is a logical combination of two or more entities, which in this work refers to two or more modalities. Modality signals are intercepted by the fusion agent and then combine them based on some given semantic rules.

As per literature review, two sets of fusion schemes exist: the *early fusion* and the *late fusion* [33]. *Early fusion* [34] refers to a fusion scheme that integrates unimodal features before learning concept. The fusion takes effect on signal level or within the actual time that an action is detected [35]. On the other hand, *late fusion* [36] is a scheme that first reduces unimodal features to separately learned concept scores, and then these scores are integrated to the learned concepts. The fusion is effected on semantic level. In this work, the fusion process used is the *late fusion*.

The processes involved in the multimodal fusion are shown in Figure 3. Two or more modalities may be invoked by the user in this undertaking. Consider for example the command "*Replace this file with that file*" wherein the user uses speech and a mouse click to denote "*this file*" and another mouse click to denote "*that file*". In this case, the modalities involved are: *input modality 1 = speech* and media supporting *input modality 2 = mouse*. The processes involved in the fusion of these modalities are as follows:

- *Context Information* – Detects interaction context using available sensors and gadgets and user's profile.
- *Recognition* – this component converts the activities involving modalities into their corresponding XML files.
- *Parser Module*, *Parameter Extraction Module* and *Multimodal and Fusion Module* – The parser takes an XML file as input, extracts information from it and yields an output indicating the concerned modality and its associated parameters. The output from the parser serves as input to the parameter extractor his module which extracts the parameters of each involved modality. And based on the given interaction context, multimodal and fusion module selects the optimal modality and the parameters involved in each selected modality as well as the time in consideration; it decides if fusion is possible.
- *Action* – this involves the corresponding action to be undertaken after the fusion has been made. The resulting output may be implemented using output modality 1, 2, …, *n*. In the same case cited earlier, the media implementing the *output modality involved is the screen.* It is also possible that the confirmation of such action may be presented using a *speaker*.
- *Feedback* – when conflict arises, a user receives a feedback from the system. For example, if "*this file*" and "*that file*" refer to the same entity, the user is informed about it via feedback.

All of these modules need to have been installed in the user's computing device or are situated in any location within the network. The modules themselves communicate with one another in order to exchange information or do a task.

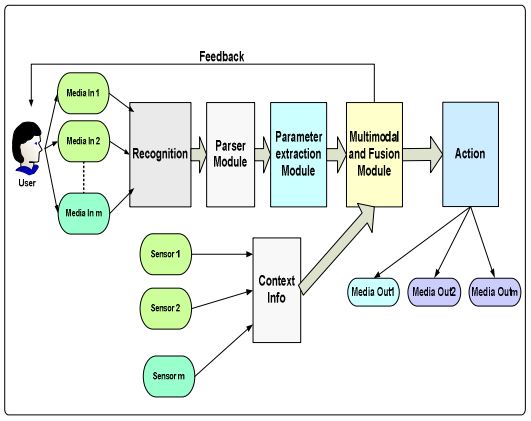


**Figure 3:** Framework of multimodal fusion

Assume for instance the arrival of modality A, along with its parameters (e.g. time, etc.) and another modality B with its own parameters (e.g. time, etc.), then the fusion agent will produce a logical combination of A and B, yielding a result, C. The command/event C is then sent to the application or to the user for implementation. The multimodal fusion can be represented by the relationship f: C = A + B.

In general, the steps involved in the fusion are as follows: (1) determining if a scenario is in the database, (2) for a new scenario, a check of the semantics of the operation to be performed is done, (3) resolution of the conflict (e.g. using speech, user says: "*Write 5*" and using stylus, for example, he

writes "*4*"), (4) feedback to the user to resolve the conflict, (5) storage of the scenario to the database, (6) queries sent to the database, fusion of modalities and storage of the result to the database, and (7) result yields the desired action to be performed using the involved modalities. Further details are available in [37].

The system component tasked to do the fusion process is the fusion agent. The fusion agent itself is composed of three sub-components, namely:

- *Selector* – it interacts with the database in selecting the desired modalities. It retrieves 1 .. *m* modalities at any given time.
- *Grammar* – verifies the grammatical conditions and all the possible interchanges among the modalities involved.
- *Fusion* – this is the module that implements the fusion function.

For diagram and related details, as well as the fusion algorithm, please refer to our previous work in [37] .

Failure in grammatical conditions may also arise. For example, a vocal command "*Put there*" is a failure if there is no other complementary modality action – such as touch, eye gaze, mouse click, etc. – is associated with it. If such case arises, the system looks at some other modalities that come within the same time interval as the previous one that was considered.

## 6 COMPONENTS OF A MULTIMODAL FUSION SYSTEM

Here, we present the different components that are involved in the multimodal fusion process and describe each component's functionality. The formal specification tool Petri Net as well as an actual program in Java are used to demonstrate the sample application and its specification.

### 6.1 The User Interface

Our system has a user interface [9] which allows the users to communicate with the computing system. Here, the user may select modalities that he wishes (note again that all available modalities are already proven suitable to the user's current interaction context). An event concerning the modality is always detected (e.g. *was there a mouse click? was there a vocal input?*, etc.). The system keeps looping until it senses an event involving modality. The system connects to the database and verifies if the event is valid. An invalid event, for example, is a user's selection of two events using two modalities at the same time when the system is expecting only one event execution at a given time. If the event involving modality is valid, an XML file is created, noting the modality and its associated parameters. The XML file is forwarded to the parsing module. The parser then extracts data from the XML tags and sends the result it obtained to the Multimodal and Fusion module.

### 6.2 The Parser and the Parameter Extractor

The parser module receives as input XML files containing data on modalities. From each XML file, this module extracts some tag data that it needs for fusion. Afterwards, it creates a resulting XML file containing the selected modalities and each one's corresponding parameters.

In conformity with W3C standard on XML tags for multimodal applications, we use EMMA notation [38]. EMMA is a generic tagging language for multimodal annotation. The EMMA tags represent the semantically recovered input data (e.g. gesture, speech, etc.) that are meant to be integrated to a multimodal application. EMMA was developed to allow annotation of data generated by heterogeneous input media. When applied on target data, EMMA result yields a collection of multimedia, multimodal and multi-platform information as well as all other information from other heterogeneous systems.

For example, using speech and touch screen modalities, a sample specimen combined XML file is shown in Figure 4.a(Left). The fusion of these two modalities yields the result that is shown in Figure 4.a(Right). The fusion result indicates that the object cube is moved to location (a,b).

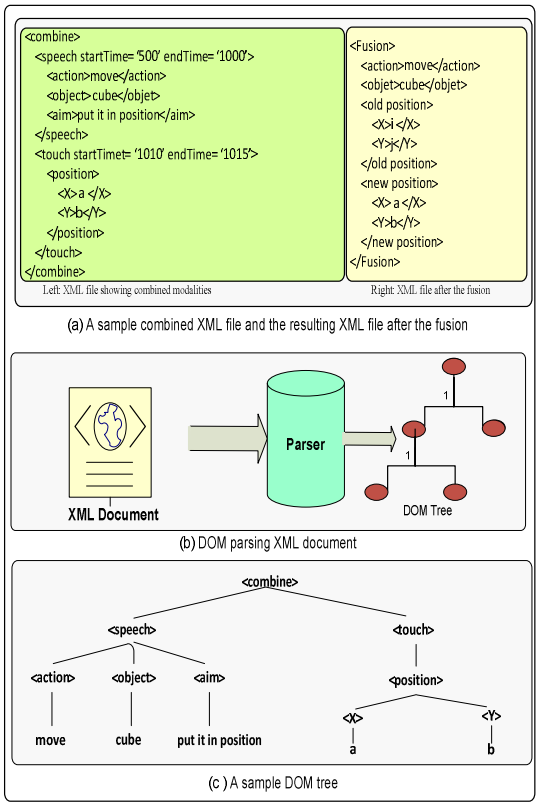


**Figure 4:** The parsing process and the DOM parameter extractor.

The manipulation of an XML file is usually

performed within the development phase of an application, usually undertaken by a parser. A XML parser is a library of functions that can manipulate on an XML document. In selecting a parser, we usually look for two characteristics – that of parser being efficient and rapid. The parser used in this system is called DOM (Document Object Model) [39]. It is a large, complex and stand-alone system that uses object model to support all types of XML documents. When parsing a document, it creates objects containing trees with different tags. These objects contain methods that allow a user to trace the tree or modify its contents. See Figure 4.b.

DOM works in two steps. The first involves the loading of an XML document and the second involves performing different operations on the document. Some advantages of using DOM are: (1) easy traversal of its tree, (2) easy way of modifying the contents of the tree, and (3) traversal of file in whatever direction the user desires. On the other hand, its disadvantages include: (1) consumption of large memory and (2) processing of the document before using it. Using the same example cited earlier, the resulting DOM tree after the parsing process is shown in Figure 4.c.

### 6.3 The Database

The database stores all modalities identified by the system and the modalities' associated parameters. In this work, the database used is PostgresSQL [40]. Using PostgresSQL, the parameters, values and entities of the database are defined dynamically as the module parses the XML file.

As shown in Figure 5, our database consists of eight tables, namely:

- *Context_Info* – this table contains the index of the context parameter, its name and its value as well as the modality this context information is associated with.
- *Modality* – this table contains the names of modalities, the time an action involving the modality begins and the time that it ended.
- *Modality_Added_Parameters* – this table contains all the attributes of every modality.
- *Modality_Main_Parameters* – contains the name of all parameters and their values
- *Union_Modality_Main_Parameters* – this table links the modality and their parameters
- *Fusion* – this table contains all the fusions that had been implemented. This table allows us to keep the previous historical data that can be used later for learning.
- *Fusion_Main_Parameters* – contains the names of parameters and their values that are associated with the multimodal fusion.
- *Union_Fusion_Main_Parameters* – this table serves as a link to the multimodal fusion that was just made, including its corresponding parameters

A sample Context_Info table is shown in Table 6. For all other details of the remaining tables, please refer to [37].

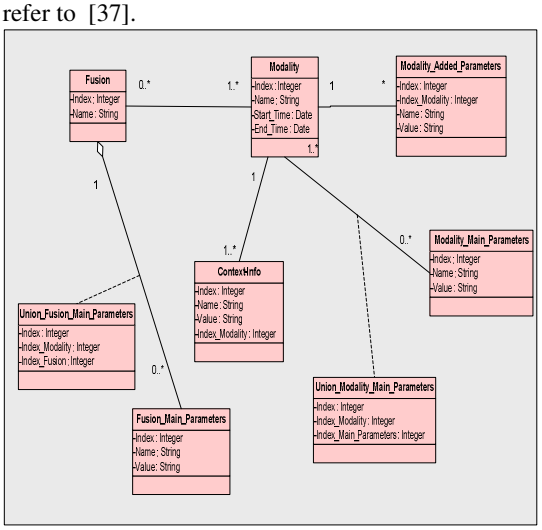


**Figure 5:** Tables that make up the database

**Table 6:** A sample Context_Info table

| Index | Name | value |
|---|---|---|
| 1 | User handicap | Regular user |
| 2 | User location | At home |
| 3 | Computing device | PC |
| 4 | Noise level | quiet |
| 5 | Brightness of workplace | dark |

### 6.4 Sample Case and Simulation using Petri Net

Here, we will demonstrate a sample application and describe its specification/actions using Petri Net. Petri Net [41] is an oriented graph. It is a formal, graphical, executable technique for the specification and analysis of a concurrent, discrete-event dynamic system. It is used in deterministic and in probabilistic variants; a good mean to model concurrent or collaborating systems. Petri Nets allow for different qualitative or quantitative analysis that can be useful in safety validation. *Places* (represented by circles) are states in a simulated diagram whereas *transitions* (represented by rectangles) are processes that are undertaken by a certain element. A certain element goes from one state to another through a transition. Usually a certain element begins in an initial state (manifested via an initial *token* in a place). When an element goes from state "*a*" to state "*b*" through a transition, it is shown in Petri Net via a movement of token from place "*a*" to "*b*" via transition "*x*".

In the specifications that will follow in this paper, only a snapshot of one of many possible outcomes is presented. The application software PIPE2 is used in simulating Petri Net. PIPE2 [42] is an open source, platform independent tool for creating and analysing

Petri nets including Generalised Stochastic Petri nets.

As shown in Figure 6, the sample application is about ticket reservation system. In Figure 6.a, it shows that the menu is composed of four selections – the reservation option, the sign-up option, the manage reservation option and the usage option. For simplicity of the discussion, the usage option, as shown in Figure 6.b, allows the user to select and identify his preferred modalities. In this example, we listed 4 specimen modalities, namely: (1) voice, (2) touch screen, (3) keyboard and (4) eye gaze. When the user signs up for a reservation, the period involved needs to be specified, hence, in Figure 6.c, the interface allows the user to specify the month, the day, and the time for both the departure and the arrival. Finally, in Figure 6.d, we provide an interface which allows the user to input his coordinates as well as the credit card information.

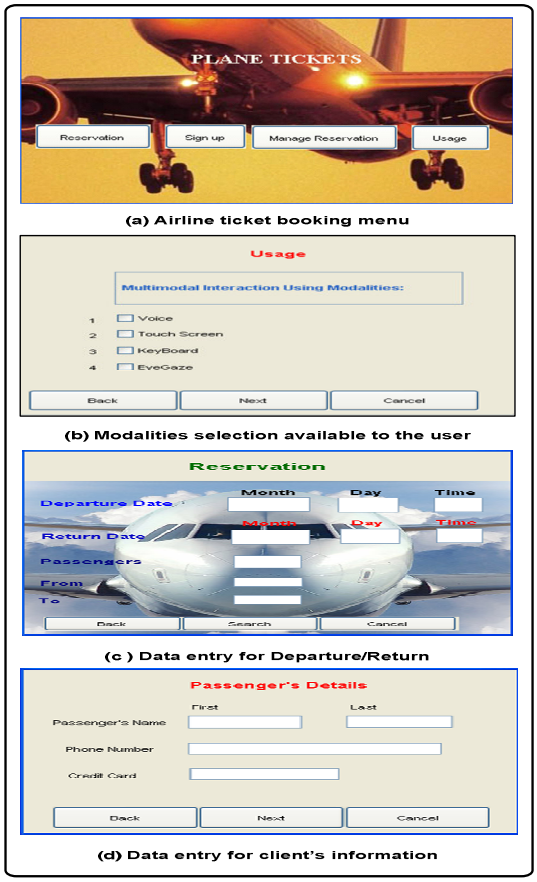


**Figure 6:** (a) Airline reservation system menu, (b) Available modalities, (c) Data entry, departure and return and (d) Data entry for client's information

To make use of the above-mentioned application, assume that a user wishes to make a trip during a vacation. François has decided to take a trip to Paris with the family. One night, he opened his computer equipped with touch screen and connected himself to an airplane tickets website. Using speech, touch screen, eye gaze and keyboard, he was able to book airplane ticket for himself and his family.

During the reservation process, some XML files are created, one for each different modality used. These files are sent to the server of the airplane ticket enterprise within the Internet network using the http protocol. These files are sent first to the "Parser" module for extraction of all involved modalities. Then this module creates another XML file that contains all the different modalities as well as their corresponding parameters. This file is then sent to the "Parameter Extractor" module which will extract all the parameters of the modalities involved and send them to the "Fusion" module.

### 6.4.1 Scenario

François runs the airplane tickets application software. The initial interface is displayed. Using voice, he selected "Reservation". Then the second interface is presented; using touch screen, he chose "Departure Day". Using keyboard, he types "20". Then using eye gaze, he selected "Departure Month" and via speech, he said "May". Then using eye gaze, he selected "Departure Time" and he entered "1:30" using Speech and "pm" using keyboard. Then "Return Day" is selected using speech and he uttered "30". Using keyboard, he selected "Return Month" and types in "6". At the end, using eye gaze, he chose "Return Time" and said "8:00 pm". Then using touch he selected "Passenger" and using speech he said "Three". He selects "departure city" with speech and say "Montreal". Using touch, he selected "Destination" and uttered "Paris". At the end of the process, he received a confirmation message through his laptop computer. This scenario is depicted in the diagram of Figure 7.

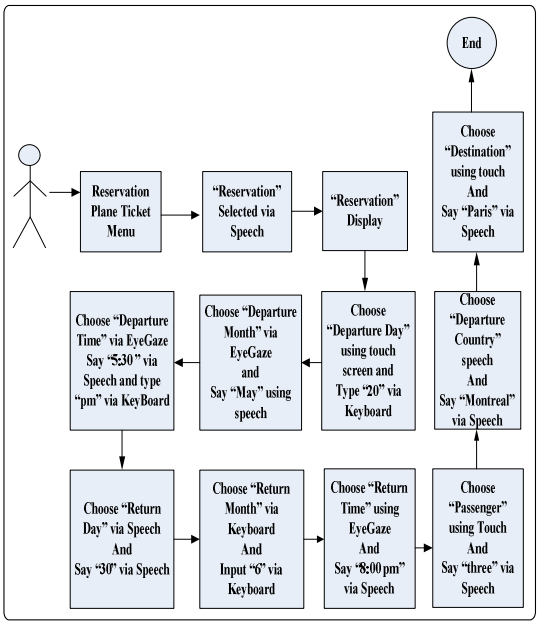


**Figure 7:** A sample scenario showing multimodal interactions between the user and the machine

### 6.4.2 Grammar

The diagram in Figure 8 shows the grammar used for the interfaces A and B of the sample ticket reservation system. The choice, for instance, is defined as a selection of one of the menus (i.e. reserve, sign up, manage reservation and usage) in the user interface. The choice of time is in American time format (example: 12:30 pm); choice of month can be numeric (e.g. 1) or alphabetic (e.g. January). There are two interfaces in the system – the first one allows the user to select a menu (i.e. reservation, usage, sign up and manage) while the second interface allows the user to enter data (day, month and time as well as the number of passengers).

The grammar is used to determine and limit the type of data that is acceptable to the system. Data entry, with respect to the established grammar, can be accomplished using user's preferred modality.

CHOICE reservation → reservation|usage|sign up| manage
ACTION → quit
SELECT → CHOICE|ACTION
INTERFACE A → SELECT

CHOICE_INIT → departure | return
SELECT_POS → day| month | time
CHOICE_DAY → 1 | 2 | … | 31
CHOICE_TIME → 12:00 am | … | 12:00 pm
CHOICE_MONTH_Num → 1 | 2 | … | 12
CHOICE_MONTH_MO → January | February | … | December
CHOICE_MONTH → CHOICE_MONTH_Num | CHOICE_MONTH_MO
CHOICE_PASSENGER_Num → 1 | 2 | … | 10
CHOICE_PASSENGER_MO → one | two | … | ten
CHOICE_PASSENGER → CHOICE_PASSENGER_MO | CHOICE_PASSENGER_Num
INTERFACE_ACTION → cancel | back | search
INTERFACE_B → CHOICE_INIT (SELECT_POS (CHOICE_DAY| CHOICE_MONTH | CHOICE_TIME | CHOICE_PASSENGER)) | INTERFACE_ACTION

**Figure 8:** Grammar for passenger departure and return information

### 6.4.3 Simulation 1

The diagram in Figure 9 shows the interactions involved in interface A in which the user would have to choose one option in a ticket reservation menu. The rest of the diagram demonstrates all activities when and after the user chooses "Reservation" via speech, and further to the "Departure" data entry. Here, an XML file is created which is then sent to the network. The Parser module parses the XML data and extracts tags that contain modality information including its associated parameters. The parameter extractor module extracts the necessary parameters and is then forwarded to the Multimodal and Fusion Module. As it is a unique action, in this example, no fusion is implemented. It is a unimodal action. Nonetheless, it is saved onto the database and the interface B and all menus associated with the "Reservation" option are to be instantiated.

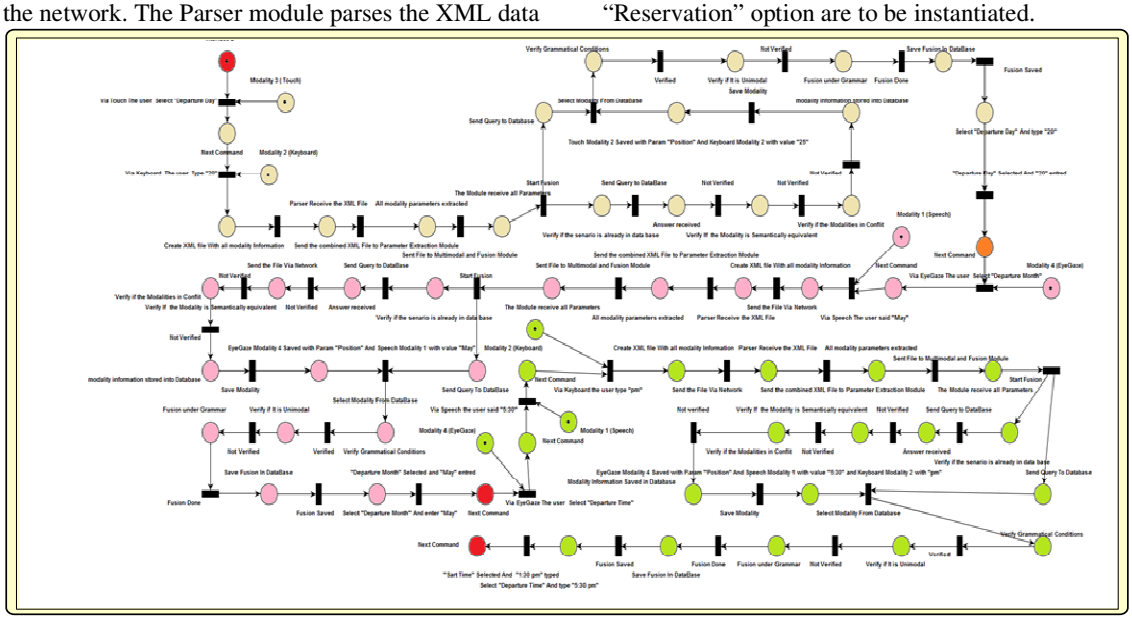

**Figure 9:** System activities as the user ticket reservation (departure) using different modalities

### 6.4.4 Simulation 2

The diagram in Figure 10 demonstrates a Petri net showing the system activities when the "Return" option is selected and is a continuation of Figure 9. At the same time that this option is selected, the three modalities are also selected (see the tokens in keyboard, speech and eye gaze modalities). The Petri Net diagram shows us all the transitions that would arise. Here, our desired output is a data entry for month, day or time which needs to be implemented using only one modality per parameter. For example, month selected by two or more modalities is invalid. In the diagram, a snapshot of one of the many possible outcomes is shown – here, the "return day" option and the day of return are provided using speech, the "return month" and the actual month are provided using keyboard, "return time" option is chosen via eye gaze. We colour the states for easy

viewing – yellow is associated with eye gaze, blue for keyboard modality and green for speech; the red circle denotes "Next command", meaning that the next diagram is a continuation of this diagram.

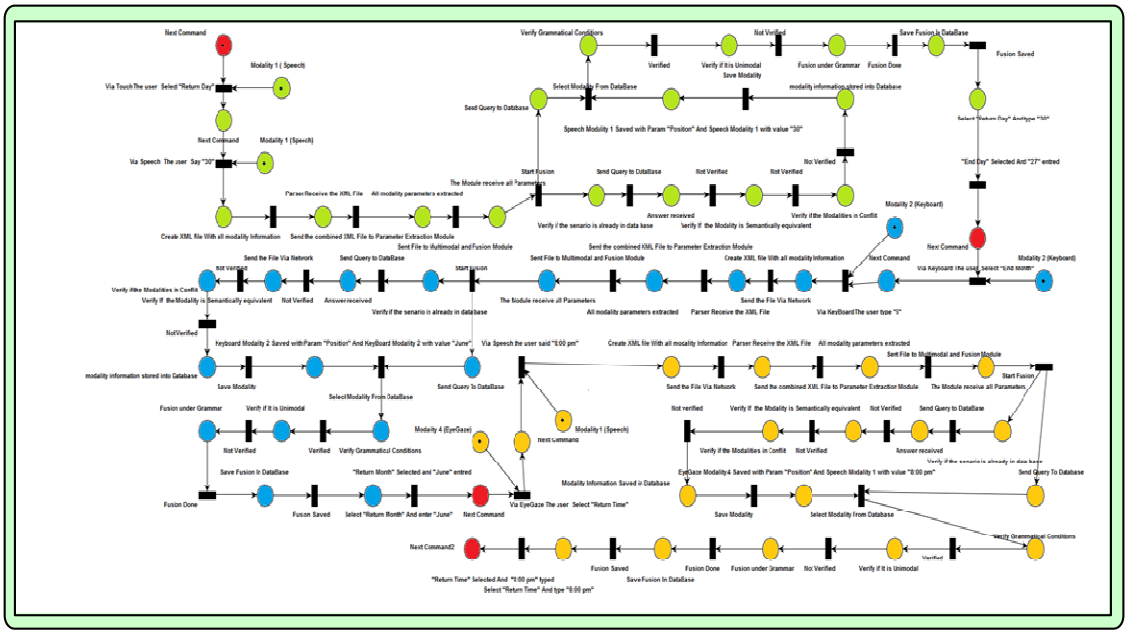

**Figure 10:** System activities as the user ticket reservation (return) using different modalities

### 6.4.5 Simulation 3

The diagram in Figure 11 is a continuation of Figure 10. Again various modalities are invoked for data entry concerning "number of passengers", the client's city of origin and city destination. As is done for each modality involved, the Petri Net shows the serial actions that are to be implemented in the fusion process: an XML file is created for each concerned modality operation, the XML file is sent to parameter extraction module, fusion is started, then query is sent to the database, then the correct modality is selected from the database, then grammar is verified, then fusion is made using the grammar involved and the fusion process is completed. Again, for simplicity purposes, we put colours on the places of the net to distinguish one modality from the others.

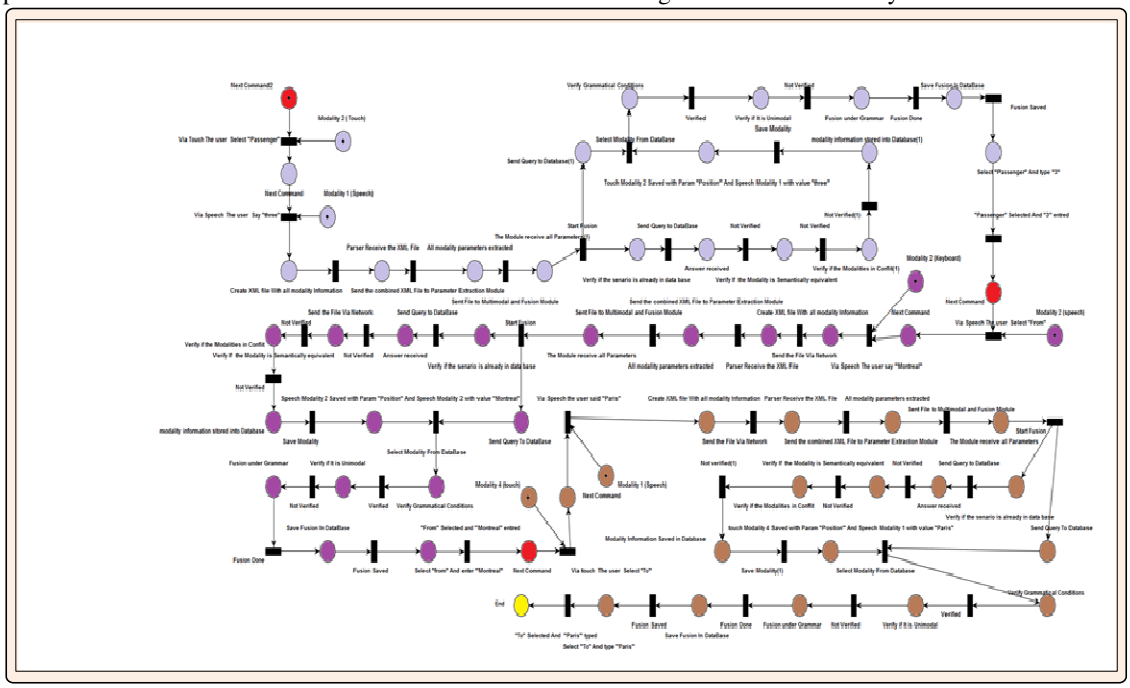

**Figure 11:** System activities during data entry for departure/return and number of passengers

## 7 CONCLUSION

Our review of the state-of-the-art tells us that current system that access web services use multimodalities that are predefined into their system from the very start. Such set-up is correct only on the condition that the fusion is implemented in a controlled environment, one in which the environment parameters remain fixed. In a real-time and real-life set-up, however, this setting is incorrect since too many parameters may change while an action (web service) is being undertaken. In this paper, we present a more flexible approach in which the user chooses the modalities that he sees fit to his situation, therefore, the fusion process is not based on the modalities that are already predefined from the very beginning but from the modalities that are already found suitable to the user's situation as well as being chosen by the user.

We consider the user situation – the user's interaction context (i.e. the combined context of the user, his environment and his computing system) – as well as available media devices in determining whether modalities are indeed apt for the situation. Hence, the modalities that are into consideration for multimodal fusion are already optimal for the user's situation. In this paper, we present our approach on multimodal fusion based on the modalities that the user himself selects. The intended application is to access web services. We showed that an event involving a multimodal action is captured in an XML file clearly identifying the involved modality and its associated parameters. We showed the parsing mechanism as well as the parsing extractor. Then, the fusion of two or more modalities is presented in concept.

The novelties presented in this research work include the selection of optimal modalities based on available media devices as well as the user's interaction context based on intended domain which is accessing web services. Also, the work presented here allows the user to access as much as *n* number of modalities, making access to web services more flexible to the desire and capability of the user.

## ACKNOWLEDGEMENT

We wish to acknowledge the funds provided by the *Natural Sciences and Engineering Council of Canada* (NSERC) which partially support the financial needs in undertaking this research work.

## 8 REFERENCES

[1] Sears, A. and Jacko, J. A., *Handbook for Human Computer Interaction*, 2nd ed.: CRC Press, 2007.

[2] Aim, T., Alfredson, J., et al., "Simulator-based human-machine interaction design," *International Journal of Vehicle Systems Modelling and Testing,* Vol. 4, No. 1/2,pp. 1-16, 2009.

[3] Yuen, P. C., Tang, Y. Y., et al., *Multimodal Interface for Human-Machine Communication* vol. 48. Singapore: World Scientific Publishing Co., Pte. Ltd., 2002.

[4] Ringland, S. P. A. and Scahill, F. J., "Multimodality - The future of the wireless user interface," *BT Technology Journal,* Vol. 21, No. 3,pp. 181-191, 2003.

[5] Ventola, E., Charles, C., et al., *Perspectives on Multimodality*. Amsterdam, the Netherlands: John Benjamins Publishing Co., 2004.

[6] Kress, G., *Multimodality: Exploring Contemporary Methods of Communication.* London, UK: Taylor & Francis Ltd, 2010.

[7] Carnielli, W. and Pizzi, C., *Modalities and Multimodalities* Vol. 12(1). Campinas, Brazil: Springer, 2008.

[8] Bolt, R., "Put that there: Voice and gesture at graphics interface," *Computer Graphics Journal of the association of computing and machinery,* Vol. 14, No. 3,pp. 262-270, 1980.

[9] Oviatt, S. L. and Cohen, P. R., "Multimodal Interfaces that Process What Comes Naturally," *Communications of the ACM,* Vol. 43, No. 3, pp. 45 - 53, 2000.

[10] Shin, B.-S., Ahn, H., et al., "Wearable multimodal interface for helping visually handicapped persons," in *16th international conference on artificial reality and telexistence* Hangzhou, China: LNCS vol. 4282, pp. 989-988, 2006.

[11] Raisamo, R., Hippula, A., et al., "Testing usability of multimodal applications with visually impaired children," *IEE, Institute of Electrical and Electronics Engineers Computer Society,* Vol. 13, No. 3,pp. 70-76, 2006.

[12] Lai, J., Mitchell, S., et al., "Examining modality usage in a conversational multimodal application for mobile e-mail access," *International Journal of Speech Technology,* Vol. 10, No. 1,pp. 17-30, 2007.

[13] Debevc, M., Kosec, P., 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* Linz, Austria: IEEE, pp. 279-283, 2009.

[14] Satyanarayanan, M., "Pervasive Computing: Vision and Challenges," *IEEE Personal Communications,* Vol. 8, No. 4,pp. 10-17, August 2001.

[15] Dey, A. K. and Abowd, G. D., "Towards a Better Understanding of Context and Context-Awareness," in *1st Intl. Conference on Handheld and Ubiquitous Computing*, Karlsruhe, Germany, pp. 304 - 307, 1999.

[16] Li, Y., Liu, Y., et al., "An exploratory study of Web services on the Internet," in *IEEE International Conference on Web Services* Salt Lake City, UT, USA, pp. 380-387, 2007.

[17] Schroeter, J., Ostermann, J., et al., "*Multimodal Speech Synthesis*," New York, NY, 2000, pp. 571-574, 2000.

[18] Hina, M. D., "*A Paradigm of an Interaction Context-Aware Pervasive Multimodal Multimedia Computing System*," Ph.D. Thesis, Montreal, Canada & Versailles, France: Université du Québec, École de technologie supérieure & Université de Versailles-Saint-Quentin-en-Yvelines, 2010.

[19] Awde, A., Hina, M. D., et al., "An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually-Impaired Users," *Journal of Multimedia,* Vol. 4, No. 3, 2009.

[20] Awdé, A., "*Techniques d'interaction multimodales pour l'accès aux mathématiques par des personnes non-voyantes,*" Thèse Ph.D., Département de Génie Électrique Montréal: Université du Québec, École de technologie supérieure, 2009.

[21] Coutaz, J., Crowley, J. L., et al., "Context is key," *Communications of the ACM,* Vol. 48, No. 3,pp. 49-53, March 2005 2005.

[22] Brown, P. J., Bovey, J. D., et al., "Context-Aware Applications: From the Laboratory to the Marketplace," *IEE Personal Communications,* Vol. 4, No. 1,pp. 58 - 64, 1997.

[23] Dey, A. K., "Understanding and Using Context " *Springer Personal and Ubiquitous Computing,* Vol. 5, No. 1,pp. 4 - 7, February 2001.

[24] Henricksen, K. and Indulska, J., "Developing context-aware pervasive computing applications; Models and approach," *Elsevier Pervasive and Mobile Computing,* Vol. 2, pp. 37 - 64, 2006 .

[25] Ballinger, K., *NET Web Services: Architecture and Implementation*. Boston, MA, USA: Addison-Wesley, 2003.

[26] Caschera, M. C., D'Andrea, A., et al., "ME: Multimodal Environment Based on Web Services Architecture " in *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*, Berlin (Heidelberg), pp. 504-512, 2009.

[27] Steele, R., Khankan, K., et al., "Mobile Web Services Discovery and Invocation Through Auto-Generation of Abstract Multimodal Interface," in *ITCC 2005 International conference on Information Technology: Coding and Computing*, Las Vegas, NV, pp. 35-41, 2005.

[28] Pfleger, N., "Context Based Multimodal Fusion," in *ICMI 04* Pennsylvannia, USA: ACM, pp. 265 - 272, 2004.

[29] Giuliani, M. and Knoll, A., "MultiML: A general purpose representation language for multimodal human utterances," in *10th International Conference on Multimodal Interfaces* Crete, Greece: ACM, pp. 165 - 172, 2008.

[30] Wang, D., Zhang, J., et al., "A Multimodal Fusion Framework for Children's Storytelling Systems," in *LNCS* Berlin / Heidelberg: Springer-Verlag, pp. 585-588, 2006.

[31] Pérez, G., Amores, G., et al., "Two strategies for multimodal fusion," in *ICMI'05 Workshop on Multimodal Interaction for the Visualisation and Exploration of Scientific Data* Trento, Italy: ACM, 2005.

[32] Lalanne, D., Nigay, L., et al., " Fusion Engines for Multimodal Input: A Survey," in *ACM International Conference on Multimodal Interfaces,* Beijing, China, pp. 153-160, 2009.

[33] Wöllmer, M., Al-Hames, M., et al., "A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams," *Neurocomputing* Vol. 73, No. 1-3,pp. 366-380, 2009.

[34] Snoek, C. G. M., Worring, M., et al., "Early versus late fusion in semantic video analysis," in *13th annual ACM International Conference on Multimedia* Hilton, Singapore: ACM, 2005.

[35] Oviatt, S., Cohen, P., et al., "Designing the user interface for multimodal speech and pen-based gesture applications: state-of-the-art systems and future research directions," *Human-Computer Interaction,* Vol. 15, No. 4,pp. 263-322, 2000.

[36] Mohan, C. K., Dhananjaya, N., et al., "Video shot segmentation using late fusion technique," in *7th International Conference on Machine Learning and Applications* San Diego, CA, USA: IEEE, pp. 267-270, 2008.

[37] Zaguia, A., Hina, M. D., et al., "Using Multimodal Fusion in Accessing Web Services " *Journal of Emerging Trends in Computing and Information Sciences* Vol. 1, No. 2,pp. 121 - 138, October 2010.

[38] Desmet, C., Balthazor, R., et al., "<emma>: re-forming composition with XML," *Literary & Linguistic Computing,* Vol. 20, No. 1,pp. 25-46, 2005.

[39] Wang, F., Li, J., et al., "A space efficient XML DOM parser," *Data & Knowledge Engineering,* Vol. 60, No. 1,pp. 185-207, 2007.

[40] PostgreSQL, 2010.

[41] ISO/IEC-15909-2, "*Petri Nets*," 2010.

[42] Bonet, P., Llado, C. M., et al., "*PIPE2*," 2010.