Towards a Context-Aware and Pervasive Multimodality*

Manolo Dulva Hina [1,2], Chakib Tadj [1], Amar Ramdane-Cherif [2], and Nicole Levy [2]

[1] Université du Québec, École de technologie supérieure,
1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3 Canada
{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
{rca, nicole.levy}@prism.uvsq.fr

Abstract. Pervasive multimodality aims to realize anytime, anywhere computing using various modes of human-machine interaction, supported by various media devices other than the traditional keyboard-mouse-screen used for data input and output. To achieve utmost efficiency, the modalities for human-machine interaction must be selected based on their suitability to a given interaction context (i.e. combined user, environment and system contexts) and the availability of supporting media devices. To be fault-tolerant, the system should be capable of finding replacement to a failed media device. This paper presents a paradigm of such computing system. Proposed solutions to some technical challenges including the establishment of relationships among interaction context, modalities and media devices, and of the mechanism for the system's incremental acquisition of knowledge on various interaction contexts and their corresponding selections of suitable modalities are presented in this work.

1 Introduction

In the very near future, we shall be living in a society in which pervasive computing (also known as ubiquitous computing) [1, 2] will no longer be a luxury but a way of life. In such a computing system, a user can continue working on a computing task, using various applications, whenever and wherever he wants. The infrastructure of pervasive computing [3] will be available and the applications seamlessly adapting accordingly to the user's context [4] and available resources. Multimodality [5], on its part, promotes the use of different modalities for human interaction (i.e. data entry and data output), the choice of modality being a function of the user's context. Media devices, subject to their availability and suitability to context, may be selected to support then chosen modality. Hence, pervasive multimodality shall be a computing trend of the future, one in which computing adapts to the needs of the users, including those

* This research work has been made possible through the funding provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the scholarship grants from Décanat des études, of Université du Québec, École de technologie supérieure.

with disabilities. To further enhance its functionalities, a pervasive multimodal system may be designed with machine learning [6, 7] capability, a mechanism in which the system "learns" from previous experiences to improve its performance, promotes autonomy and is itself fault-tolerant. The design of such system is filled with technical challenges that need optimal solutions. This paper presents our view and proposed solutions to the challenges to achieve pervasive multimodality. The rest of this paper is structured as follows. Works related to ours are listed in section 2. Section 3 lists down some of the technical challenges and our proposed solutions. The main contents, the matters related to interaction context and multimodality, and the context learning and adaptation are discussed in sections 4 and 5. The paper is concluded in section 6.

2 Related Work

Multimodality advocates the use of various modes for human interaction (data input/output) with a computer beyond the traditional keyboard-mouse-screen input and output devices. A few recent works in this domain include an interface for wireless user interface [5] and the static user interface [8]. Multimodality also refers to the fusion of two (or more) modalities. Sample works in this area include the combined speech and pen inputs [9] and the combined speech and lips movements [10]. Multimodality, as a domain in human-computer interface, aims at providing increased usability to the user, such that a modality that is weak for a given setting may be replaced by another but more appropriate modality. For example, using speech instead of mouse and keyboard as input device/s is more suitable for users doing computing in a moving car. Compared with others, our research goal on pervasive multimodality is to realize anytime, anywhere computing where users will use modalities that are suitable to their given context.

In [4], Coutaz et al explained the importance of context. Research has gone a long way since Dey provided the basic definition of context [11] as applied in context-aware computing [12]. Rey and Coutaz updated the definition in [13] and coined the term "*interaction context*" (IC) to mean the user, the environment and the system's contexts. Our work focuses on the IC in pervasive multimodal computing and considers both static and dynamic context data, including sensed, derived and profiled context information. There has been an active research going on in pervasive and mobile computing. The Prism model in Project Aura [14], for example, demonstrates a user's moving aura (i.e. user's profile and task). Our work has extended this concept by considering an incremental learning system in which acquired knowledge becomes part of pervasive information, along with user's profile, task and preferences. We inject the features of adaptability and autonomy into the system via machine learning.

3 Technical Challenges

Here, we list down some software engineering challenges of pervasive multimodality via technical requirements that need to be addressed and also describe our approach.

Our goal is to model a pervasive computing system that senses its current IC and chooses appropriate modalities and their supporting media devices. The design of such a system needs to address the key requirements cited below:

Requirement 1: *Provide a relationship between a modality and an IC (i.e. combined user, environment and system contexts) and a relationship between a modality and media devices.* Given that the application domain is multimodality, what parameters constitute the user, environment and system contexts? On what basis a specific modality is considered suitable to an IC? How media devices are selected to support a particular modality?

Requirement 2: *Provide a mechanism that allows the system to acquire incremental acquisition of knowledge related to IC-modalities-media devices scenario.* What machine learning methodology should be adopted if the system is to learn scenarios incrementally? How does it acquire knowledge on new IC scenarios?

Requirement 3: *Provide a mechanism allowing the system to be fault-tolerant on failed media devices.* If a chosen media device fails (i.e. absent or not functional), what media device replacement gets selected, and on what ground?

The technical challenges are addressed by the proposed solutions given below.

Proposed solution to requirement 1: The modalities for human-machine interaction are manual, visual and vocal both for data input and output (details in next section). An IC is composed of user, environment and system context parameters that are all related to modalities. The relationship to consider is how specific modality becomes suitable to an IC parameter and by how much (i.e. high, medium, low or inappropriate). To that effect, all media devices must be grouped in such a way that a relationship between modalities and media group may be established.

Proposed solution to requirement 2: We adopt machine learning; the system is trained with scenarios (i.e. interaction content – modalities) and each one learned is stored in a repository as an exemplar. Using case-based reasoning with supervised learning, a new IC (pre-condition scenario) is compared against stored exemplars; if a match is found, the corresponding post-condition scenario is implemented. Otherwise, a new case is considered for learning.

Proposed solution to requirement 3: We aim to design an autonomous, adaptive and fault-tolerant system. In case of faulty media device, a replacement is searched. Media devices are ranked by priority. The faulty top-ranked device is automatically replaced by second-ranked device (if available) then by the next-ranked device, and so on until a replacement is found. When replacement is not possible, the currently-chosen optimal modality is up for replacement.

4 Interaction Context and Multimodality

An *interaction context*, $\mathsf{IC} = \{\mathsf{IC}_1, \mathsf{IC}_2, \ldots, \mathsf{IC}_{max}\}$, is a set of all possible interaction contexts. At any given time, a user has a specific interaction context i denoted as IC_i, $1 \leq i \leq$ max, which is composed of variables that are present during the conduct of the user's activity. Each variable is a function of the application domain which, in this work, is multimodality. Formally, an IC is a tuple composed of a specific user context (UC), environment context (EC) and system context (SC). An instance of IC is given as:

$$\mathrm{IC_i = UC_k \otimes EC_l \otimes SC_m} \tag{1}$$

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

The user context UC is composed of application domain-related parameters that describe the state of the user during his activity. Any specific user context k is given by:

$$\mathrm{UC_k} = \bigotimes_{x=1}^{\max_k} \mathrm{ICParam}_{kx} \tag{2}$$

where $\mathrm{ICParam_{kv}}$ = parameter of $\mathrm{UC_k}$, k = the number of UC parameters. Similarly, any environment context $\mathrm{EC_l}$ and system context $\mathrm{SC_m}$ are specified as follows:

$$\mathrm{EC_l} = \bigotimes_{y=1}^{\max_l} \mathrm{ICParam}_{ly} \tag{3}$$

$$\mathrm{SC_m} = \bigotimes_{z=1}^{\max_m} \mathrm{ICParam}_{mz} \tag{4}$$

As stated, multimodality selects the modality based on its suitability to the given IC. Here, *modality* refers to the logical interaction structure (i.e. the mode for data input and output between a user and computer). A modality, however, may only be realized if there is/are media devices that would support it. Here, a *media* refers to a set of physical interaction devices (and some software supporting the physical devices). With natural language processing as basis, modalities are grouped as follows: (1) *Visual Input* (VI_{in}), (2) *Vocal Input* (VO_{in}), (3) *Manual/Tactile Input* (M_{in}), (4) *Visual Output* (VI_{out}), (5) *Vocal Output* (VO_{out}), and (6) *Manual/Tactile Output* (M_{out}). Multimodality, therefore, is possible if there is at least one modality for data input and at least one modality for data output:

$$\text{Modality} = (VI_{in} \vee VO_{in} \vee M_{in}) \wedge (VI_{out} \vee VO_{out} \vee M_{out}) \tag{5}$$

Accordingly, media devices themselves are grouped as follows: (1) Visual Input Media (VIM), (2) Visual Output Media (VOM), (3) Oral Input Media (OIM), (4) Hearing Output Media (HOM), (5) Touch Input Media (TIM) (6) Manual Input Media (MIM), and (7) Touch Output Media (TIM). See Fig. 1.

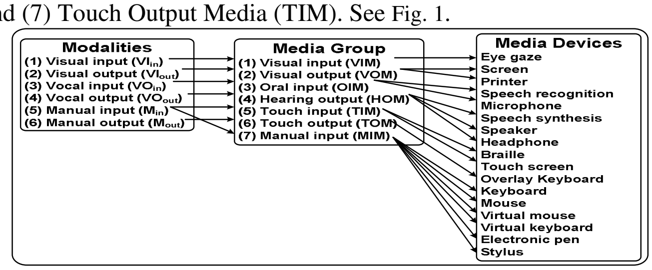

Fig. 1. The relationship among modalities, media group and physical media devices.

For the relationship between modalities and media devices, let there be a function $\mathbf{g_1}$ that maps a modality to a media group, given by $\mathbf{g_1}$*: **Modality → Media group***. This is shown in Fig. 1. There is often a case when two or more media devices belong to one media group. In such a case, media device selection is determined through their priority rankings. Hence, let there be a function $\mathbf{g_2}$ that maps a media group to a media device and the media's priority ranking, denoted by $\mathbf{g_2}$**: *Media group → (Media device, Priority)***. Sample elements of the two stated functions are:

$\mathbf{g_1}$ = {(VI_{in},VIM), (VI_{out},VOM), (VO_{in},OIM), (VO_{out},HOM), (M_{in},TIM), (M_{in},MIM), (M_{out},TOM)}

$\mathbf{g_2}$ = {(VIM, (eye gaze, 1)), (VOM, (screen, 1)), (VOM, (printer, 1)), (OIM, (speech recognition, 1)), (OIM, (microphone, 1)), (HOM, (speech synthesis, 1)), (HOM, (speaker, 2)), (HOM, headphone, 1)), etc.}

A modality's suitability to IC is equal to its collective suitability to IC's individual parameters. Instead of binary (suitable or not), our measure of suitability is that of *high, medium, low* or *inappropriate*. *High suitability* means that the modality being considered is the preferred mode for computing; *medium suitability* means the modality is simply an alternative mode, hence, its absence is not considered as an error but its presence means added convenience to the user. *Low suitability* means the modality's effectiveness is negligible and is the last recourse when everything else fails. *Inappropriateness* recommends that the modality should not be used at all.

If the collective IC is composed of n parameters, then a modality being considered has n suitability scores, one score for each parameter. The following conventions are adopted:

1. A modality's suitability to an IC parameter is one of the following: H (high), M (medium), L (low), and I (inappropriate). Mathematically, H = 1.00, M = 0.75, L = 0.50, and I = 0.
2. The modality's suitability score to an IC is given by:

$$\text{SuitabilityScore}_{\text{modality}} = \sqrt[n]{\prod_{i=1}^{n} \text{context\_parameter}_{\ i}} \tag{6}$$

   where i = parameter index and n = total number of parameters. Given the calculated value, a modality's IC suitability is given by:

$$\text{Suitability}_{\text{modality}} = \begin{cases} \text{H if SuitabilityScore}_{\text{modality}} = 1.00 \\ \text{M if } 0.75 \le \text{SuitabilityScore}_{\text{modality}} < 1.00 \\ \text{L if } 0.50 \le \text{SuitabilityScore}_{\text{modality}} < 0.75 \\ \text{I if SuitabilityScore}_{\text{modality}} < 0.50 \end{cases} \tag{7}$$

Fig. 2 shows the algorithm for determining the suitability of modalities to a given IC and if multimodality is possible (i.e. equation 5). Checking the possibility of multimodality is done by checking that not all of input modalities (i.e. specified by indexes 1, 2 and 3) are scored "inappropriate". The same is true for output modalities (i.e. specified by indexes 4, 5 and 6). The *optimal input modality* is chosen from a group of input modalities, and is one with the highest IC suitability score. The same principle applies to the selection of *optimal output modality*. Subject to the availability of media devices, an optimal modality is ought to be implemented; all other modalities are considered optional. In the absence of supporting media devices, an alternative modality is chosen and is one with the next highest score. The process is repeated until the system finds a replacement modality that can be supported by currently available media devices.

If multimodality is possible and the optimal modalities are chosen, then supporting media devices are checked for availability. Using function $\mathbf{g_1}$, the media group that support the chosen modality may be identified. Given that **Modality** = {VI_{in}, VO_{in}, M_{in}, VI_{out}, VO_{out}, M_{out}} and **Media Group** = {VIM, OIM, MIM, TIM, VOM, HOM, TOM} and that $\mathbf{g_1}$**:** ***Modality → Media group***, then formally, for all media group **p**, there exists a modality **q** such that the mapping between **p** and **q** is in set $\mathbf{g_1}$, that is $\forall$**p: Media group,** $\exists$**q: Modality** $|\ \mathbf{p} \rightarrow \mathbf{q} \in \mathbf{g_1}$. Using function $\mathbf{g_2}$, the top-ranked media devices that belong to such media group are also identified. Given function $\mathbf{g_2}$, a media device **d**, the priorities **p1** and **p2** where Priority: $\mathbf{N_1}$ (positive numbers excluding zero), then the specification for finding the top-ranked device for a media group **m** is $\exists$**m:** Media group, $\forall$**d:** Media device, $\exists$**p1:** Priority, $\forall$**p2:** Priority $|\ \mathbf{d} \bullet \mathbf{m} \rightarrow (\mathbf{d}, \mathbf{p1}) \in \mathbf{g_2} \wedge (\mathbf{p1} < \mathbf{p2})$.

```
//Initialization
Assignment  index i ← 1 to 6 to represent modalities (VIin, VOin, Min, VIout, VOout, and Mout respectively)
 //Evaluate IC suitability of individual modality
Loop i ← 1 to modality_max
    // Calculate modality's IC suitability score
        Loop j ← 1 to parameter_max
       // read suitability level of a modality with respect to parameter i
       Determine suitabilityLevel(j)
            if suitabilityLevel(j) equals
                  (1) High then score ← 1.00, (2) Medium then score ← 0.75
               (3) Low then score ← 0.50, (4) Inappropriate then score ← 0.0
      Calculate finalScore = score ↑ (1/parameter_max)
   If finalScore equals
      (1) 1.00 then Suitability ← High
      (2) 0.75 ≤ finalScore <  1.00 then Suitability ← Medium
      (3) 0.50 ≤ finalScore <  0.75 then Suitability ← Low
      (4) <  0.50 then Suitability ← Inappropriate
  Assign modality[i] ← Suitability
```

```
//check if multimodality is possible
if ((Modality[1] ≠ Inappropriate) OR (Modality[2] ≠  Inappropriate) OR
   (modality[3] ≠ Inappropriate))
      then input modality = OK else input modality = Failure
If (((modality[4] ≠ Inappropriate) OR (modality[5] ≠ Inappropriate) OR
     (modality[6] ≠  Inappropriate))
     then output modality = OK else output modality = Failure

// Multimodality is possible if none of input modality and output modality failed
If input modality = OK and output modality = OK then
  // implement the chosen modalities
  // choose the optimal modality for data input and output
   optimalInputModality ← largest (modality[1], modality[2], modality[3])
   optimalOutputModality ← largest (modality[4], modality[5], modality[6])
```

Fig. 2. Algorithm to determine modality's suitability to IC and if multimodality is possible.

Let there be a *media devices priority table* (MDPT) (see Table 1) which tabulates all media groups, and each media group's set of supporting media devices, arranged by priority ranking. Let **T** = {T_1, T_2… T_{max_table}} be the set of MDPT's. The elements of table $T_n \in \mathbf{T}$, where n = 1 to *max_table*, are similar to elements of function $\mathbf{g_2}$. Every T_n is unique; no two MDPT's are identical. To create a new table, at least one of its elements is different from all other tables that have already been defined. The priority ranking of a specific media device may be different in each MDPT. In general, any given IC scenario and its suitable modalities is mapped/assigned to a specific MDPT.

5 Context Learning and Adaptation

In concept, *machine learning* (ML) is about programming that optimizes a system's performance through the use of sample data or past experiences. ML is important when human expertise does not exist. Hence, learning rule is formulated from acquired data [7]. A machine is said to have "learned" if its performance in doing a task improves with its experience [6]. In this work, the objective of adopting ML is for the system to learn various IC's and each one's selection of appropriate modalities and supporting media devices. Each learned IC is a knowledge learned and is stored in a repository (i.e. it becomes an "exemplar"). When the same IC case reoccurs, the system automatically adopts the IC case's corresponding multimodality selections with little or no human intervention.

Table 1. A sample media devices priority table (MDPT).

Media Group	Media Devices				
	Priority = 1	Priority = 2	Priority = 3	::	Priority = n
Visual Input	Eye Gaze				
Oral Input	Microphone, Speech Recognition				
Touch Input	Touch Screen	Braille Terminal			
Manual Input	Mouse, Keyboard	Virtual Mouse, Virtual keyboard	Electronic Pen	Stylus	Braille
Visual Output	Screen	Printer	Electronic Projector		
Hearing Output	Speaker	Headphone, Speech Synthesis			
Touch Output	Braille	Overlay Keyboard			

System knowledge acquisition begins with the establishment of a priori knowledge, those related to IC parameters. An example of an IC parameter is shown in Table 2. As shown, the IC parameter is the "user location". The value of this parameter is deduced from the data taken by a sensor (i.e. a GPS). To formulate this specific a priori knowledge, some specified values of latitude and longitude are assigned with specific meanings (we call them "conventions"). When sample sensor readings are taken, the system compares them with the conventions and concludes whether the user is "at home", "at work" or "on the go". Then, the expert (i.e. end user) is supplies his perceived suitability score of each modality for each user location convention (see Table 2(b)). Hence, based on the given value of an IC parameter (e.g. user location), the system easily retrieves the suitability score of each modality.

Table 2. Sample User context parameter – convention and modalities selections

(a): User location convention table using GPS values

Convention No.	Latitude	Longitude	Meaning
1	$<value_{11}>$	$<value_{12}>$	*At home*
2	$<value_{21}>$	$<value_{22}>$	*At work*
3	not $<value_{11}>$ *AND* not $<value_{21}>$	not $<value_{21}>$ *AND* not $<value_{22}>$	*On the go*

(b): Modality selection based on user location

Type of Modality	User location = At home	User location = At work	User location = On the go
Visual Input	H	H	L
Visual Output	H	H	H
Vocal Input	H	H	H
Vocal Output	H	H	H
Manual Input	H	H	H
Manual Output	H	H	H

In general, if a system is to become reliable in its detection of the suitability of all modalities to a given IC, it needs the most a priori knowledge on context parameters as possible. In our work, an end user can add, modify, and delete one context parameter at a time using our layered virtual machine for incremental definition of IC (implemented but not shown here due to space constraints). When all the a priori knowledge are collected and grouped together, it forms a tree-like IC structure, as shown in

Fig. 3. Every new IC parameter is first classified as either UC or EC or SC parameter and is appended as a branch of UC or EC or SC. Then, the conventions of the parameter are identified as well as the modalities' suitability scores in each convention.

There are cases, however, when a certain IC parameter's value could nullify the importance of another IC parameter. For example, the declaration

user_handicap (blind) *nullifies* light_intensity()

states that UC parameter "user handicap" nullifies the EC parameter "light intensity". As such, whatever light intensity value or convention is identified by a sensor is simply ignored in the calculation of the overall modality's suitability to the given IC.

Distinct scenarios that the system had encountered are stored in the knowledge database as an exemplar while a current one becomes a "case". A *case* is composed of three elements: (1) *the problem* – the IC in consideration, composed of UC, EC and SC parameters and their values or conventions, (2) *the solution* – the final IC suitability of each modality, and (3) *the evaluation* – the relevance score of the solution.

When the ML component receives a new scenario (i.e. new IC), it converts it into a case, specifying the problem. Using the similarity algorithm, it compares the problem in the new case against all the available problems/exemplars in the knowledge database. The scenario of the closest match is selected and its solution is returned. The evaluation is the score of how similar it is to the closest match. If no match is found (relevance score is low), the ML component takes the closest various scenarios and regroup and organized them to find the solution of the new case. The user may or may not accept the proposed solution. In case of expert refusal, a new case with supervised learning is produced, the problem to the case resolved. Afterwards, ML component adds the new case in its knowledge database. This whole learning mechanism is called *case-based reasoning with supervised learning.*

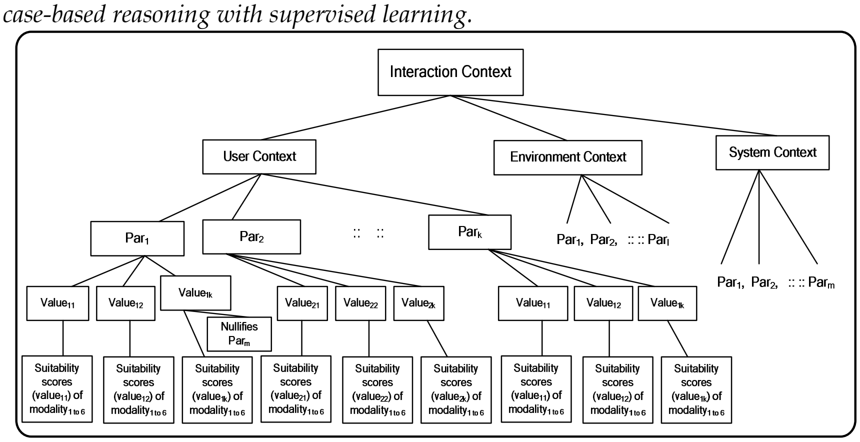

Fig. 3. The structure of stored IC parameters.

Inspired by [15], we modify their similarity scoring scheme to reflect the needs of our system. Hence, given a new case (NC) and an individual case stored in the knowledge database (MC), the similarity of the problem between the two cases, that is NC against MC as denoted by the subscripts, is equal to their similarity in the case's UC, EC and SC and is given by:

$$\text{Sim}(\text{NC},\text{MC}) = \tfrac{1}{3}\text{Sim}(\text{UC}_{\text{NC}},\text{UC}_{\text{MC}}) + \tfrac{1}{3}\text{Sim}(\text{EC}_{\text{NC}},\text{EC}_{\text{MC}}) + \tfrac{1}{3}\text{Sim}(\text{SC}_{\text{NC}},\text{SC}_{\text{MC}}) \quad (8)$$

The similarity between the UC of NC against the UC of MC is given by:

$$\text{Sim}(\text{UC}_{\text{NC}},\text{UC}_{\text{MC}}) = \frac{\sum_{i=1}^{\max_{\text{NC}}} \text{Sim}(\text{UC\_Par}_{i_{\text{NC}}},\text{UC}_{\text{MC}})}{\max(\text{UC\_Par}_{\text{NC}},\text{UC\_Par}_{\text{MC}})} \quad (9)$$

where UC_Par_i, i = 1 to max, is the individual UC parameter, $max(UC\_Par_{NC}, UC\_Par_{MC})$ is the greater between the number of UC parameters between NC and MC, and $\text{Sim}(\text{UC\_Par}_{i\text{NC}},\text{UC}_{\text{MC}}) = \max_{j=1 \text{ to } \max_{\text{MC}}} \text{Sim}(\text{UC\_Par}_{i\text{NC}},\text{UC\_Par}_{j_{\text{MC}}})$ where $\text{UC\_Par}_{j_{\text{MC}}} \in \text{UC}_{\text{MC}}$ and $\text{Sim}(\text{UC\_Par}_{i\text{NC}},\text{UC\_Par}_{j_{\text{MC}}}) \in [0, 1]$ is the similarity between a specific UC parameter i of NC and parameter j of MC.

For the similarity measures of EC of NC against EC of MC, and the SC of NC against SC of MC, the same principle as Equation 9 must be applied, with the formula adjusted accordingly to denote EC and SC, respectively, yielding:

$$\text{Sim}(\text{EC}_{\text{NC}},\text{EC}_{\text{MC}}) = \frac{\sum_{i=1}^{\max_{\text{NC}}} \text{Sim}(\text{EC\_Par}_{i_{\text{NC}}},\text{EC}_{\text{MC}})}{\max(\text{EC\_Par}_{\text{NC}},\text{EC\_Par}_{\text{MC}})} \quad (10)$$

$$\text{Sim}(\text{SC}_{\text{NC}},\text{SC}_{\text{MC}}) = \frac{\sum_{i=1}^{\max_{\text{NC}}} \text{Sim}(\text{SC\_Par}_{i_{\text{NC}}},\text{SC}_{\text{MC}})}{\max(\text{SC\_Par}_{\text{NC}},\text{SC\_Par}_{\text{MC}})} \quad (11)$$

Equation 8 assumes that the weights of UC, EC and SC are equal (i.e. each is worth 33.3%). This figure is not fixed and can be adjusted to suit the need of the expert. An ideal case match is a perfect match. However, a score of 90% means that a great deal of IC parameters is correctly considered and is therefore 90% accurate. The expert, however, decides the threshold score of what is considered as an acceptable match.

When the IC-appropriate modalities are satisfactorily identified, the media devices supporting the modalities are checked for availability. If available, the devices are simply activated. Otherwise, a replacement is searched. Via MDPT, the media device that is next in priority is searched. The process is repeated until a replacement is found (see Fig. 4). Formally, given a failed device **d** of priority **p1,** the specification for finding the replacement media device $\mathbf{d_{rep}}$ is **∃m:** Media Group, **∀drep:** Media Device, **∃p1:** Priority, **∀p2:** Priority $\mid$ **(p1 = p1 + 1) ∧ (p1 < p2) ∧ m → (drep, p1) ∈** $\mathbf{g_2}$ **•** $\mathbf{d_{rep}}$**.**

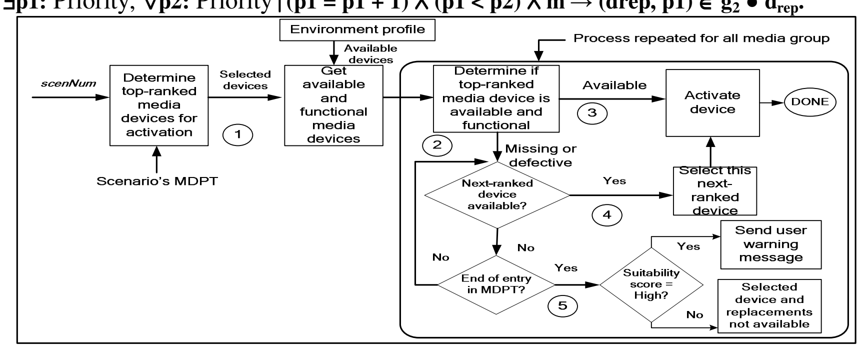

Fig. 4. Algorithm for finding replacement to a failed media device.

6 Conclusion

In this paper, we presented the major challenges in designing the infrastructure of pervasive multimodality. We address those challenges by presenting the elements that comprise a given interaction context, the grouping of modalities, media groups and media devices. We establish the relationship among them and provide their formal specifications. Machine learning is used to build an autonomous and interaction context-adaptive system. Such learning system needs a priori knowledge on context parameters and the methodology to augment it incrementally. Also, the acquisition of various scenarios is presented in this paper. Finally, we demonstrate one fault-tolerant characteristic of the system by providing the mechanism that finds a replacement to a failed media device.

References

1. Weiser, M., The computer for the twenty-first century. Scientific American, Vol. **265**(3), (1991) 94 - 104.
2. Vasilakos, A.V. and Pedrycz, W. Ambient Intelligence, Wireless Networking, Ubiquitous Computing, ArtecHouse, USA (2006).
3. Satyanarayanan, M., Pervasive Computing: Vision and Challenges. IEEE Personal Communications, Vol. **8**(4), (2001) 10-17.
4. Coutaz, J., et al., Context is key. Communications of the ACM, Vol. **48**(3) (2005), 49-53.
5. Ringland, S.P.A. and F.J. Scahill, Multimodality - The future of the wireless user interface. BT Technology Journal, Vol. **21**(3), (2003) 181-191.
6. Mitchell, T., Machine Learning, McGraw-Hill (1997).
7. Alpaydin, E., Introduction to Machine Learning, Cambridge, MA, USA, MIT Press (2004).
8. Oviatt, S.L. and Cohen, P.R., Multimodal Interfaces that Process What Comes Naturally. Communications of the ACM, Vol. **43**(3), (2000) 45 - 53.
9. Oviatt, S.L., Designing the User Interface for Multimodal Speech and Gesture Applications: State-of-the-art Systems and Research Directions. Human Computer Interaction, Vol. **15**(4), (2000), 263-322.
10. Rubin, P., Vatikiotis-Bateson, E. and Benoit, C., Audio-Visual Speech Processing. Speech Communications, Vol. **26**(1-2) (1998).
11. 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: Springer-Verlag, LNCS 1707 (1999).
12. Dey, A.K., Understanding and Using Context Springer Personal and Ubiquitous Computing, Vol. **5**(1), (2001) 4 - 7.
13. Rey, G. and Coutaz. J., Le contexteur: une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte in Proceedings of the 14th French-speaking Conference on Human-Computer Interaction (Conférence Francophone sur l'Interaction Homme-Machine) IHM '02, Poitiers, France, ACM Press (2002).
14. Garlan, D., et al., Project Aura: Towards Distraction-Free Pervasive Computing. IEEE Pervasive Computing, Special Issue on Integrated Pervasive Computing Environments, Vol. **21**(2), (2002) 22 - 31.
15. Lajmi, S., Ghedira, C. and Ghedira, K., Une méthode d'apprentissage pour la composition de services web. L'Objet, Vol. **8**(2), (2007), 1 - 4.