ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MANUSCRIPT-BASED THESIS
PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DE VERSAILLES-SAINT-QUENTIN-EN-YVELINES
(COTUTORSHIP)

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

COTUTORSHIP
UNIVERSITÉ DE VERSAILLES-SAINT-QUENTIN-EN-YVELINES-QUEBEC

BY
Manolo Dulva HINA

A PARADIGM OF AN INTERACTION CONTEXT-AWARE PERVASIVE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM

MONTREAL, SEPTEMBER 14, 2010

©Copyright 2010 reserved by Manolo Dulva Hina

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Chakib Tadj, Thesis Supervisor
Département de Génie électrique à l'École de technologie supérieure

Ms. Nicole Lévy, Thesis Co-director
Laboratoire PRISM à l'Université de Versailles-Saint-Quentin-en-Yvelines, France

Mr. Michael J. McGuffin, President of the Board of Examiners
Département de génie logiciel et des TI à l'École de technologie supérieure

Mr. Roger Champagne, Professor
Département de génie logiciel et des TI à l'École de technologie supérieure

Mr. Amar Ramdane-Cherif, Professor
Laboratoires PRISM & LISV à l'Université de Versailles-Saint-Quentin-en-Yvelines, France

Ms. Isabelle Borne, Professor
Laboratoire VALORIA à l'Université de Bretagne-Sud – IUT de Vannes, France

THIS THESIS WAS PRESENTED AND DEFENDED BEFORE A BOARD OF EXAMINERS AND THE PUBLIC

23 JULY 2010

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This thesis is a work of partnership within the framework of cotutorship (cotutelle de thèse) between the "Laboratoire des Architectures du Traitement de l'Information et du Signal" (LATIS) laboratory of Université du Québec, École de technologie supérieure in Canada and of "Parallélisme, des Réseaux, des Systèmes et de la Modélisation" (PRISM) laboratory of Université de Versailles-Saint-Quentin-en-Yvelines in France.

The theme of this research work is related to the design of an infrastructure and modeling of a pervasive multimodal multimedia computing system that can adapt accordingly to a large context called interaction context. This adaptation is through dynamic configuration of architecture, meaning the system intervenes on behalf of the user to modify, add or delete a system component and activate another without explicit intervention from the user. This is in conformity to calm technology as emphasized by Weiser in his vision of ubiquitous computing. The architecture design of our system is intelligent and its components are robust.

This work is a result of research work and partnership of LATIS and PRISM laboratories, its advisors and its student researchers. In PRISM laboratory, under the supervision of Dr. Nicole Lévy and Dr. Amar Ramdane-Cherif, previous researches were made in the multi-agent platforms for dynamic reconfiguration of software architectures, such as that of Djenidi (Djenidi 2007) and Benarif (Benarif 2008). In LATIS laboratory, under the supervision of Dr. Chakib Tadj, great effort were made to come up with research of deep significance on the use multimodality and multimedia. Some of these works are those of Awdé (Awdé 2009) and Miraoui (Miraoui 2009). Those research works are related to this work in more areas than one. The programming of the layered virtual machine for incremental interaction context was done in coordination with an ÉTS student partner, provided to me by Dr. Tadj. Other works that have great influenced to this thesis include that of Dey (Dey 2000), Chibani (Chibani 2006) and Garlan (Garlan, Siewiorek et al. 2002).

ACKNOWLEDGEMENTS

Thanks are in order for all the people who have helped me in realizing this doctorate thesis.

My heartfelt gratitude is in order for my thesis director, Dr. Chakib Tadj. He helped me since day one in getting into the doctorate program, in giving me advices in courses, in article writing, in securing grants, in dealing language problems between English and French, in day-to-day personal problems, and in thesis writing. Without his guidance, this thesis would not have been made possible.

My deepest thank also goes to Dr. Amar Ramdane-Cherif. He was practically my guide in getting the day-to-day affair done whenever I am in France. And I have been there 3 times, each time being a 3 month stint. Of course, he was and has been a guiding light to me in all my article writings. He also supported me in other academic endeavors which I am truly grateful.

My sincerest thank goes to Dr. Nicole Lévy as well who happened to be my thesis co-director, based in France. Her criticisms with regards to my work helped me a lot in polishing all my works.

Many thanks as well to all the members of the jury – all for taking their time in reading and criticizing my work for the better – to Dr. Michael J. McGuffin for presiding the jury, to Dr. Roger Champagne and Dr. Isabelle Borne for their effort in reviewing this work. Without them, the thesis defence would not have been made possible.

I wish to say thank you as well to all the men and women behind all the grants that I received during my doctorate studies Thanks are in order to all the following institutions: (1) École de technologie supérieure, Décanat des études – for the grants they have given me for three years; (2) Bourse de cotutelle /Bourse Frontenac/Cooperation France-Quebec – for the grants that made it possible for me to stay in Paris and do research in Université de Versailles-Saint-

Quentin-en-Yvelines; (3) National Bank of Canada – for the grant; (4) Natural Sciences and Engineering Research Council of Canada – for the grant they accorded to my thesis director, Dr. Tadj, which also helped us, researchers, to do our work; and (5) ÉGIDE in Paris, France – for helping me in facilitating my grant, accommodation and medical and health needs during all my stints in France.

Apart from academic people mentioned above, I would also like to thank all my colleagues in LATIS and PRISM laboratories who accorded me helps during my laboratory works in Montreal and in Versailles. Special thanks to Ali Awdé and Lydia Michotte for all the support – both academic and personal – they accorded me.

I would also like to thank Dr. Sylvie Ratté of ÉTS who happened to be my professor in MGL 806 ("Specifications formelles et semi-formelles") for all the assistance she accorded me during the course of my study at ÉTS.

Without citing specific names because there are many, I wish to thank as well all my friends in Concordia University, at home in Muntinlupa City, Philippines and the Bonyad clan in Montreal, my colleagues in Benix & Co. in Montreal, and all friends in Paris, France who, in one way or another, helped me morally to get over through this study. They were all the wind beneath my wings.

Wherever they may be, I wish to dedicate this thesis to my parents, Tobias Hina and Magdalena Dulva. It is just unfortunate that I was not able to complete this thesis while they were still alive. Wherever they are, thanks mom and dad for all your love.

LE PARADIGME D'UN SYSTÈME MULTIMODAL MULTIMÉDIA UBIQUITAIRE SENSIBLE AU CONTEXTE D'INTERACTION

Manolo Dulva HINA

RÉSUMÉ

La communication est un aspect très important de la vie humaine ; elle permet aux êtres humains de se rapprocher les uns avec les autres comme individus et en tant que groupes indépendants. En informatique, le but même de l'existence de l'ordinateur est la diffusion de l'information - de pouvoir envoyer et recevoir l'information. Cependant, la capacité d'échanger de l'information entre humains ne se transfère pas quand l'humain interagit avec l'ordinateur. Sans intervention externe, les ordinateurs ne comprennent pas notre langue, ne comprennent pas comment le monde fonctionne et ne peuvent percevoir des informations sur une situation donnée. Dans une installation typique traditionnelle (souris - clavier - écran) l'information explicite fournie à l'ordinateur produit un effet contraire à la promesse de transparence et à la technologie *calme* ; c'était la vision du calcul omniprésent de Weiser (Weiser 1991 ; Weiser et Brown 1996). Pour renverser cette tendance, nous devons trouver les moyens et la méthodologie qui permettent à des ordinateurs d'avoir accès au contexte. C'est par ce dernier que nous pouvons augmenter la richesse de la communication dans l'interaction personne-ordinateur, et donc de bénéficier des avantages le plus susceptibles des services informatiques.

Comme le montre bien la littérature, le contexte est une idée subjective qui évolue dans le temps. Son interprétation est généralement propre au chercheur. L'acquisition de l'information contextuelle est essentielle. Cependant, c'est l'utilisateur qui décidera si le contexte envisagé est correctement capturé/acquis ou pas. La littérature montre que l'information contextuelle est prédéfinie par quelques chercheurs dès le début – ceci est correcte si le domaine d'application est fixe. Cette définition devient incorrecte si nous admettons qu'un utilisateur typique réalise différentes tâches de calcul à différentes occasions. Dans le but de proposer une conception plus concluante et plus inclusive, nous pensons que le contenu de l'information contextuelle ne devrait être défini que par l'utilisateur. Ceci nous mène au concept de l'acquisition incrémental du contexte où des paramètres de contexte sont ajoutés, modifiés ou supprimés, un paramètre de contexte à la fois.

Dans ce même ordre d'idée, nous élargissons la notion du contexte au *contexte de l'interaction* (CI). Le CI est le terme qui est employé pour se rapporter au contexte collectif de l'utilisateur (c.-à-d. *contexte d'utilisateur*), de son milieu de travail (c.-à-d. *contexte d'environnement*) et de son système de calcul (c.-à-d. *contexte de système*). Logiquement et mathématiquement, chacun de ces éléments de CI - contexte d'utilisateur, contexte d'environnement et contexte de système - se compose de divers paramètres qui décrivent

l'état de l'utilisateur, de son lieu de travail et de ses ressources informatiques pendant qu'il entreprend une activité en accomplissant sa tâche de calcul. Chacun de ces paramètres peut évoluer avec le temps. Par exemple, la localisation de l'utilisateur est un paramètre de contexte d'utilisateur et sa valeur évoluera selon le déplacement de l'utilisateur. Le niveau de bruit peut être considéré comme paramètre de contexte d'environnement ; sa valeur évolue avec le temps. De la même manière, la largeur de bande disponible qui évolue sans interruption est considérée comme paramètre de contexte de système. Pour réaliser une définition incrémentale du contexte, nous avons développé un outil appelé *machine virtuelle à couches pour le contexte de l'interaction*. Cet outil peut être utilisé pour : a) ajouter, modifier et supprimer un paramètre de contexte d'une part et b) déterminer le contexte dépendamment des senseurs (c.-à-d. le contexte est déterminé selon les paramètres dont les valeurs sont obtenues à partir des données brutes fournies par des senseurs).

Afin de maximiser les bienfaits de la richesse du CI dans la communication personne-machine, la modalité de l'interaction ne devrait pas être limitée à l'utilisation traditionnelle souris-clavier-écran. La multimodalité tient compte d'un éventail de modes et de formes de communication, choisis et adaptés au contexte de l'utilisateur. Dans la communication multimodale, les faiblesses d'un mode d'interaction sont compensées en le remplaçant par un autre mode de communication qui est plus appropriée à la situation. Par exemple, quand l'environnement devient fâcheusement bruyant, l'utilisation de la voix n'est pas appropriée ; l'utilisateur peut opter pour la transmission de texte ou l'information visuelle. La multimodalité favorise également l'informatique inclusive comme ceux ayant un handicap permanent ou provisoire. Par exemple, la multimodalité permet d'utiliser une façon originale pour présenter des expressions mathématiques aux utilisateurs malvoyants (Awdé 2009). Avec le calcul mobile, la *multimodalité ubiquitaire et adaptative* est plus que toujours susceptible d'enrichir la communication dans l'interaction personne-machine et de fournir les modes les plus appropriés pour l'entrée / la sortie de données par rapport à l'évolution du CI.

Un regard à la situation actuelle nous informe qu'un grand effort a été déployé en trouvant la définition du contexte, dans l'acquisition du contexte, dans la diffusion du contexte et l'exploitation du contexte dans un système qui a un domaine d'application fixe (par exemple soins de santé, l'éducation, etc.). Par ailleurs, des efforts de recherches sur le calcul ubiquitaire étaient développés dans divers domaines d'application (par exemple localisation de l'utilisateur, identification des services et des outils, etc.). Cependant, il ne semble pas y avoir eu un effort pour *rendre la multimodalité ubiquitaire et accessible à diverses situations de l'utilisateur*. À cet égard, nous fournissons un travail de recherche qui comblera le lien absent. Notre travail – *Le paradigme du système multimodal multimédia ubiquitaire sensible au contexte de l'intéraction* – est une conception architecturale qui montre l'adaptabilité à un contexte beaucoup plus large appelé le *contexte d'interaction*. Il est intelligent et diffus, c.-à-d. fonctionnel lorsque l'utilisateur est stationnaire, mobile ou sur la route. Il est conçu avec deux buts à l'esprit. D'abord, étant donné une instance de CI qui évolue avec le temps, notre système *détermine les modalités optimales* qui s'adaptent à un tel CI. Par optimal, nous

entendons le choix des modalités appropriées selon le contexte donné de l'interaction, les dispositifs multimédias disponibles et les préférences de l'utilisateur. Nous avons conçu un mécanisme (c.-à-d. un paradigme) qui réalise cette tâche. Nous avons également simulé sa fonctionnalité avec succès. Ce mécanisme utilise l'*apprentissage de la machine* (Mitchell 1997 ; Alpaydin 2004 ; Hina, Tadj et al. 2006) et un *raisonnement à base de cas avec apprentissage supervisé* (Kolodner 1993 ; Lajmi, Ghedira et al. 2007). L'entrée à ce composant est une instance de CI. Les sorties sont a) la modalité optimale et b) les dispositifs associés. Ce mécanisme contrôle continuellement le CI de l'utilisateur et s'adapte en conséquence. Cette adaptation se fait par la reconfiguration dynamique de l'architecture du système multimodal diffus. En second lieu, étant donné une instance de CI, la tâche et les préférences de l'utilisateur, nous avons conçu un mécanisme qui permet le choix automatique des applications de l'utilisateur, les fournisseurs préférés à ces applications et les configurations préférées de la qualité du service de ces fournisseurs. Ce mécanisme fait sa tâche en consultation avec les ressources informatiques, percevant les fournisseurs disponibles et les restrictions possibles de configuration.

Indépendamment des mécanismes mentionnés ci-dessus, nous avons également formulé des scénarios quant à la façon dont un système doit présenter l'interface utilisateurs étant donné que nous avons déjà identifié les modalités optimales qui s'adaptent au CI de l'utilisateur. Nous présentons des configurations possibles d'interfaces unimodales et bimodales fondées sur le CI donné et les préférences de l'utilisateur.

Notre travail est différent du reste des travaux précédents dans le sens que notre système capture le CI et modifie son architecture dynamiquement de façon générique pour que l'utilisateur continue de travailler sur sa tâche n'importe quand n'importe où, indépendamment du domaine d'application. En effet, le système que nous avons conçu est généralement générique. Il peut être adapté ou intégré facilement dans divers systèmes de calcul, dans différents domaines d'applications, avec une intervention minimale. C'est notre contribution à ce domaine de recherche.

Des simulations et des formulations mathématiques ont été fournies pour soutenir nos idées et concepts liés à la conception du paradigme. Un programme Java a été développé pour soutenir notre concept de la machine virtuelle à couches pour le CI incrémental.

Mots clés : Interaction homme-machine, interface multimodale, système diffus, système multimodal multimédia, architecture logicielle.

A PARADIGM OF INTERACTION CONTEXT-AWARE PERVASIVE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM

Manolo Dulva HINA

ABSTRACT

Communication is a very important aspect of human life; it is communication that helps human beings to connect with each other as individuals and as independent groups. Communication is the fulcrum that drives all human developments in all fields. In informatics, one of the main purposes of the existence of computer is information dissemination – to be able to send and receive information. Humans are quite successful in conveying ideas to one another, and reacting appropriately. This is due to the fact that we share the richness of the language, have a common understanding of how things work and an implicit understanding of everyday situations. When humans communicate with humans, they comprehend the information that is apparent to the current situation, or *context*, hence increasing the conversational bandwidth. This ability to convey ideas, however, does not transfer when humans interact with computers. On its own, computers do not understand our language, do not understand how the world works and cannot sense information about the current situation. In a typical computing set-up where we have an impoverished typical mechanism for providing computer with information using mouse, keyboard and screen, the end result is we explicitly provide information to computers, producing an effect that is contrary to the promise of *transparency* and *calm technology* in Weiser's vision of *ubiquitous computing* (Weiser 1991; Weiser and Brown 1996). To reverse this trend, it is imperative that we researchers find ways that will enable computers to have access to context. It is through context-awareness that we can increase the *richness* of *communication* in *human-computer interaction*, through which we can reap the most likely benefit of more useful computational services.

Context is a subjective idea as demonstrated by the state-of-the art in which each researcher has his own understanding of the term, which continues to evolve nonetheless. The acquisition of contextual information is essential but it is the end user, however, that will have the final say as to whether the envisioned context is correctly captured/acquired or not. Current literature informs us that some contextual information is already predefined by some researchers from the very beginning – this is correct if the application domain is fixed but is incorrect if we infer that a typical user does different computing tasks on different occasions. With the aim of coming up with *more conclusive* and *inclusive* design, we conjecture that what contextual information should be left to the judgment of the end user who is the one that has the knowledge determine which information is important to him and which is not. This leads us to the concept of *incremental acquisition of context* where context parameters are added, modified or deleted one context parameter at a time.

In conjunction with our idea of inclusive context, we broaden the notion of context that it has become *context of interaction*. Interaction context is the term that is used to refer to the

collective context of the user (i.e. *user context*), of his working environment (i.e. *environmental context*) and of his computing system (i.e. *system context*). Logically and mathematically, each of these interaction context elements – user context, environment context and system context – is composed of *various parameters* that describe the state of the user, of his workplace and his computing resources as he undertakes an activity in accomplishing his computing task, and each of these parameters may evolve over time. For example, user location is a user context parameter and its value will evolve as the user moves from one place to another. The same can be said about noise level as an environment context parameter; its value evolves over time. The same can be said with available bandwidth that continuously evolves which we consider as a system context parameter. To realize the incremental definition of incremental context, we have developed a tool called the *virtual machine for incremental interaction context*. This tool can be used to add, modify and delete a context parameter on one hand and determine the sensor-based context (i.e. context that is based on parameters whose values are obtained from raw data supplied by sensors) on the other.

In order to obtain the full benefit of the richness of interaction context with regards to communication in human-machine interaction, the modality of interaction should not be limited to the traditional use of mouse-keyboard-screen alone. *Multimodality* allows for a much wider range of modes and forms of communication, selected and adapted to suit the given user's context of interaction, by which the end user can transmit data to the computer and computer can respond or yield results to the user's queries. In multimodal communication, the weaknesses of one mode of interaction, with regards to its suitability to a given situation, is compensated by replacing it with another mode of communication that is more suitable to the situation. For example, when the environment becomes disturbingly noisy, using voice may not be the ideal mode to input data; instead, the user may opt for transmitting text or visual information. Multimodality also promotes inclusive informatics as those with a permanent or temporary disability are given the opportunity to use and benefit from information technology advancement. For example, the work on presentation of mathematical expressions to visually-impaired users (Awdé 2009) would not have been made possible without multimodality. With mobile computing within our midst coupled with wireless communication that allows access to information and services, *pervasive* and *adaptive multimodality* is more than ever apt to enrich communication in human-computer interaction and in providing the most suitable modes for data input and output in relation to the evolving interaction context.

A look back at the state of the art informs us that a great amount of effort was expended in finding the definition of context, in the acquisition of context, in the dissemination of context and the exploitation of context within a system that has a fixed domain of application (e.g. healthcare, education, etc.). Also, another close look tells us that much research efforts on ubiquitous computing were devoted to various application domains (e.g. identifying the user whereabouts, identifying services and tools, etc.) but there is rarely, if ever, an effort made to *make multimodality pervasive and accessible to various user situations*. In this regard, we come up with a research work that will provide for the missing link. Our work – *the paradigm of an interaction context-sensitive pervasive multimodal multimedia computing* *system* is an architectural design that exhibits *adaptability* to a much larger context called interaction context. It is intelligent and pervasive, meaning it is functional even when the end user is stationary or on the go. It is conceived with two purposes in mind. First, given an instance of interaction context, one which evolves over time, our system *determines the optimal modalities* that suit such interaction context. By optimal, we mean a selection decision on appropriate multimodality based on the given interaction context, available media devices that support the modalities and user preferences. We designed a mechanism (i.e. a paradigm) that will do this task and simulated its functionality with success. This mechanism employs *machine learning* (Mitchell 1997; Alpaydin 2004; Hina, Tadj et al. 2006) and uses *case-based reasoning with supervised learning* (Kolodner 1993; Lajmi, Ghedira et al. 2007). An input to this decision-making component is an instance of interaction context and its output is the optimal modality and its associated media devices that are for activation. This mechanism is continuously monitoring the user's context of interaction and on behalf of the user continuously adapts accordingly. This *adaptation* is through *dynamic reconfiguration* of the *pervasive multimodal system's architecture*. Second, given an instance of interaction context and the user's task and preferences, we designed a mechanism that allows the *automatic selection of user's applications, the preferred suppliers* to these applications and the *preferred quality of service (QoS) dimensions'* configurations of these suppliers. This mechanism does its task in consultation with computing resources, sensing the available suppliers and possible configuration restrictions within the given computing set-up.

Apart from the above-mentioned mechanisms, we also formulated scenarios as to how a computing system must provide the *user interface* given that we have already identified the optimal modalities that suit the user's context of interaction. We present possible configurations of *unimodal* and *bimodal interfaces* based on the given interaction context as well as *user preferences*.

Our work is different from previous work in that while other systems capture, disseminate and consume context to suit the preferred domain of application, ours captures the interaction context and *reconfigures* its *architecture dynamically* in *generic* fashion in order that the user could continue working on his task anytime, anywhere he wishes regardless of the application domain the user wishes to undertake. In effect, the system that we have designed along with all of its mechanisms, being generic in design, can be adapted or integrated with ease or with very little modification into various computing systems of various domains of applications.

Simulations and mathematical formulations were provided to support our ideas and concepts related to the design of the paradigm. An actual program in Java was developed to support our concept of a virtual machine for incremental interaction context.

Keywords: Human-machine interface, multimodal interface, pervasive computing, multimodal multimedia computing, software architecture.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBEVIATIONS, INITIALS AND ACRONYMS

ADD	attribute-driven design
CBR	case-based reasoning with supervised learning
CMA	The Context Manager Agent
CPU	central processing unit
EC	environmental context
EMA	Environmental Manager Agent
GPRS	general packet radio services
HCI	human-computer interaction
HKA	History and Knowledge-based Agent
HMIIPT	human-machine interaction interface priority table
HOM	hearing output media group
HPSim	a software package used to implement Petri Net in modeling software systems
IC	interaction context
M_{in}	manual input modality
M_{out}	manual output modality
MC	old cases or memory cases in CBR
MDPT	media devices priority table
MIM	manual input media group
ML	machine learning
NC	new case in CBR
OCL	object constraint language; similar to Z but used to described the system informally; it is used to describe systems in object-oriented concepts.
OIM	oral input media group
QoS	quality of service
PMMCS	pervasive multimodal multimedia computing system
RDF	resource description framework
SC	system context
SCA	System Context Agent
TIM	touch input media group

TMA	Task Manager Agent
UC	user context
UML	unified modeling language; similar to OCL; used to describe the system informally that uses diagrams to show relationships among system components.
UMTS	universal mobile telecommunications system
VI_{in}	visual input modality
VI_{out}	visual output modality
VIM	visual input media group
VM	virtual machine
VMIC	Virtual Machine for Interaction Context
VOM	visual output media group
VO_{in}	vocal input modality.
VO_{out}	vocal output modality
W3C CC/PP	world wide web consortium composite capabilities/preferences profile
WiFi	wireless fidelity
Z	a specific formal specification language, one that is commonly used to describe a system using mathematical and logical formulation based on the concept of sets.

LIST OF SYMBOLS AND UNITS OF MEASUREMENT

$\in$	denotes element of a set
[x, y]	closed interval; the range of possible values are greater than or equal to x but less than or equal to y.
(m, n]	half-open interval; the range of possible values are greater than m but less than or equal to n.
$\mathcal{P}$(M)	power set of M, all the possible subsets of set M.
¡(M)	power set of M, all the possible subsets of set M.
$\hat{M}$	optimal value of set M
M	set M (note the bold letter denoting that a letter signifies a set).
$\forall$	universal quantifier (i.e. for all)
$\exists$	existential quantifier (i.e. there exists)
£$_1$	set of integers whose minimum value is 1
£	set of all integers – negative numbers, zero and positive numbers
$\wedge$	logical AND
$\vee$	logical OR
$\rightarrow$	propositional logic of implication
$\otimes$	Cartesian product, yields all possible ordered pairs
Π	product of all the items that are considered
Σ	summation of all the items in consideration

g_1: Modality → Media Group – a logical function that maps a modality to a media device group.

g_2: Media Group → (Media Device, Priority) - a logical function that maps or associates each element of the set of media group to a set of media devices and their corresponding priority rankings.

f_1: Data format → Application – a logical function that maps a set of data format (i.e. of form filename.extension) to a certain application

f_2: Application → (Preferred Supplier, Priority) – a logical function that maps or associates an application to a user's preferred supplier and its corresponding priority in user's preference.

f_3: Application → (QoS dimension j, Priority) – a logical function that maps a specific application to its set of quality of service dimension j (j = 1 to max) and such dimension's priority ranking.

INTRODUCTION

Context of Research Work

In 1988, Marc Weiser envisioned the concepts of *ubiquitous computing* (Weiser 1991) also known as *pervasive computing*: (1) that the purpose of a computer is to help you do something other than thinking of its configuration, (2) that the best computer is a quiet, invisible servant, (3) that the more the user uses intuition, the smarter he becomes and that the computer should the user's unconscious, and (4) that the technology should be calm, one that informs but not demands our focus and attention. Indeed, in this era, the user can do computing stationary- or mobile-wise, enabling him to continue working on his task whenever and wherever he wishes. To this effect, the user's computing task should be made ubiquitous as well. This can be accomplished by making the user's task, profile, data and task registry transportable from one environment to another. To realize ubiquitous computing, a network system that supports *wired* and *wireless* computing (Tse and Viswanath 2005) must exist.

A *multimodal multimedia system* advocates the use of human action (e.g. speech, gesture) along with the usual computing media devices (e.g. mouse, keyboard, screen, speaker, etc.) as means of data input and output. Multimodality along with multimedia is important as it advances information technology in accepting what is human in conveying information (i.e. speech, gesture, etc.). Likewise, it enables people with disability to take advantage of human action (e.g. speech) to replace devices that otherwise are not suited for their situation. The recognition of user's situation is necessary in deciding which modality and media devices are suitable to the user at a given time. The effectiveness of multimodality lies in the computing system's ability to decide, on behalf of the user, the appropriate media and modalities for the user as the user works on his task, whether he is stationary or mobile, and as the parameters of the user's situation (e.g. noise level in the workplace) varies. Indeed, *pervasive multimodality* is effective if it adapts to the given user's *context of interaction* (i.e. the combined context of the user, his working environment and his computing system).

A *user task* is a general description of what a user wants to accomplish in using computing facilities (e.g. buying a second-hand car in the Internet). Usually, a task is realized with a user utilizing many *applications* (e.g. web browser, text editor, etc.). In general, there are several possible *suppliers* for each application (e.g. MS Word, WordPad, etc. as text editor). Every application has several *quality-of-service* (QoS) parameters (e.g. latency and page richness for web browser). When the application's QoS parameters are better (e.g. more frames rates per second for video), the same application consumes more *resources* (e.g. CPU time, memory and bandwidth). In a computing set-up, it is possible that computing resources may not be available (e.g. downloading a file may take a long time due to bandwidth constraints), hence when there is constraint in computing resources, an *automated reconfiguration* of QoS parameters of applications needs to be made so that the abundant resources are consumed while the scarce resource is freed. When situation returns to normal, in which resources are not constrained, the QoS configurations of these applications return to normal as well.

In this research work, decisions need to be made as to which media devices and modalities suit a given interaction context as well as which QoS configurations need to be made when resource constraints exist. Each of these variations in context constitutes an event. In this work, the pre-condition of an event (also called pre-condition scenario) is the given context of interaction while the resulting output of such event (called post-condition scenario) will be the selection of media and modalities and the resulting QoS configuration of applications.

In summary, two paradigms or models were made to demonstrate the infrastructure of a pervasive multimodal multimedia computing system, namely:

1. **A paradigm for interaction context-sensitive pervasive multimodality** – in this sub-system, when a specific instance of interaction context is given, the system determines the most appropriate modalities as well as their supporting media devices.

2. **A paradigm for interaction context-sensitive pervasive user task** – in this sub-system, the system reconfigures the QoS parameters of the applications based on the constraints in computing resources.

Statement of Research Problem

Nowadays, more and more of computing systems integrate dynamic components in order to respond to new requirements of adaptability, based on the evolution of context, internal failures and the deterioration of quality. This requirement could not be truer than in the case of multimodal interface which must take into account the context of application. Multimodality is favourable in its adaptation to various situations and on varying user profiles. If the environment is noisy, for example, the user has, within his disposition, various modes for data entry. If the complex data needs to be reconstituted, the system may complete an audio message with text messages or graphics. Multimodality is also favourable in appropriating various computing tools on people having temporary or permanent handicap. Multimodal interfaces are crucial in developing access to information in mobile situations as well as on embedded systems. With the novel norms of radio diffusion of information, such as GPRS (General Packet Radio Services), UMTS (Universal Mobile Telecommunications System), WiFi (Wireless Fidelity) and BlueTooth, more and more people would be connected in permanence. The mobile usage has never been more reinforced.

The dynamic configuration of multimodal multimedia architectures is a method that satisfies the important conditions in multimodal architecture in terms of improved interaction in order to render it more precise, more intuitive, more efficient and adaptive to different users and environments. Here, our interest lies in the system's adaptation, via dynamic reconfiguration, on a much larger context, called the user's interaction context. These so-called context-aware systems must have the capacity to perceive the user's situation in his workplace and in return adapt the system's behaviour to the situation in question without the need for explicit intervention from the user.

In this work, we focus on the means of the multimodal multimedia system's adaptation of behaviour to suit the given context of interaction with the aim that the user may continue working on his task anytime and anywhere he wishes. It is this principal contribution that we offer in this research domain where lots of interests were expended for the capture and dissemination of context without offering us profound tools and approach for the adaptation of applications on different contextual situations.

Objective and Methodology

Our objective is to develop an intelligent infrastructure that will allow the end user to do computing anytime, anywhere he wishes. The system is intelligent enough that it implicitly acts on behalf of the user to render computing possible. It detects the user's location, profile, and task, and related data, detects the user's working environment and computing system in order to offer the most appropriate modalities based on available supporting media devices. It offers reconfiguration of QoS parameters of applications in times of constraints in computing resources. Indeed, our objective is to provide a multimodal multimedia computing infrastructure that is capable of adapting to a much larger context called interaction context.

In order to attain this objective, the following approaches were conceived:

1. The paradigm that is to be developed should be *generic* in concept in order that the proposed solution can be applied to any domain of application with no or very little adjustments.

2. For the system to be adaptive to all possible instances of interaction context, it must be able to remember and learn from all previous experiences. To this extent, the invocation of *machine learning* (Mitchell 1997; Giraud-Carrier 2000; Alpaydin 2004) is inevitable.

3. For the system to be able to reconfigure its architecture dynamically to adapt to the given instance of context, the invocation of the principles of *autonomic computing* (Horn 2001; Kephart and Chess 2001; Salehie and Tahvildari 2005) is necessary.

4. The *software architecture* (Clements, Kazman et al. 2002; Clements, Garlan et al. 2003; Bachmann, Bass et al. 2005) of the multimodal multimedia computing system as it undergoes dynamic reconfiguration must be presented along with the simulation of results using various *formal specification tools*, such as *Petri Net* (Pettit and Gomaa 2004).

The following methodologies were used in the course of our research work and documentation:

1. The concept of *agent and multi-agent system* (Wooldridge 2001; Bellifemine, Caire et al. 2007) as *software architecture components* of the paradigm is used. The design of the multiagent system is layered, a design choice in order to make every system component robust with regards to the modifications and debugging made in other layers.

2. The concept of *virtual machine* was used to implement the agent that is responsible for incremental definition of interaction context and the detection of current instance of interaction context. Virtualization means the end users are detached from the intricacies and complexities of sensors and gadgets that are used to detect some parameters of interaction context (e.g. GPS to detect user location). The end user sees software which interacts on behalf of the whole machine. Programming of the virtual machine was done in Java.

3. *Specification* of dynamism among various components of the architecture was implemented using popular *specification languages* such as *Z, OCL and UML*. The formal specification of the proposed system is important in the sense that through formal specification, the system design is apparent and logical without the necessity of providing the reader with actual codes of a programming language that will be used to program the system.

4. The simulation of interaction context was done through specimen parameters. We used the *Petri Net software* (called *HPSim*) to demonstrate the dynamic detection of interaction context. Although the concept of interaction context is that it can grow with as many parameters as the user may wish to include, its simulation using limited numbers of parameters is essential only to prove that our ideas and concepts are correct and functional.

5. Mathematical equations and logical specifications were formulated to support various concepts and ideas within this thesis. This renders the presented ideas clearer from the mathematical and logical points of view.

Organization of the Thesis

The organization of this thesis is as follows:

The first chapter is a review of the literature whose goal is to illustrate the contributions of previous researchers' works with regards to our work as well as to differentiate ours with them, therefore illustrating our contributions to the domain. The three chapters that follow are published works, the first two in journals of international circulation while the last one is published as a book chapter.

The second chapter is an article that was published in the Research in Computing Science Journal:

Hina, M. D.; Tadj, C.; Ramdane-Cherif, A.; Levy, N., "*Towards a Context-Aware and Pervasive Multimodality*", Research in Computing Science Journal, Special Issue: "Advances in Computer Science and Engineering", Vol. 29, 2007, ISSN: 1870-4069, Mexico.

In this article, we presented the major challenges in designing the infrastructure of context-aware pervasive multimodality. We presented our proposed solutions to those challenges. We presented machine learning as a tool to build an autonomous and interaction context-adaptive system. We also demonstrated one fault-tolerant characteristic of the proposed system by providing the mechanism that finds a replacement to a failed media device.

The third chapter is an article that was published in the Journal of Information, Intelligence and Knowledge in 2008:

Hina, M. D.; Ramdane-Cherif, A.; Tadj, C.; Levy, N., "*Infrastructure of a Context Adaptive and Pervasive Multimodal Multimedia Computing System*", Journal of Information, Intelligence and Knowledge, Vol. 1, Issue 3, 2008, pp. 281-308, ISSN: 1937-7983.

In this article, we review the state of the art and noted the absence of research in the domain of pervasive multimodality. We proposed an infrastructure that will serve this needs and present our proposed solutions on the selection of optimal unimodal/multimodal interface which takes into account the user's preferences. Sample cases were cited as well as the conceived solutions to the given cases.

The fourth chapter is an article that was published as a chapter in the book "Autonomic Communication", published by Springer in 2009:

Hina, M. D.; Tadj, C.; Ramdane-Cherif, A.; Levy, N., "*Autonomic Communication in Pervasive Multimodal Multimedia Computing System*", a chapter in the book "Autonomic Communication", Vasilakos, A.V.; Parashar, M.; Karnouskos, S.; Pedrycz, W. (Eds.), 2009, XVIII, pp. 251- 283, ISBN: 978-0-387-09752-7.

In this article, we presented the communication protocols to realize autonomic communication in a pervasive multimodal multimedia computing system. The adoption of layered virtual machine to realize incremental interaction context is also demonstrated. The article also presented the rules and schemes in prioritizing and activating media devices, and the system's adaptation in case of failed devices. The system also adapts seamlessly in the event that a new media device is introduced for the first time into the system.

Finally, the fifth chapter is devoted in the conclusion of this thesis document. In this chapter, we expound on what we have contributed in this domain of research with regards to advancing the interest of pervasive multimodality and the adaptation of a multimodal computing system with regards to all the possible variations that may take place in the user's interaction context.

CHAPITRE 1

REVIEW OF THE STATE OF THE ART AND OUR INTERACTION CONTEXT-ADAPTIVE PERVASIVE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM

In this chapter, we present the previous research works that were related to ours and thereafter, with our objectives on hand, we build the infrastructure of the interaction context-adaptive pervasive multimodal multimedia computing system. Whenever there is a need to diffuse confusion, we will define the terminologies used in this research work to diminish ambiguity that may arise in the discussion.

1.1 Definition and Elucidation

Given that many terms used in this research work may elicit multiple meanings and connotations, it is in this light that we provide the correct definitions of these terms as they are used in this work. Afterwards, after we have given our own definition to the term in question, we proceed on elucidating the concepts for further clarification.

1.1.1 Pervasive or Ubiquitous Computing

We take the original definition of pervasive or ubiquitous computing in the 1990's from where it all begun, Mark Weiser (Weiser 1991; Weiser 1993). Ubiquitous computing is meant to be the *third wave in computing*. The first wave refers to the configuration of many people, one computer (the mainframes), the second wave being one person, one computer (PC). The third wave of computing – the ubiquitous computing – is a set-up wherein *computer is everywhere* and available throughout the physical environment, hence one person, many computers (Satyanarayanan 2001).

Ubiquitous computing also refers to the age of "*calm* technology" (Weiser and Brown 1996), when technology recedes into the background of our lives. In notion in pervasive computing is (1) that the purpose of a computer is to help user to do something else, (2) that the computer is a quiet, invisible servant, (3) that as the user uses intuition, he becomes smarter and that computer should use the user's unconscious, and (4) that the technology must be calm, informing but not demanding user's focus and attention.

In the context of this thesis, the notion of *pervasive computing* (Grimm, Anderson et al. 2000; Garlan, Siewiorek et al. 2002) is to be able to realize an infrastructure wherein it is possible for the user to continue working on his computing task anytime and anywhere he wishes (Hina, Tadj et al. 2006).

1.1.2 Context and Context-Aware Computing

The term "context" comes in many flavours, depending on which researcher is talking. Here we listed some of these definitions and take ours.

In Shilit's early research, (Schilit and Theimer 1994), context means the answers to the questions "*Where are you?*", "*With whom are you?*", and "*Which resources are in proximity with you?*" He defined context as the changes in the physical, user and computational environments. This idea is taken later by Pascoe (Pascoe 1998) and Dey (Dey, Salber et al. 1999). Brown considered context as "*the user's location, the identity of the people surrounding the user, as well as the time, the season, the temperature, etc*." (Brown, Bovey et al. 1997). Ryan defined context as the environment, the identity and location of the user as well as the time involved (Ryan, Pascoe et al. 1997). Ward viewed context as the possible environment states of an application (Ward, Jones et al. 1997). In Pascoe's definition, he added the pertinence of the notion of state: "*Context is a subset of physical and conceptual states having an interest to a particular entity*". Dey specified the notion of an entity: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*" (Dey 2001). This definition became the basis for Rey and Coutaz to coin the term interaction context: "*Interaction context is a combination of situations. Given a user U engaged in an activity A,* *then the interaction context at time t is the composition of situations between time t_0 and t in the conduct of A by U*" (Rey and Coutaz 2004).

We adopted the notion of "*interaction context*", but define it in the following manner: An interaction context, IC = $\{IC_1, IC_2, \ldots, IC_{max}\}$, is a set of all possible parameters that describe the given interaction context of the user. 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 in the conduct of the user's activity. Each variable is a function of the application domain. Formally, an IC is a tuple composed of a specific user context (UC), environment context (EC) and system context (SC).

A context-aware system is, by the very definition, one that is aware of its context. As a consequence of being aware, the system reacts accordingly, performing a context-triggered reconfiguration and action.

1.1.2.1 Context-Triggered Reconfiguration

Reconfiguration is the process of *adding new components*, *removing existing components* or *altering the connections between components*. Typical components and connections are servers and their communication channels to clients. However reconfigurable components may also include loadable device drivers, program modules, hardware elements, etc. In the case of an interaction context-aware system as applied in the domain of multimodality, the reconfiguration would be the addition, removal or alteration of the appropriate modalities, media devices, and configuration of QoS parameters as a function of their consumption of computing resources and user preferences.

1.1.2.2 Context-Triggered Actions

Context-triggered actions are simple IF-THEN rules used to specify how context-aware systems should adapt. Information about context-of-use in a condition clause triggers consequent commands, something like a rule-based expert system. A context-aware system is similar to contextual information and commands, except that context-triggered action commands are invoked automatically according to previously specified or learned rules. In the case of a pervasive multimodal computing system, the simple IF-THEN becomes cascaded IF-THEN-ELSE rules that continue to be in effect as long as the user is logged into the system. A change in the value of a single context parameter is sufficient enough for the system to trigger an action or a configuration. For example, when the environment becomes noisy – noisy enough that the added noise will render input vocal data to be corrupted – the corresponding reconfiguration is the shutting down of the vocal input modality. As a consequence, the next action would be the detection of which input modality should be activated in place of the vocal input modality. This alone would constitute a series of succeeding actions and reconfigurations.

1.1.3 Multimodality and Multimedia

Multimodal interaction provides the user with multiple modes of interfacing with a computing system. Multimodal user interfaces are a research area in human-computer interaction (HCI). In the domain of multimodal interfaces, two groups have emerged – the multimodal input and the multimodal input and output.

1.1.3.1 Multimodal Input

The first group of multimodal interfaces combine various user input modes, beyond the usual keyboard and mouse input/output, such as speech, pen, touch, manual gestures, gaze and head and body movements. The most common such interface combines a visual modality (e.g. a display, keyboard, and mouse) with a voice modality (speech recognition for input, speech synthesis and recorded audio for output). However other modalities, such as pen-based input or haptic input/output may be used. A sample detailed work in which mouse and speech were combined to form a multimodal fusion of input data is that of (Djenidi, Ramdane-Cherif et al. 2002; Djenidi, Ramdane-Cherif et al. 2003; Djenidi, Lévy et al. 2004).

The advantage of multiple input modalities is increased usability: the weaknesses of one modality are offset by the strengths of another. Multimodal input user interfaces have implications for accessibility. A well-designed multimodal application can be used by people with a wide variety of impairments. For example, the presentation of mathematical expressions for visually-impaired users using multimodal interface was proven to be possible and feasible by (Awdé 2009).Visually impaired users rely on the voice modality with some keypad input. Hearing-impaired users rely on the visual modality with some speech input. Other users will be "situationally impaired" (e.g. wearing gloves in a very noisy environment, driving, or needing to enter a credit card number in a public place) and will simply use the appropriate modalities as desired.

1.1.3.2 Multimodal Input and Output

The second group of multimodal systems presents users with multimedia displays and multimodal output, primarily in the form of visual and auditory cues. Other researchers also started to make use of other modalities, such as touch and olfaction. Proposed benefits of multimodal output system include synergy and redundancy. The information that is presented via several modalities is merged and refers to various aspects of the same process.

1.1.3.3 Classification of Modality

In this thesis, *modality* refers to the logical structure of man-machine interaction, specifically the mode for data input and output between a user and computer. Using natural language processing as basis, we classify modalities into 6 different groups:

1. **Visual Input (VI_{in})** – the user's eyes are used as mechanism for data entry.
2. **Vocal Input (VO_{in})** – voice or sound is captured and becomes the source of data input.
3. **Manual Input (M_{in})** – data entry is done using hand manipulation or human touch.
4. **Visual Output (VI_{out})** – data output is presented in the form as to be read by the user.
5. **Vocal Output (VO_{out})** – sound is produced as data output; the user obtains the output by listening to it.

6. **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.

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.

1.1.3.4 Media and Media Group

There are *two different meanings* of multimedia. The *first definition* is that multimedia is *media* and *content* that uses a combination of different *content forms*. The term is used to describe a medium having multiple content forms. The term is used in contrast to media which only use traditional forms of printed or hand-produced material. Multimedia includes a combination of *text, audio*, *still images*, *animation*, *video*, and interactivity content forms. The *second definition* is that of multimedia describing electronic media devices used to store and experience multimedia content.

In this thesis, we take the second definition of multimedia and refer to the individual *media* (i.e. should be "medium" if we follow "correct" English but medium in this context is rarely, possibly never, used in usual conversation) as physical device that is used to implement a modality. Regardless of size, shape, colour and other attributes, all media devices – past, present or future – can be classified based on the human body part that uses the device to generate data input and the body part that uses the device to consume the output data. Hence, our classification of media devices is as follows:

1. **Visual Input Media (VIM)** – these devices obtain user input from human sight,
2. **Visual Output Media (VOM)** – these devices generate output that is meant to be read,
3. **Audio Input Media (AIM)** – devices that use user's voice to generate input data,
4. **Audio Output Media (AOM)** – devices that output meant to be heard,
5. **Touch Input Media (TIM)** – these devices generate input via human touch,
6. **Manual Input Media (MIM)** – these devices generate input using hand strokes, and

7. **Touch Output Media (TIM)** – the user touches these devices to obtain data output

1.1.3.5 Relationship between Modalities and Media Devices

It is necessary that we build a relationship between modalities and media devices for if we find a specific modality to be suitable to the given context of interaction, it follows that the media devices supporting the chosen modality would be automatically selected and activated on the condition that they are available and functional. We will use formal specification in building 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**. This relationship is shown in Figure 1.1.

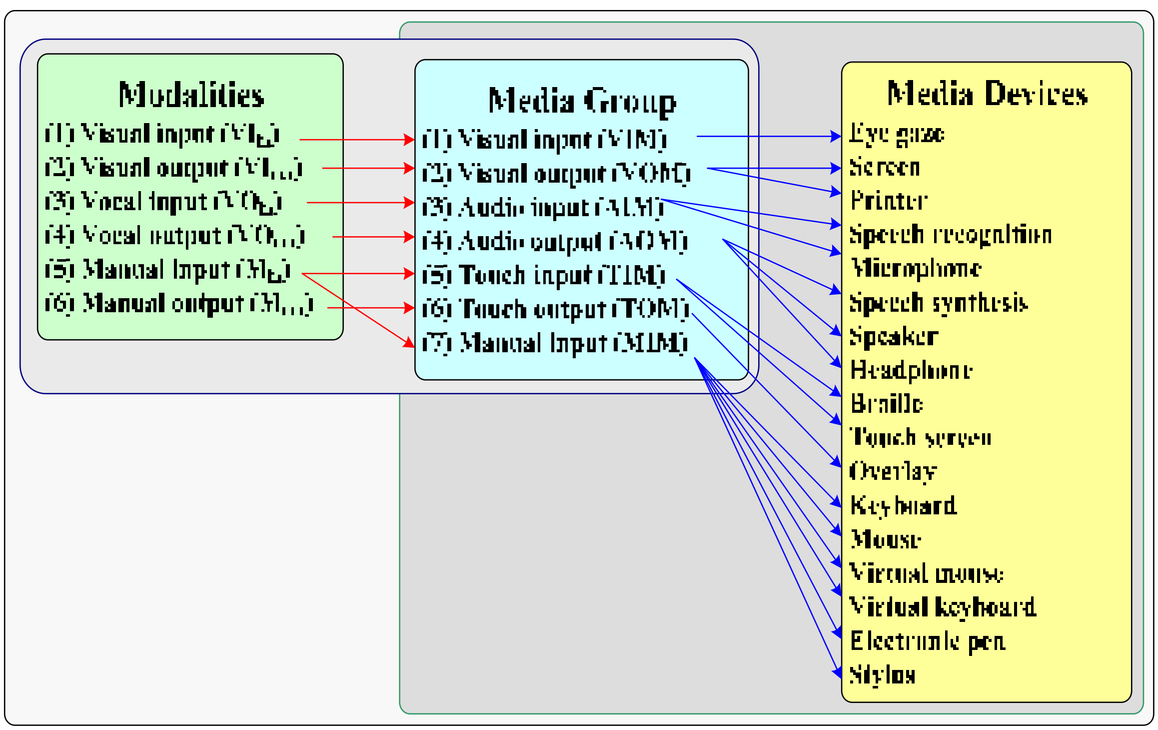


Figure 1.1 The relationship between modalities and media, and media group and media devices.

Often, there are many available devices that belong to the same media group. If such is the case then instead of activating them all, devices activation is determined through their priority rankings. To support this scheme, let there be a function $\mathbf{g_2}$ that maps a media group to a media device and its priority rank, and is denoted **g_2: Media Group → (Media Device, Priority)**. Hence sample elements of these functions are:

g_1 = {(VIin, VIM), (VIout, VOM), (VOin, OIM), (VOout, HOM), (Min, TIM), (Min, MIM), (Mout, TOM)}

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.}.

It must be noted, however, that although media technically refers to a hardware element, we opted to include a few software elements without which VO_{in} and VO_{out} modalities could not possibly be implemented. These are the *speech recognition* and *speech synthesis* software.

1.1.3.6 Ranking Media Devices

The priority ranking of media devices is essential in determining which device would be activated, by default, when a certain modality is selected as apt for a given interaction context. Here, we outline the rules for prioritizing media devices:

1. The priority ranking of media devices shall be based on the relationship **g_2: Media Group → (Media Device, Priority)** and the elements of the function **g_2**.

2. When two or more media devices happen to belong to one media group, the priority of these devices would be based on these rules:
   a. If their functionalities are identical (e.g. a mouse and a virtual mouse), activating both is incorrect because it is plain redundancy. Instead, one should be ranked higher in priority than the other. The most-commonly-used device gets the higher priority.
   b. If their functionalities are complementary (e.g. a mouse and a keyboard), activating both is acceptable and their priority is identical.

c. In case that one device is more commonly used than the other (i.e. they do not always come in pair), then the more-commonly-used one gets the higher priority.

d. If both devices always come together as a pair, then both are ranked equal in priority.

In the early stage of setting up the pervasive multimodal multimedia computing system, it is essential that the end user provides this ranking. For example, in a quiet workplace, a speaker can be the top-ranked hearing output device. In a noisy environment, however, the headphone gets the top priority. An important component that implements this priority ranking is the *media devices priority table* (MDPT). See Tableau 1.1. A MDPT is associated with every scenario.

Tableau 1.1 Sample media devices priority table (MDPT)

Media Group	Media Devices				
	Priority = 1	Priority = 2	Priority = 3	...	Priority = n
Visual Input	Eye Gaze				
Audio 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		
Audio Output	Speaker	Headphone, Speech Synthesis			
Touch Output	Braille	Overlay Keyboard			

1.2 Limitations of Contemporary Research Works

The efforts made in defining context within the domain of context awareness were in fact attempts in formalism, as in the case of definition proposed in (Abowd and Mynatt 2000). Other researchers, not satisfied with general definitions, attempted to define context formally

(Chen and Kotz 2000; Dey 2001; Prekop and Burnett 2003). Pascoe (Pascoe 1998) and Dey (Dey and Abowd 1999) brought more precision in context definition by specifying that context is a set of information that describes an entity and that an entity may be a person, a place or an object that is relevant in the interaction between the user and an application and that the entity itself may include the user and the application themselves.

In other works related to sensitivity to context, various researchers started resolving the issue concerning the user's mobility. Then, research deepens within their emphasis on the whereabouts of the user. For example, Teleporting (Bennett, Richardson et al. 1994) and Active Map (Schilit and Theimer 1994) are few works on applications that are sensitive to the geographic location of a user. Dey (Dey 2001) and Chen and Kotz (Chen and Kotz 2000) made constraints on context research by putting emphasis on applications, the contextual information that is being used and their use. Gwizdka (Gwizdka 2000) identified two categories of context: internal and external. The categorization, however, was done with respect to the user's status. Dey and Abowd (Dey and Abowd 1999) and even Schilit (Schilit, Adams et al. 1994) categorize contextual information by levels. In the case of Dey's work, the primary level contains information that are related to the user's location, activity and time whereas with Schilit, the primary level refers to the user's environment, the physical environment and the computing environment. One more time, the contextual information considered in these categorizations did not sufficiently take environment context in depth. To respond to the problems raised in the previous categorizations, Razzaque (Razzaque, Dobson et al. 2005) proposed a finer categorization of contextual information. Dey's Context Toolkit (Dey, Salber et al. 2001) is one of the first architectures which considered three (3) important steps in works on context sensitivity (that is, the *capture*, *representation* and *exploitation of context*). In this architecture, the modeling of context uses an approach called sets of pairs of (entity, attribute).

Other approaches in context representation used RDF (Resource Description Framework) which is an extension of W3C CC/PP (World Wide Web Consortium Composite Capabilities/Preferences Profile) as in the work proposed by (Held 2002) and (Indulska, Robinson et al. 2003). *Ontology* was also used in context modeling in which approach context is considered as a set of entities having aspects describing its characteristics (Strang and Linnhoff-Popien 2003).

After modeling and storage, context needs to be disseminated to the application. Here, we draw our attention to the conceptual platforms of the architectural aspects of systems that are sensitive to context (Dey, Salber et al. 2001; Kindberg and Barton 2001). The works of (Indulska, Loke et al. 2001) and (Efstratiou, Cheverst et al. 2001) present service platforms related to providing necessary services to the user based on a given context. The interoperability environments dealing with the resolution of problem related to heterogeneity and mobility of a user are presented in (DeVaul and Pentland 2000) and (Eustice, Lehman et al. 1999). Other works were oriented towards the development of distributed applications which deals with the conception of physical and logical infrastructure in developing distributed systems as in the case of works presented in (Banavar, Beck et al. 2000) and (Esler, Hightower et al. 1999). After the publication of the work of Weiser on distributed information systems (Weiser 1993), various works on context sensitivity in this genre of application has allowed the development of *ParcTab* (Schilit, Adams et al. 1993; Want, Schilit et al. 1995), *Mpad* (Kantarjiev, Demers et al. 1993), *LiveBoard* (Elrod, Bruce et al. 1992) and other interesting works (Dey 2001; Kephart and Chess 2001). The *Active Badge* project (Want, Hopper et al. 1992) of Olivetti Research and the *InfoPad* project (Truman, Pering et al. 1998) of Berkeley also embraced this axis of research on distributed computing, as in the case of other various centers of excellence, such as the Carnegie Mellon University (CMU_CS 2010), IBM (Horn 2001; Kephart and Chess 2001) and Rutgers (CS_Rutgers 2010), just to cite a few. We also note the works of (Kantarjiev, Demers et al. 1993), (Want, Schilit et al. 1995) and (Garlan, Siewiorek et al. 2002) which are some of the contributions in the research on adaptations of distributed applications on based on the given context. Also, an important work on the taxonomies of input devices include that of (Buxton 1983).

In conclusion, in the existing context-sensitive applications, very large efforts were expended by researchers in defining how to capture context and then disseminate it to the system. And yet, precise answer is still missing as to how the *application itself will adapt to the given context*. It is in this last direction that this thesis work registers.

1.3 Contribution – The Interaction Context-Aware Pervasive Multimodal Multimedia Computing System

Our general objective is to build a paradigm of dynamic reconfiguration of a multimodal multimedia architecture that will take into account the user's context of interaction. Towards this end, we have proposed automation solutions which will reinforce the system's adaptability to the user's situation as well as to support system decision in general, and multimodal multimedia in particular. These proposed solutions refer to the following propositions:

1. An automated mechanism for the selection of modalities and supporting media devices that suit the given context of interaction. This pertains to finding optimal configuration and quantifying it.

2. An automated mechanism for the selection of applications and the most suitable configuration to the user's context of interaction.

The diagram demonstrating these proposed solutions is shown below (see Figure 1.2).

In this proposed system, the selection of optimal configuration is based on a compromise in which we take into consideration the constraints related to the user, his material environment, software and other factors. This contextual information represents the *context of interaction of the user*. Such context of interaction is the combination of situations that exist while the user undertakes an activity. These situations are real-time, those that exist from the time the user starts working on a task up to the time of its completion. During the execution of this activity, some situations remain stable while others change or evolve as time passes by. Briefly, the *context of interaction* is made up of the *context of the user*, of his *environment* and of his *computing system*. A change in the context of interaction may result in the modification of appropriate modalities (and therefore of the media devices that support the modalities). We examined how an ever changing context of interaction affects the stability of the multimodal multimedia computing system so as it will continue providing services to the user. We validated our approach through specifications as well as simulations using stochastic Petri Net.

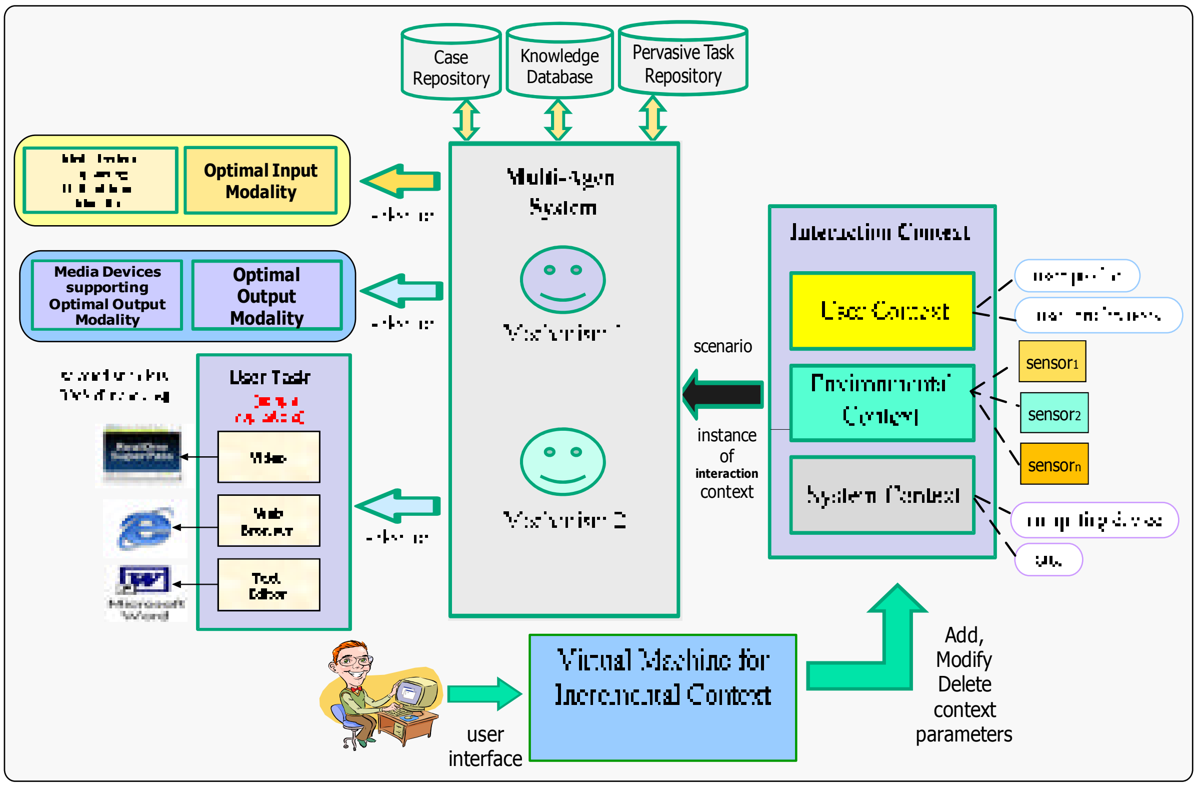


Figure 1.2 The overall structure of our proposed multimodal multimedia computing system.

The discussion that will follow discusses the architectural framework, and the design of the two mechanisms of the system's adaptation to the given context of interaction.

1.3.1 Architectural Framework

Before expounding on the development of the two mechanisms cited above, we first present the multi-agent architectural structure of our interaction context-sensitive multimodal multimedia computing system. Our objective is to come up with mechanisms for adaptability of the multimodal system with respect to instances of interaction context in a ubiquitous computing environment.

The main components of our computing system are shown in Figure 1.3. The emphasis in this diagram is focused on the different agents that comprise the system. As can be easily seen, our proposed system is a multi-agent system. The functionalities of these components (i.e. agents) are as follows:

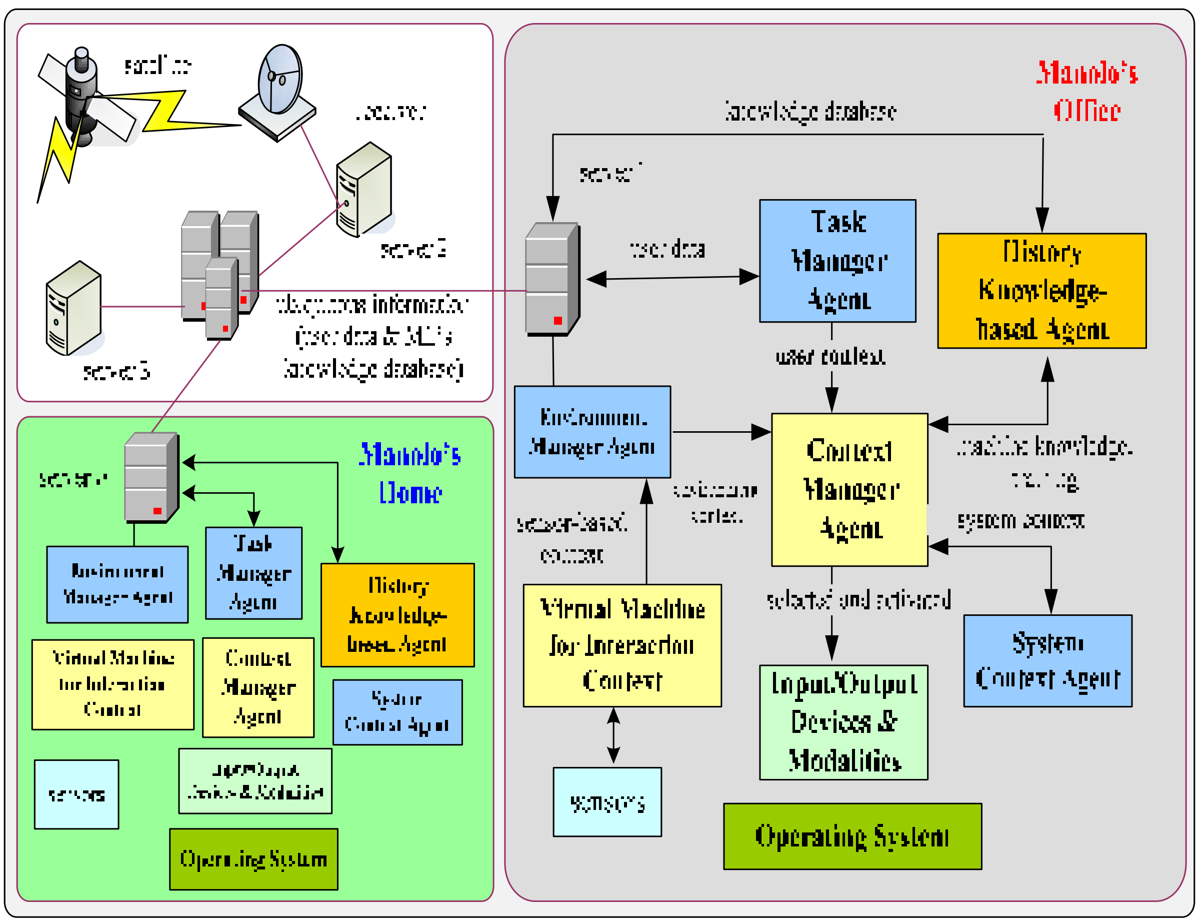


Figure 1.3 Architecture of interaction context-sensitive pervasive multimodal multimedia computing system.

- **The Task Manager Agent (TMA)** – manages user's profile, task and related data and their deployment from a server to the user's computing device, and vice versa.

- **The Context Manager Agent (CMA)** – detects interaction context taken from sensors and user profile, environment and computing system and select the modality and its supporting media devices that are most suitable to the given context.

- **The History and Knowledge-based Agent (HKA)** – responsible for machine learning training and knowledge acquisition.

- **The Virtual Machine for Interaction Context (VMIC)** – detects sensor-based context and allows the incremental definition of context by considering one context parameter at a time.

- **The Environmental Manager Agent (EMA)** – detects available and functional media devices in the user's environment.

- **The System Context Agent (SCA) –** detects the status of available computing devices and computing resources (e.g. bandwidth, CPU, memory and battery).

As shown in the diagram, a user (i.e. Manolo) may work at home, logs off and later reconnects to a computing device in order to continue working on an interrupted task whenever and wherever he wishes. Due to user's mobility, there are variations in the user's context of interaction as well as available resources; these variations are compensated by corresponding variations in the selection of modalities, activation of supporting media devices and the necessary adaptation in the configuration to execute user's task.

As shown in Figure 1.4, different parameters make up the context of interaction. The User Context Agent detects the user's context; the Environment Context Agent in coordination with VMIC agent detects the context of the user's environment and the System Context

Agent detects the available computing resources. All these parameters are consolidated and form the overall context of interaction at that particular instance.

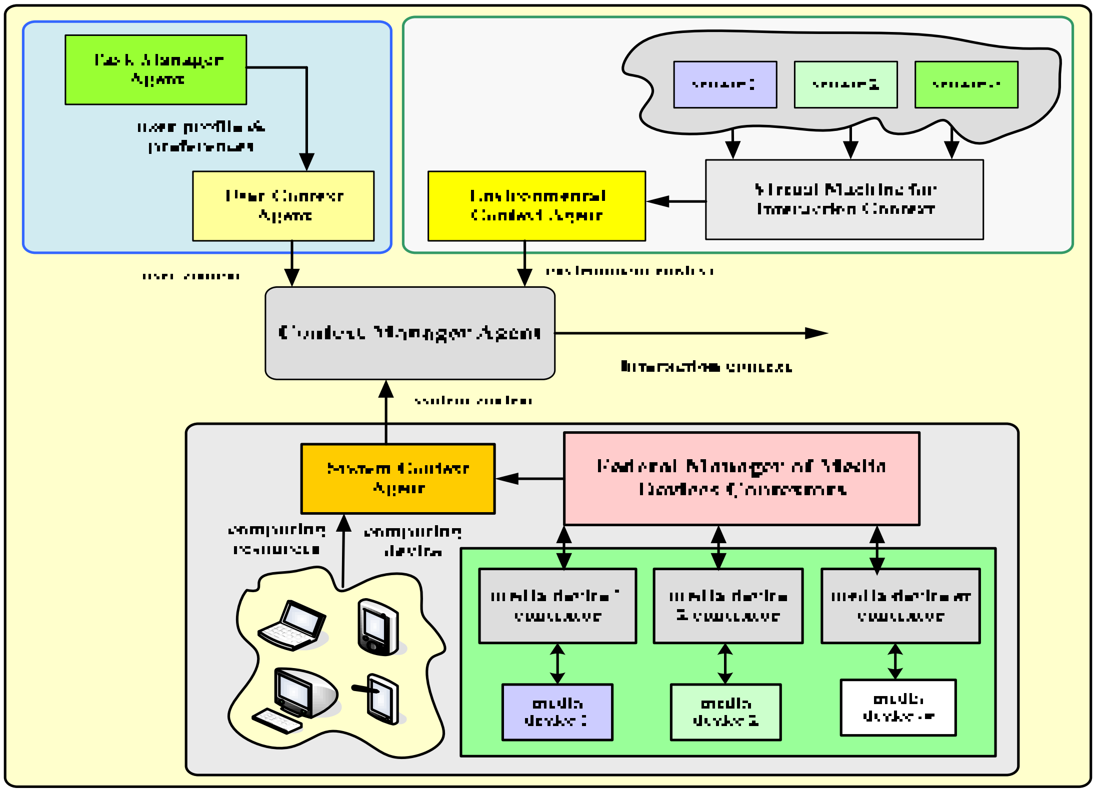


Figure 1.4 The parameters that are used to determine interaction context.

1.3.2 Attribute-Driven Architectural Design and Architectural Views

Software architecture refers to the structure of the components of a program or a software system, their interrelationships, and the principles and guidelines governing their design and evolution over time. In designing our system architecture, we use the *attribute-driven design* (ADD) (Bachmann and Bass 2001) methodology because our architectural design is aimed at achieving the system's desired quality attributes. The steps that we followed in implementing ADD were described in our previous work in IEEE CCECE '06 conference (Hina, Tadj et al. 2006). A simple level 1 data flow diagram of our system's major system components is shown in Figure 1.5.

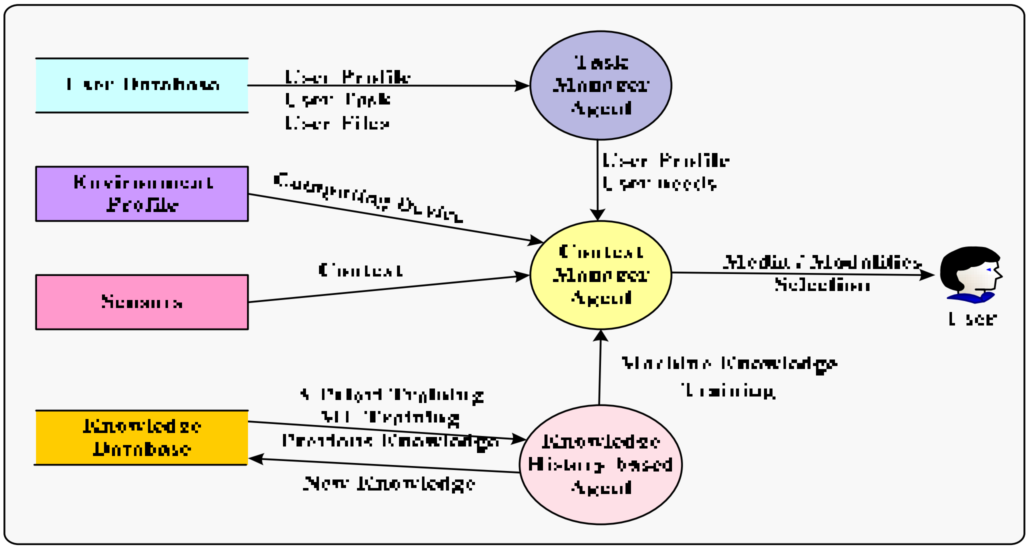


Figure 1.5 Data Flow Diagram, Level 1.

An *architectural view* is a representation of the coherent set of architectural elements. It represents a set of elements or components and their relationships. In some literature, *architectural view* and *structure* are sometimes used interchangeably. For an architectural design to demonstrate the necessary information of the interested stakeholders, one architectural view is often not enough; there needs to be two or three (or more) so that the management, analyst, programmers, end user and customer could see, understand and appreciate the architectural design based on each one's perspective.

In our pervasive multimodal multimedia computing system, the architectural views come in three (3) types, namely:

1. **Module.** The elements are generally the modules, which are the units of implementation. Modules are code-based way of considering the system. This view shows the relationship among different modules (see Figure 1.6).

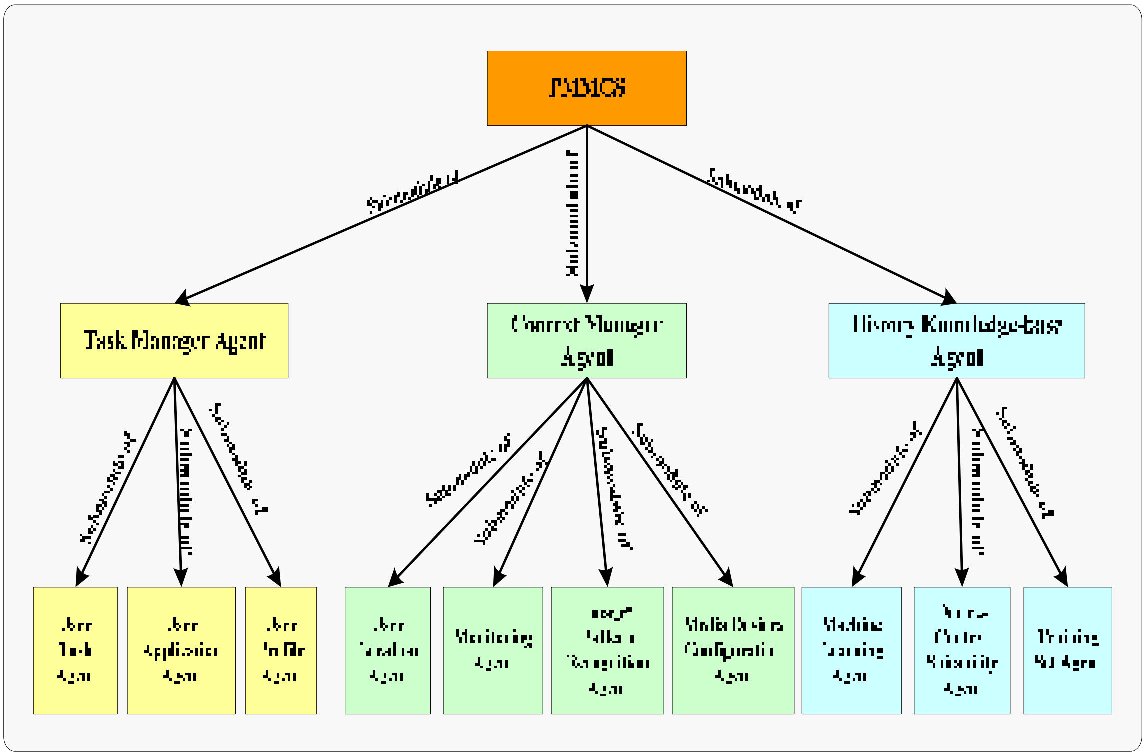


Figure 1.6 First-level Modular view (PMMCS = pervasive multimodal multimedia computing system).

2. **Component-and-Connector**. The elements are generally the runtime components (i.e. units of computations) and connectors (i.e. communication vehicle or protocol between elements). It satisfies the questions related to some shared data stores, parts of the system that are replicated, and parts of the system that run in parallel (Figure 1.7).

3. **Allocation**. This structure or view shows the relationship between software elements and the hardware or files that are created, used or executed. (See Figure 1.8). Note in the figure that we specify specific hardware and sensors; this is done to provide specimen parameters and devices. In general, this has to be interpreted as we are referring to context parameters *1*, *2*, … and so on and the processing and sensors/gadgets associated with it.

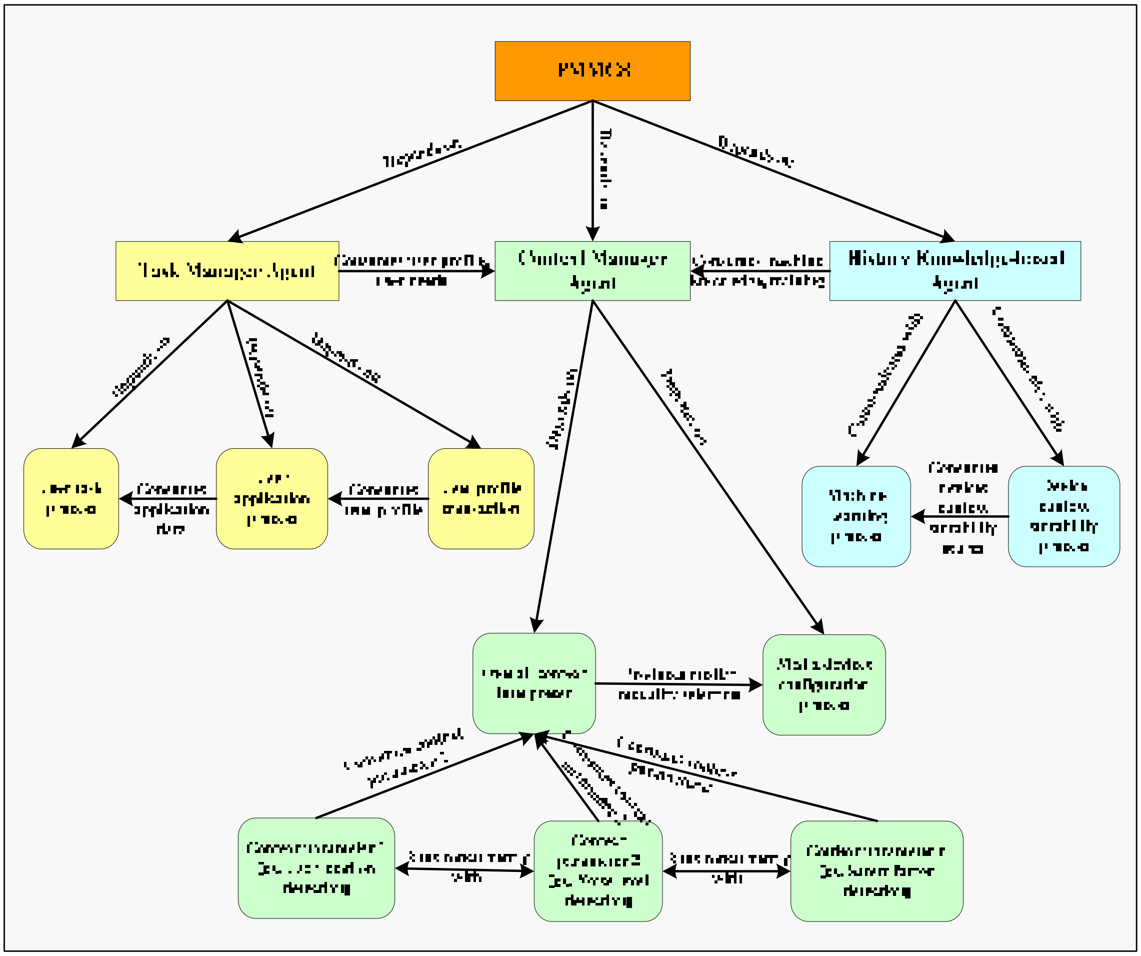


Figure 1.7 First-level component-and-connector view.

1.3.3 The Virtual Machine for Incremental User Context (VMIUC)

Given that context pertains to conditions that exist while an activity is taking place and that context itself is a subjective concept, we aim, therefore, in obtaining all the necessary parameters that will reflect the condition that the user cares about. In this regard, we believe that the definition of context, based on parameters, needs to be progressive. We believe in the end user's judgment as to what parameters are important when considering context. Hence, our system should allow the end user to have control on the parameters that constitute context (i.e. add, delete and change) – one parameter at a time – as the user sees fit and necessary. This leads us towards a context formation called *incremental context*.

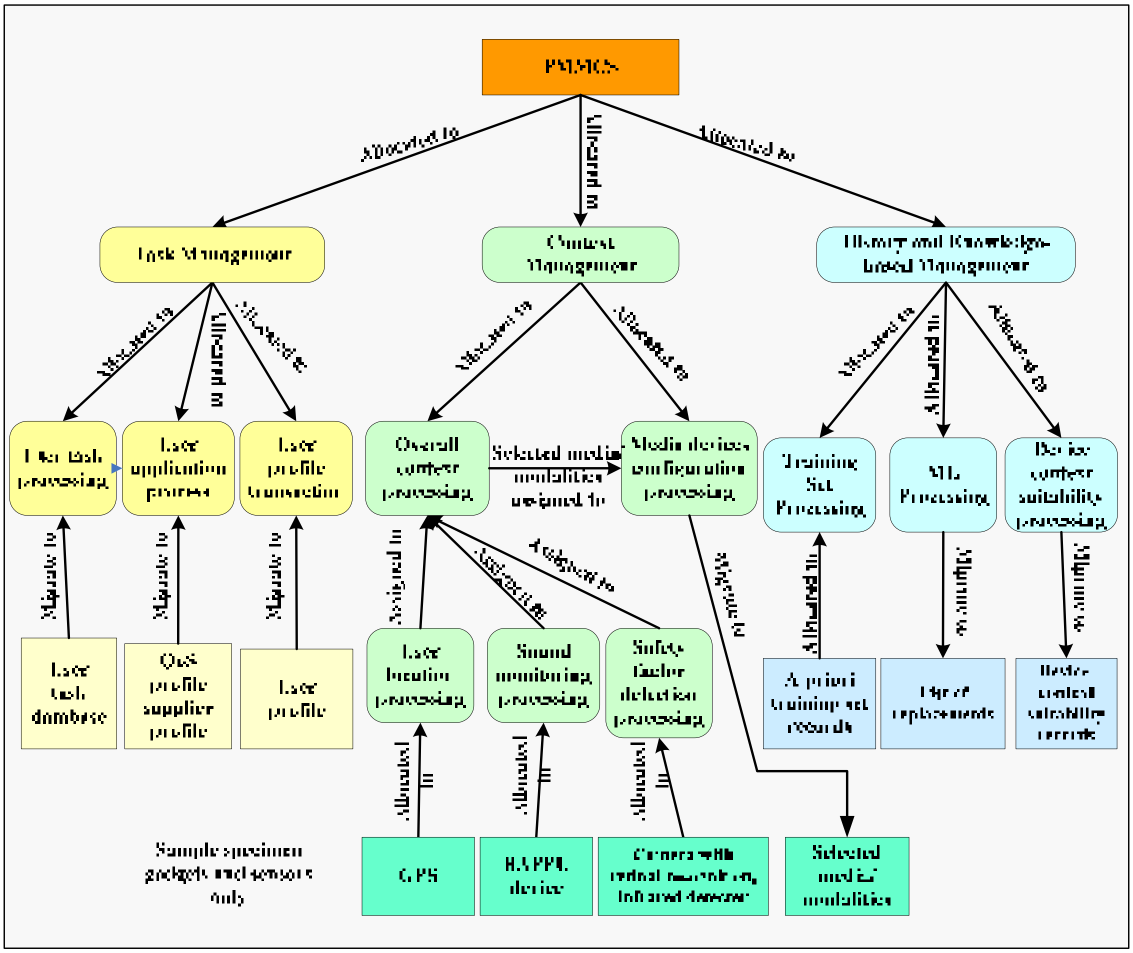


Figure 1.8 First level allocation view.

To realize *incremental context* that is *sensor-based*, meaning that certain context parameters are interpreted based on the values obtained from certain sensors, we developed the VMIUC using layered architectural approach (see Figure 1.9). These architectural layers interact with one another; specifically the layers that are adjacent with one another interact directly. Layering is a technique that is used to prevent possible cascading of errors or ripple effect whenever one wishes to debug or modify an element of a particular layer. Whenever possible, layering is chosen as a design consideration due to this benefit. Generally, in this structure, the top layer is associated with the interface interacting directly with an end user while the bottom layer is usually associated with gadgets or hardware elements.

As shown in the diagram, the VM Layer 4 acts as the human-machine interface; its "*instruction set*" are the four functions found in Layer 3 – the "*add parameter*", "*modify parameter*", and "*delete parameter*" are basic commands that manipulate the sensor-based context parameters while "*determine context*" yields the sensor-based context based on the values of currently-defined parameters. VM Layer 2 is a "*library of functions*" that collectively supports Layer 3 instructions while Layer 1 is another "*library of functions*" that acts as a link between Layer 2 and Layer 0. Layer 0 is assigned to a collection of sensors (or machines or gadgets) that generate some raw data representing the value of a certain context parameter. Each lower layer supports the upper layer by providing the results to the functions demanded by the latter. This interdependence continues top down up to the very last layer. Consequently, the transfer of resulting data is propagated bottom up (meaning from layers 0 to 4). Layers 4, 3 and 2 are robust: the functions in these layers are independent of the context parameters, and therefore could be used by any system that deals with sensor-based context. If a new parameter needs to be added, then a minor modification may be needed in the functions in Layer 1 and the probe, one that will supply raw data for a certain parameter, may be need to be installed in layer 0. For example, the interactions among the layers to add a new context parameter (i.e. Noise Level) are shown in Figure 1.10, the deletion of a context parameter in Figure 1.11 and the detection of the sensor-based context in Figure 1.12. Further details on how to add, modify and delete a context parameter as well as the detection of the current sensor-based context are provided in Chapters 3 and 4. These tools were conceived in generic fashion such that they can be used and integrated into any kind of system, independent of the system's application.

The design of a virtual machine is always to come up with an efficient, isolated duplicate of a real machine. The real machine is always complicated, difficult to understand, and its behavior is usually controlled by its designer. The aim of virtual machine is therefore to provide regular users ways of controlling and using the real machine without the necessity of having to know the intricacies of the actual machine. The end users, therefore, control the actual machine, asks for it to do something using very simple instructions.

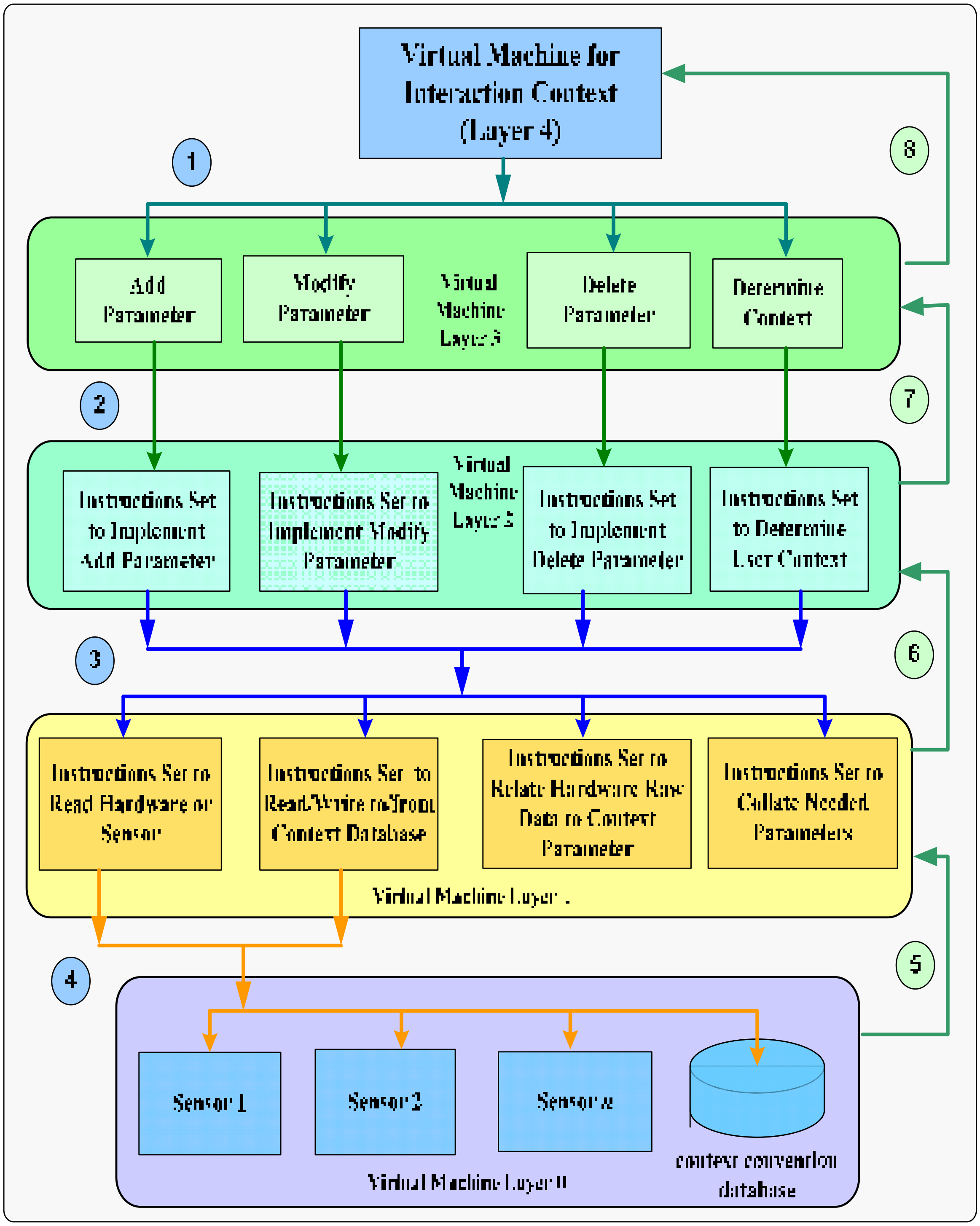


Figure 1.9 The design of a virtual machine for incremental user context.

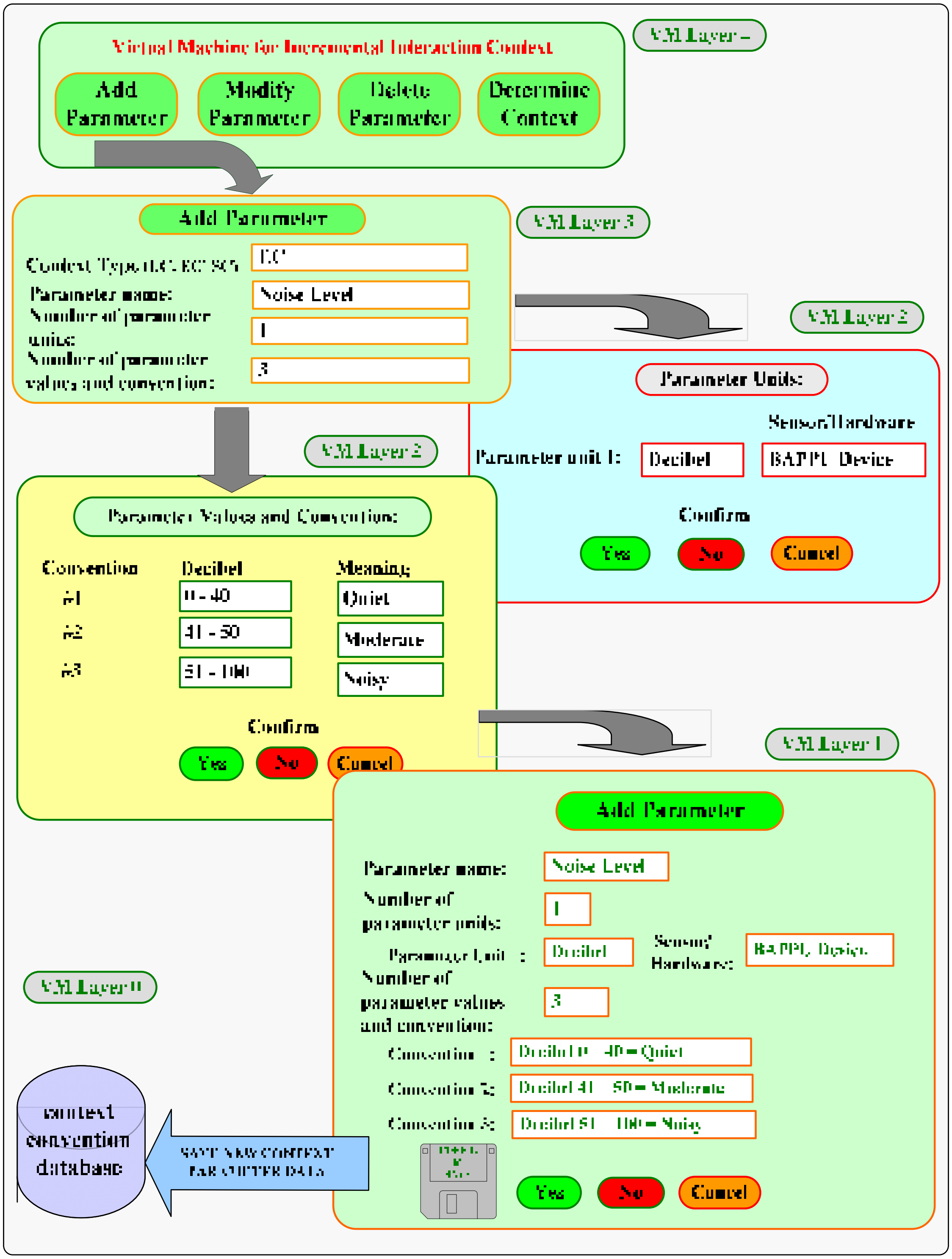


Figure 1.10 The interactions among layers to add new context parameter: "Noise Level".

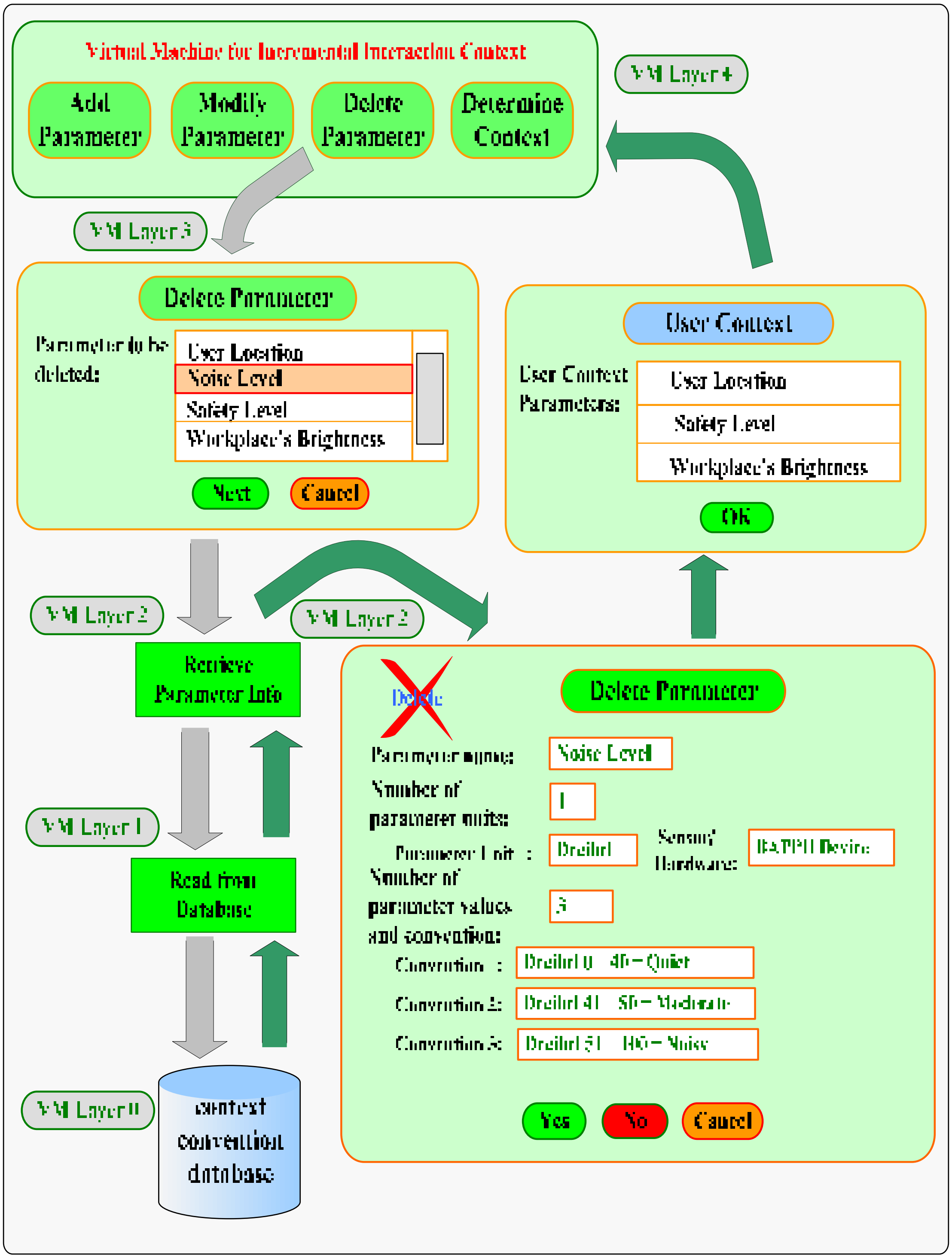


Figure 1.11 The VM layers interaction to realize "deleting a user context parameter".

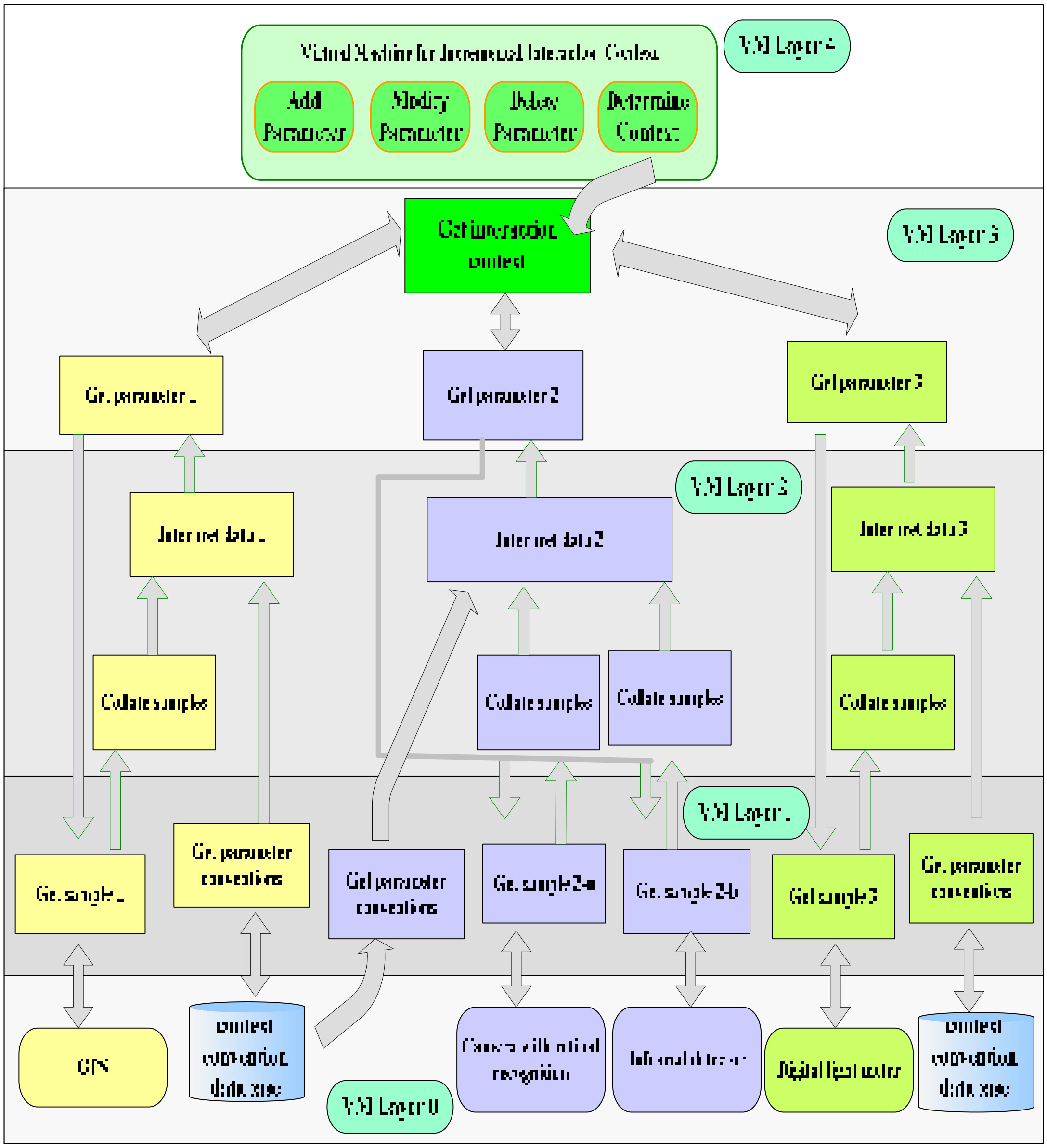


Figure 1.12 VM layers interaction in detecting the current interaction context using specimen sensors.

1.3.4 The History and Knowledge-based Agent (HKA)

As shown in Figure 1.13, HKA is the agent tasked with the selection of modalities, of supporting media devices, and of the applications configurations based on the given instance of context of interaction. Here, we discuss the concept behind HKA's knowledge acquisition. A *scenario* is the base of such knowledge.

Figure 1.13 shows the generic diagram demonstrating how knowledge is acquired by HKA. The input to the *machine learning* (ML) component (responsible for analysis) is called the *pre-condition scenario* while the resulting output is called the *post-condition scenario*. The pre-condition scenarios as well as those of the post-condition scenarios are stored in a storage called *scenario repository*. Whenever the ML component encounters a situation, it takes into account the parameters involved in the pre-condition scenario as well as consult its initial knowledge (also called *a priori* knowledge) or other previously-stored knowledge. If the pre-condition scenario is already found similar (or identical) to the one held in the scenario repository, the ML component simply takes in the corresponding post-condition scenario which is then taken as the necessary output that is bound for implementation.

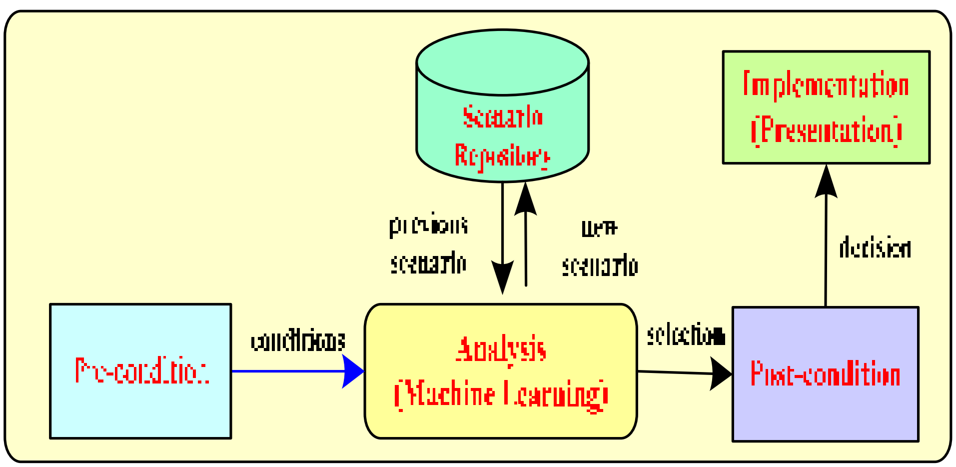


Figure 1.13 Diagram showing knowledge acquisition within HKA.

In this work, we proposed the use of *case-based reasoning with supervised learning* (CBR) as a learning tool in order to automate the system's adaptation. If no knowledge is found (meaning, the scenario is new), the ML component performs calculations using the mechanisms (i.e. mechanism 1 or 2 depending on the given case) in determining the corresponding post-condition scenario which itself afterwards will be stored in the scenario repository. With time, the ML component will accumulate enough knowledge that it will be able to "learn" almost all situations and that it will be able to react to each one of them accordingly.

Using CBR makes it possible to find scenarios of related case. The technique, however, necessitates that a case must be identified. To use this approach, we need to model a case in such a way that we will end up finding a solution to the problem on hand. As stated by Kolodner (Kolodner 1993), a *case* is always composed of the same components, regardless of the whatever application domain that is in consideration. Its components are made up of a problem, a solution and an eventual evaluation:

- **The problem** – this corresponds to the pre-condition scenario.
- **The solution** – this corresponds to the resulting post-condition scenario
- **The evaluation** – this would refer to the rate of relevance of the proposed solution.

The process of reuse consists of, for a new case, recovering a previously stored similar case, evaluate it and then store the new case onto the repository. This process is made up of the following steps:

- **Problem representation** – For every scenario that is sent to HKA, we consult an identical case or cases that are most similar to the one in question. To do this, we need to formalize the problem part as if it is a new case in order to compare it against others that are already stored in the repository.

- **Similarity calculation** – the case that is most pertinent is generally found through its similarity score with the new case. In order to do this, we come up with similarity calculation algorithm that helps in facilitating the search for similar cases.

- **Search of pertinent result** – the search is based on the highest similarity score.

- **Memorization** – the memorization is a choice that is usually left to the end user to decide since he is the most apt in deciding if the new case needs to be remembered or not. Even then, we proposed to the user to memorize his case.

Inspired by (Lajmi, Ghedira et al. 2007), we modify the similarity scoring scheme to reflect the needs of our system. Hence, given a *new case* (NC) and an individual c*ase stored in the knowledge database,* also called *memorized case* (MC), the similarity of the problem between the two cases (i.e. subscript indicates which case is considered) is equal to their similarity in their interaction context (IC) parameters. This relationship is given by:

$$Sim(IC_{NC}, IC_{MC}) = \frac{\sum_{i=1}^{NCC} Sim(IC_{i_{NC}}, IC_{MC})}{max(IC_{NCC}, IC_{MCC})} \tag{1.1}$$

where IC_{NC} and IC_{MC} are the interaction context of the new case and the memorized case, respectively. NCC and MCC are the total number of community (i.e. total number of parameters) of the new case and the memorized case, respectively. Hence, IC_{NCC} tells us the number of community (i.e. parameters) that makes up the interaction context of the new case while IC_{MCC} denotes the number of community (i.e. parameters) that makes up the interaction context of the memorized case. The term $IC_{i_{NC}}$ denotes the i^th interaction context parameter of the new case where *i* is a variable that loops from 1 to NCC. The expression max(IC_Par_{NC}, IC_Par_{MC}) takes whichever is greater between the number of parameters of NC and MC. $Sim(IC_{i_{NC}}, IC_{MC}) = max_{j=1...MCC}\, Sim(IC_{i_{NC}}, IC_{j_{MC}})$ where $IC_{j_{MC}} \in IC_{MC}$ and $Sim(IC_{i_{NC}}, IC_{j_{MC}}) \in [0, 1]$ is the similarity between parameter *i* of NC and parameter *j* of MC.

For the comparison of parameter *i* of NC against parameter *j* of MC, we need to compare how similar they are in terms of their names and values. Hence, the similarity between parameter *i* of NC and parameter *j* of MC is given by:

$$Sim(IC_{i\,NC}, IC_{j\,MC}) = Sim(IC_{i\,Name\,NC}, IC_{j\,Name\,MC}) * Sim(IC_{i\,Value\,NC}, IC_{j\,Value\,MC}) \tag{1.2}$$

The numerical value associated to the results of the comparisons of the names and values of parameters *i* of NC and *j* of MC is given below:

$$Sim(IC_{i\,Name\,NC}, IC_{j\,Name\,MC}) = \begin{cases} 0 \text{ if } IC_{i\,Name\,NC} \neq IC_{j\,Name\,MC} \\ 1 \text{ if } IC_{i\,Name\,NC} = IC_{j\,Name\,MC} \end{cases} \tag{1.3}$$

The relationship above indicates that when the parameter name of I of NC and parameter name of j of MC are different, the similarity score between the two is zero. That means, if we compare, for instance, the parameter name "temperature" against parameter name "noise level", the similarity between them is automatically zero.

If and when the two parameters have the same name, their values do matter. The relationship is given by:

$$Sim(IC_{i\,Value\,NC}, IC_{j\,Value\,MC}) = 1 - \left(\frac{\left| IC_{j\,Value\,MC} - IC_{i\,Value\,NC} \right|}{\left| IC_{j\,Value\,MC} - IC_{i\,Value\,NC} \right| + 1} \right) \tag{1.4}$$

Hence, if we compare the name of the parameter of NC against the name of the parameter of MC and both are found to be identical, say both parameters are named "noise level", then the similarity score for their names is 1 and we will then proceed to comparing their parameter values.

In this work, as can be seen in various chapters of this document, each context parameter has a name and a value, and such value (or range of values) is associated with a convention number. For example, for noise level (see Figure 1.10), if the measured or sampled value is 0 to 40 dB, we say that noise level = “quiet” (this is considered convention 1), noise level is from 41 to 50 dB, the noise level = “moderate” (this is convention 2) and noise level is 51 dB or more, then noise level = “noisy” (this is convention 3). Indeed, for every given context parameter, it is associated with one or more conventions. In our work, as a rule of thumb, if the convention number is 1, the value of the context parameter is ideal to the user. As the convention number increases in numerical value, the context parameter is beginning to shift to the unwanted condition. Example, taking into account the conventions, if noise level = 1, then the working environment is quiet, whereas if the noise level = 3, the working environment is noisy. The convention number associated with a context parameter will be used in measuring the similarity score of two parameters being compared.

If the name of the parameter in NC and the name of the parameter of MC are the same, if so we then proceed to determining their similarity score with regards to their values. Consider the following cases:

Case 1: Parameter of NC: Noise level = 1. Parameter of MC: Noise level = 1. In this case, $Sim(Name_{NC}, Name_{MC}) = 1$ and $Sim(Value_{NC}, Value_{MC})$ will be computed as follows: 1 – (1-1)|/|1-1|+1) = 1 – (0/1) = 1 – 0 = 1. They are completely the same, both in names and values.

Case 2: Parameter of NC: Noise level = 1. Parameter of MC: Noise level = 2. Again, their names are the same, $Sim(Name_{NC}, Name_{MC}) = 1$. $Sim(Value_{NC}, Value_{MC})$ will be computed as follows: 1 – (|1-2|/|1-2|+1)| = 1 – (1/2) = 0.5. The value indicates that they are quite closed enough (i.e. 50% similar with each other).

Case 3: Parameter of NC: Noise level = 1. Parameter of MC: Noise level = 3. Again, their names are the same. Sim($Value_{NC}$, $Value_{MC}$) will be computed as follows: 1 – (|1-3|/|1-3|+1)| = 1 – (2/3) = 1/3. The value indicates that they are quite far from each other and is 33.3% similar with one another.

Case 4: Parameter of NC: Noise level = 2. Parameter of MC: Noise level = 3. Again, their names are the same. Sim($Value_{NC}$, $Value_{MC}$) will be computed as follows: 1 – (|2-3|/|2-3|+1)| = 1 – (1/2) = 1/2. The value indicates that they are quite closed to each other (50% similar).

In general, the similarity value of two parameters having identical names is 1/distance between them. It means, if they have the same value, their similarity score is 1; if they are 2 values apart, their similarity score is ½. If they are 3 values apart, their similarity score is 1/3, and 1/*n* if they are *n* values apart from each other.

In our previously published articles (i.e. Chapters 2, 3 and 4), we have specified that interaction context (**IC**) is actually the composition of all the elements of user context (**UC**), environment context (**EC**) and system context (**SC**). In this respect, we made a decision that the weight of **UC**, **EC** and **SC** in the composition of **IC** is identical (meaning, **UC** = **EC** = **SC** = 1/3 of **IC**), hence the similarity formula specified in Equation 1.1 becomes:

$$Sim(NC, MC) = \frac{1}{3} Sim(UC_{NC}, UC_{MC}) + \frac{1}{3} Sim(EC_{NC}, EC_{MC}) + \frac{1}{3} Sim(SC_{NC}, SC_{MC}) \tag{1.5}$$

The similarity between the **UC** of NC vs. the **UC** of MC is given by:

$$Sim(UC_{NC}, UC_{MC}) = \frac{\sum_{i=1}^{UC_{NCC}} Sim(UC_{i_{NC}}, UC_{MC})}{max(UC_{NCC}, UC_{MCC})} \tag{1.6}$$

For the similarity measures of **EC** and **SC** of NC vs. MC, the same principle as Equation 1.3 must be applied, with the formula adjusted accordingly to denote **EC** and **SC**, respectively, yielding:

$$Sim(EC_{NC}, EC_{MC}) = \frac{\sum_{i=1}^{EC_{NCC}} Sim(EC_{i_{NCC}}, EC_{MCC})}{max(EC_{NCC}, EC_{MCC})} \quad (1.7)$$

$$Sim(SC_{NC}, SC_{MC}) = \frac{\sum_{i=1}^{SC_{NCC}} Sim(SC_{i_{NC}}, SC_{MC})}{max(SC_{NCC}, SC_{MCC})} \quad (1.8)$$

where UC_{NCC} denotes the total number of community (parameters) in the UC of NC. The same principle applies to EC_{NCC} (total number of EC parameters in NC) and SC_{NCC} (total number of SC parameters in NC). This also applies to UC_{MCC}, EC_{MCC} and SC_{MCC}. Note that the number of community (number of parameters) of a new case is equal to the sum of all of its community (i.e. parameters) in **UC**, **EC** and **SC**. That is, $NCC = UC_{NCC} + EC_{NCC} + SC_{NCC}$. Similarly, $MCC = UC_{MCC} + EC_{MCC} + SC_{MCC}$.

As shown in the above relationship, we are in the assumption that the weights of **UC**, **EC** and **SC** are equal (each is 33.3%) but this figure can be easily adjusted by the expert (i.e. end user) to suit his needs.

For simplicity of discussion, we revert back our discussion of **IC** (rather than the individual components **UC**, **EC** and **SC**) to simplify the issue of finding scenarios that are similar to the new case in consideration. The formula given in Equation 1.1 is used to compare its similarity against other scenarios that are within its "*neighborhood*". Those considered *neighbors* are actually scenarios that are very close to the case in question with respect the values of its interaction context parameters. Using these parameters, we can identify the index of the case in question within the database of scenarios. We call this index "*scenario number*" or *scenNum*. For example, if scenario *i* is composed of individual interaction context parameters IC_1, IC_2, …, IC_{max}, that is $\mathbf{IC_i} = (IC_1, IC_2, \ldots, IC_{max})$, the scenario index assigned to this specific case is given by:

$$ScenNum = IC_{max} + \sum_{i=1}^{max-1}((IC_i - 1) \bullet \prod_{j=i+1}^{max-1} card(IC_j)) \tag{1.9}$$

where $card(IC_j)$ is the cardinality $\bullet$ total number of values for the convention of a specific interaction context parameter *j*.

In the discussion that follows, we will elaborate on the learning and adaptation mechanisms that are used by the HKA agent.

1.3.5 Mechanism/Paradigm 1: Selection of Modalities and Supporting Media Devices Suitable to an Instance of Interaction Context

Let *interaction context*, $\mathbf{IC} = \{IC_1, IC_2, \ldots, IC_{max}\}$, be the 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:

$$IC_i = UC_k \otimes EC_l \otimes SC_m \tag{1.10}$$

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 describing the state of the user during his activity. A specific user context k is given by:

$$UC_k = \bigotimes_{x=1}^{max_k} IC_{kx} \tag{1.11}$$

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

$$EC_l = \bigotimes_{y=1}^{max_l} IC_{ly} \tag{1.12}$$

$$SC_m = \bigotimes_{i=1}^{max_m} SC_i \tag{1.13}$$

The first knowledge that the ML component must learn is to relate the context of interaction to appropriate modality. Let function s_1**:** $\mathbf{IC} \rightarrow \mathbf{M}$ maps interaction context with appropriate modalities. This function takes an instance of $\mathbf{IC} = \{IC_1, IC_2, \ldots, IC_{max}\}$ as pre-condition scenario input to HKA and as a result returns a set of optimal modalities $\mathbf{M_o} = \{m_1, m_2, \ldots, m_{max}\}$ as post-condition output.

Let $\mathbf{M} = \{VI_{in}, VO_{in}, M_{in}, VI_{out}, VO_{out}, M_{out}\}$ be the set of modalities. Modalities are possible when the following condition holds:

$$Modality\ Possible = (VI_{in} \vee VO_{in} \vee M_{in}) \wedge (VI_{out} \vee VO_{out} \vee M_{out}) \tag{1.14}$$

Consequently, modality fails under the following condition:

$$Modality\ Failure = ((VI_{in} = Failed) \wedge (VO_{in} = Failed) \wedge (M_{in} = Failed)) \vee ((VI_{out} = Failed) \wedge (VO_{out} = Failed) \wedge (M_{out} = Failed)) \quad (1.15)$$

wherein symbols $\wedge$ and $\vee$ represent logical AND and OR, respectively.

Let $\mathbf{M_j}$ = element of the power set of **M**, that is, $\mathbf{M_j} \in \mathcal{P}(\mathbf{M})$ where $1 \leq j \leq mod_{max}$ (maximum modality). Also, let $\hat{M}$ = the most suitable $\mathbf{M_j}$ for a given interaction context IC_i. This set is given by the following relationship:

$$\hat{M} = arg \max_{j} P(M_j \mid IC_i) \quad (1.16)$$

To simplify calculation, *Bayes Theorem* (Kallenberg 2002), given below, can be adopted, and $P(M_j/IC_i)$ becomes:

$$P(M_j|IC_i) = \frac{P(IC_i|M_j) \times P(M_j)}{P(IC_i)} \quad (1.17)$$

The implementation of Bayes Theorem leads to the *Naive Bayes algorithm* (Mitchell 1997). The Naive Bayes algorithm is a classification algorithm that assumes that the IC_i attributes $IC_1,\ldots,IC_{max}$ are all conditionally independent of one another given a post condition M_j. The representation of $P(IC_i|M_j)$ becomes:

$$\begin{aligned} P(IC_i \mid M_j) &= P(IC_1,\ldots,IC_{max} \mid M_j) \\ &= P(IC_1 \mid M_j) \times \ldots \times P(IC_{max} \mid M_j) \\ &= \prod_{i=1}^{max} P(IC_i \mid M_j) \end{aligned} \quad (1.18)$$

Here, our goal is to train a classifier that, given a new IC_i to classify, will provide the probability distribution on all possible values of **M** (i.e. M_1, M_2, …, M_m). Given that IC_i = (IC_1, IC_2, ...,IC_{max}), then equation above becomes:

$$P(M_j \mid IC_1 ... IC_{max}) = \frac{P(M_j) P(IC_1 ... IC_{max} \mid M_j)}{\sum_{k=1}^{m} P(M_k) P(IC_1 ... IC_{max} \mid M_k)} \tag{1.19}$$

The above equation can also be written as:

$$P(M_j \mid IC_1 ... IC_{max}) = \frac{P(M_j) \prod_{i=1}^{max} P(IC_i \mid M_j)}{\sum_{k=1}^{m} P(M_k) \prod_{i=1}^{max} P(IC_i \mid M_k)} \tag{1.20}$$

which is the fundamental equation for the Naive Bayes classifier. Given a *new* instance of interaction context IC_i = (IC_1, …, IC_{max}), the equation shows how to calculate the probability that M_j will take given the observed attribute values of IC_i and given that the distributions $P(M_j)$ and $P(IC_i|M_j)$ are estimated values taken from training data (SR). If we are interested only in the *most suitable value* of M_j, then we have the Naive Bayes classification rule:

$$\hat{M} = \arg\max_j \left(\frac{P(M_j) \prod_{i=1}^{max} P(IC_i \mid M_j)}{\sum_{k=1}^{m} P(M_k) \prod_{i=1}^{n} P(IC_i \mid M_k)} \right) \tag{1.21}$$

Given that the denominator does not depend on parameter *j*, then the above equation becomes

$$\hat{M} = \arg\max_j \left(P(M_j) \prod_{i=1}^{max} P(IC_i \mid M_j) \right) \tag{1.22}$$

Given that IC_i is composed of elements of UC, and SC, then P(ICi/Mj) can be written as:

$$P(IC_i \mid M_j) = P(UC_k \mid M_j) \times P(EC_l \mid M_j) \times P(SC_m \mid M_j) \tag{1.23}$$

Note that in IC_i, the parameters $IC_1, \ldots IC_{max}$ are mutually exclusive. Using Equation 1.11, Equation 1.12 and Equation 1.13, we replace each element of IC_i with corresponding value. For example, the relationship involving user context UC_k with respect to modality $\mathbf{M_j}$ is given by:

$$P(UC_k \mid M_j) = \prod_{i=1}^{uc_{max}} P(UC_i \mid M_j) \tag{1.24}$$

In conclusion, the *optimal modality* for whatever instance of interaction context is given by:

$$\hat{M} = \arg\max_j \left(\left(\prod_{k=1}^{uc_{max}} P(UC_k \mid M_j) \times \prod_{l=1}^{ec_{max}} P(EC_l \mid M_j) \times \prod_{m=1}^{sc_{max}} P(SC_m \mid M_j) \times P(M_j) \right) \right) \tag{1.25}$$

where P(**Mj**) = frequency count of **Mj** in scenario repository (SR) • cardinality of SR and uc_{max}, ec_{max} and sc_{max} are, respectively, the total number of **UC** parameters, **EC** parameters and **SC** parameters with the interaction context being considered.

To illustrate the usage of the derived equation above, let us consider, for example, an interaction context that is composed of the following parameters: **IC** = (IC_1, IC_2, IC_3, IC_4, IC_5, IC_6, IC_7) wherein

- IC_1 = {true | false} = if user is manually disabled,
- IC_2 = {true | false} = if user is mute,
- IC_3 = {true | false} = if user is deaf,
- IC_4 = {true | false} = if user is familiar with Braille,

- IC_5 = {quiet | noisy} = environment's noise level,
- IC_6 ={silence required | silence optional} = environment's noise level restriction, and
- IC_7 = {PC or Laptop or MAC | PDA | Cell phone} = user's computing device.

Let us assume further that the intended user is a visually-impaired one. In this case, $Modality\ Possible = (VO_{in} \vee M_{in}) \wedge (VO_{out} \vee M_{out})$ wherein the only modalities are vocal input and output and tactile (designated here as manual) input and output.

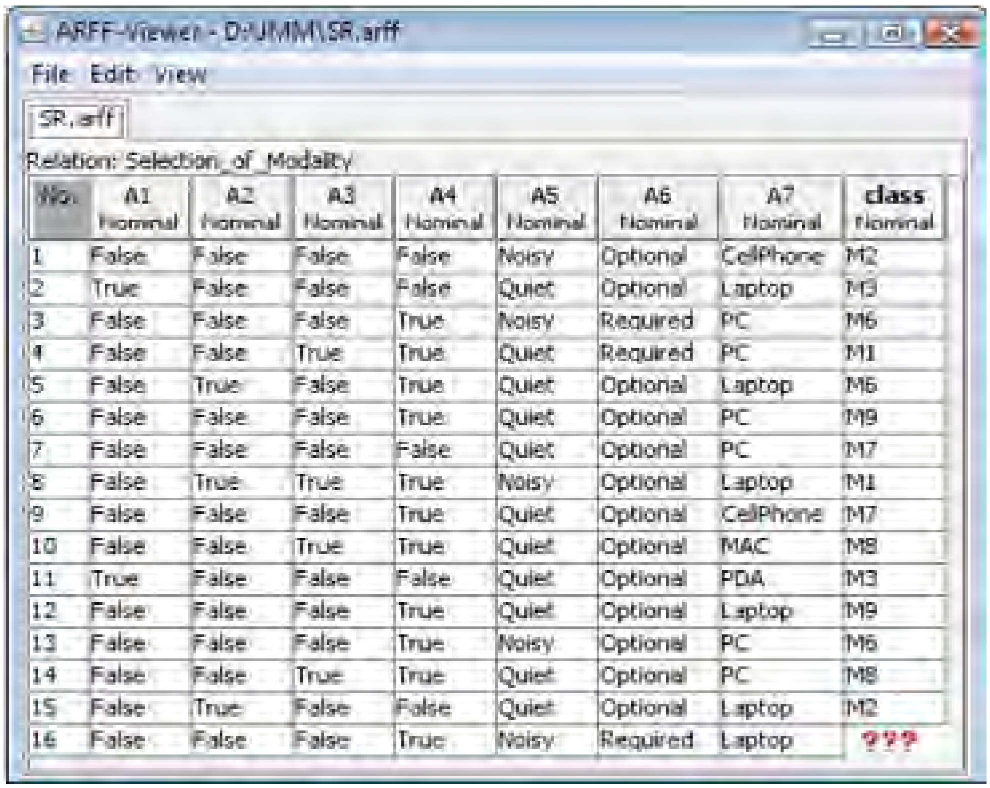
Given the above constraints, the set of possible modalities is given by **M** = {M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9} wherein M_1 = {M_{in}, M_{out}}; M_2 = {M_{in}, VO_{out}}; M_3 = {VO_{in}, VO_{out}}; M_4 = {VO_{in}, M_{out}}; M_5 = {VO_{in}, M_{out}, VO_{out}}; M_6 = {M_{in}, M_{out}, VO_{out}}; M_7 = {M_{in}, VO_{in}, VO_{out}}; M_8 = {M_{in}, VO_{in}, M_{out}}; M_9 = {M_{in}, VO_{in}, M_{out}, VO_{out}}. In this example, let us assume the following interaction context: (a) **user context**: blind with no further handicaps, familiar with Braille; hence IC_1 = False, IC_2 = False, IC_3 = False, IC_4 = True (b) **environmental context**: the user is in a classroom, then IC_5 = noisy, IC_6 = silence required (c) **system context**: the user works on a laptop; IC_7 = Laptop.

The system now finds the modality that suits the given interaction context. Let us assume that a certain multimodal computing system's SR contains recorded scenarios as shown in Figure 1.14. The given figure is generated by using WEKA (*Waikato Environment for Knowledge Analysis*) (Witten and Frank 2005) which is a collection of machine learning algorithms for data mining tasks. It is used in testing a machine learning algorithm as it contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

As shown in the diagram, there are already 15 scenarios representing the system's acquired knowledge. The 16^{th} scenario represents a new case. Using Equation 1.22, and with reference to the given interaction context and SR, the suitability score of $\mathbf{M_j}$ (where j = 1 to 9) can be calculated. Let us consider, for instance, the calculations involved with modality M_1:

Suitability_Score (M1) = $P(IC_1 = False \mid M_1) * P(IC_2 = False \mid M_1) * \ldots * P(IC_7 = Laptop \mid M_1) * P(M_1) = 1 * 0.5 * 0 * \ldots. * 2/15 = 0$

wherein $P(IC_1 = False \mid M_1) = 2/2$, $P(IC_2 = False \mid M_1) = ½$, $P(IC_3 = False \mid M_1) = 0/2$, $P(IC_4 = True \mid M_1) = 2/2$, $P(IC_5 = Noisy \mid M_1) = ½$, $P(IC_6 = \text{silence required} \mid M_1) = ½$, and $P(IC_7 = Laptop \mid M_1) = ½$. Also, $P(M_1) = 2/15$.



No.	A1 Nominal	A2 Nominal	A3 Nominal	A4 Nominal	A5 Nominal	A6 Nominal	A7 Nominal	class Nominal
1	False	False	False	False	Noisy	Optional	CellPhone	M2
2	True	False	False	False	Quiet	Optional	Laptop	M3
3	False	False	False	True	Noisy	Required	PC	M6
4	False	False	True	True	Quiet	Required	PC	M1
5	False	True	False	True	Quiet	Optional	Laptop	M6
6	False	False	False	True	Quiet	Optional	PC	M9
7	False	False	False	False	Quiet	Optional	PC	M7
8	False	True	True	True	Noisy	Optional	Laptop	M1
9	False	False	False	True	Quiet	Optional	CellPhone	M7
10	False	False	True	True	Quiet	Optional	MAC	M8
11	True	False	False	False	Quiet	Optional	PDA	M3
12	False	False	False	True	Quiet	Optional	Laptop	M9
13	False	False	False	True	Noisy	Optional	PC	M6
14	False	False	True	True	Quiet	Optional	PC	M8
15	False	True	False	False	Quiet	Optional	Laptop	M2
16	False	False	False	True	Noisy	Required	Laptop	???

Figure 1.14 A sample snapshot of a scenario repository (SR).

Similarly, we do calculate the suitability score of all other remaining modalities. Using the same procedure, the modality that yields the highest suitability score is M_6:

Suitability_Score (M6) = $P(IC_1 = False \mid M6) \times P(IC_2 = False \mid M6) \times \ldots \times P(IC_7 = Laptop \mid M6) \times P(M6) = 1 \times 2/3 \times 1 \times 1 \times 2/3 \times 1/3 \times 1/3 \times 3/15 = 0.00976$.

By applying the ML algorithm (see Figure 1.15), M_6 appears to respect the conditions on possibility of modality; hence, it is chosen as the optimal modality for the given **IC**. This new scenario will then be added to SR as a newly-acquired knowledge (i.e. as scenario #16 in SR).

Figure 1.15 shows the algorithm in finding the optimal modalities given an instance of interaction context. The first algorithm calculates the suitability score of each element of the power set of **M**, **¡(M)** for the given context. The second algorithm finds the optimal modality.

```
Suitability Score of Modality Mj
Input:
 1. Interaction Context ICi
2. Modality Mj
3. Scenario repository SR
Output: Suitability score of Mj
Procedure:
  get ICi, UCmax, ECmax and SCmax
  for each UC Parameter UCj do
    P(UCj|Mj) = P(UCj|Mj) * P(UCj|Mj)
  endfor
  for each EC Parameter ECj do
    P(ECj|Mj) = P(ECj|Mj) * P(ECj|Mj)
  endfor
  for each SC Parameter SCj do
    P(SCmj|Mj) = P(SCmj|Mj) * P(SCj|Mj)
  endfor
  P(Mj) = #(Mj)/card(SR)
  Suitability Score Mj = P(UCj|Mj)*P(ECj|Mj)*P(SCmj|Mj)*P(Mj)
  Return Suitability Score Mj
Endprocedure
```

```
Finding_Optimal_Modality
Input:
 1. Interaction Context ICi
2. Modality set M
3. Scenario Repository SR
Output: Optimal modality
Procedure:
  get ICi and Modality set M
  init
  Max ← Suitability Score of modality M1,1
  Optimal modality ← M1,1
  for each element Mj of power set of M
  if (VIin OR VOin OR Min) ∈ Mj AND (VIout OR VOout OR Mout) ∈ Mj then
    if Suitability Score of Mj > Max then
              Max ← Suitability Score of Mj
              Optimal modality ← Mj
    endif
  endif
  endfor
  Return Optimal modality
Endprocedure
```

Figure 1.15 Algorithms: (Left) Given an interaction context IC_i, the algorithm calculates the suitability score of each modality M_j belonging to the power set ¡(M), (Right) Algorithm for finding the optimal modality.

Once the optimal modalities for data input and output to the instance of interaction context are chosen (i.e. $\hat{M}$), we then proceed to determining whether there are sufficient media devices that will support the selected modalities. If such is the case, then the concerned media devices are activated, otherwise, the optimal modality that was chosen earlier needs to be updated. That is to say that the modality that cannot be supported by available media devices is taken out from the list. This results in the HKA's evaluation if Equation 1.11 still holds true. If the result is affirmative, then the modality is possible, the replacement modality is calculated and the newly found optimal modality is implemented. Otherwise, there is a failure of modality.

Generally, there is more than one media device that supports a specific modality. Consequently, when too many devices are available to support the chosen modality, then only the top-ranked media device needs to be activated. Moreover, the ranking of media devices also serves purpose in finding replacement to a failed device. When a top-ranked device is malfunctioning or not available then the device that is next in priority ranking is activated. Given a specific media device D_i ($1 \leq i \leq n$), where i is priority index and n is the number of media devices that can support the chosen modality $\mathbf{M_j}$, then the probability that it is adopted to implement the chosen modality is given by:

$$P(D_i \mid M_j) = 1 - \sum_{1}^{i-1}(1/n) \tag{1.26}$$

To demonstrate the essence of the above equation, supposed that there are 3 media devices (n = 3) supporting modality M_j, denoted by D_1 (first priority), D_2 (second priority) and D_3 (third priority). The probability that D_1 is selected for implementation of modality M_j is 1 – 0 = 1, that of D_2 is 1 – 1/3 = 2/3 and that of D_3 is 1 – (1/3 + 1/3) = 1 – 2/3 = 1/3. Note that in this thesis, the numerical subscripts of those devices with higher priority are numerically smaller than the numerical subscript of those with lower priority. Example is device D_1 (subscript is 1) has a higher priority than device D_2 (subscript is 2).

The *media devices priority table* (MDPT) is a tool that we used in implementing the media devices' priority ranking. In MDPT, devices are grouped based on their category and ranked by priority. The top-ranked media device is always chosen for activation. When the highly-ranked device is found defective, the MDPT helps in determining which one replaces the defective one.

When a *new media device* $\mathbf{d_{new}}$ is *added* or *introduced* to the system for the *first time*, the device is associated to a modality and is given a priority ranking **r** by the user. What happen to the rankings of other devices $\mathbf{d_i}$, $(1 \le i \le n$, and n = number of media devices) which are in the same modality as $\mathbf{d_{new}}$ in the MDPT? Two things may happen, depending on the user's selection. The first possibility is that after having the new device's priority Priority($\mathbf{d_{new}}$) set to **r** then the priority of the other device **i**, $(1 \le i \le n)$ denoted Priority($\mathbf{d_i}$), remains the same. The second possibility is the priority rankings of all media devices ranked **r** or lower are adjusted such that their new priority rankings are one lower than their previous rankings. Formally, in Z (Lightfoot 2001), this is specified as: $\forall$i, $\exists$r: £$_1$; $\forall \mathbf{d_i}$, $\exists \mathbf{d_{new}}$: **Devices |** (Priority($\mathbf{d_{new}}$) = r $\wedge$ Priority($\mathbf{d_i}$) ≥ r) $\Rightarrow$ Priority($\mathbf{d_i}$)' = Priority($\mathbf{d_i}$) + 1.

We also proposed mechanism for dynamic reconfiguration of the system to make it more fault tolerant, keeping the system persistent and able to resist breakdown in case certain media devices supporting chosen modalities are found to be missing or malfunctioning. Figure 1.16 shows how a MDPT is assigned to a specific scenario. Figure 1.17 demonstrates how the system finds a replacement to a missing or defective media device while Figure 1.18 shows how MDPT is manipulated to accommodate a newly installed device that the system has not encountered in the past.

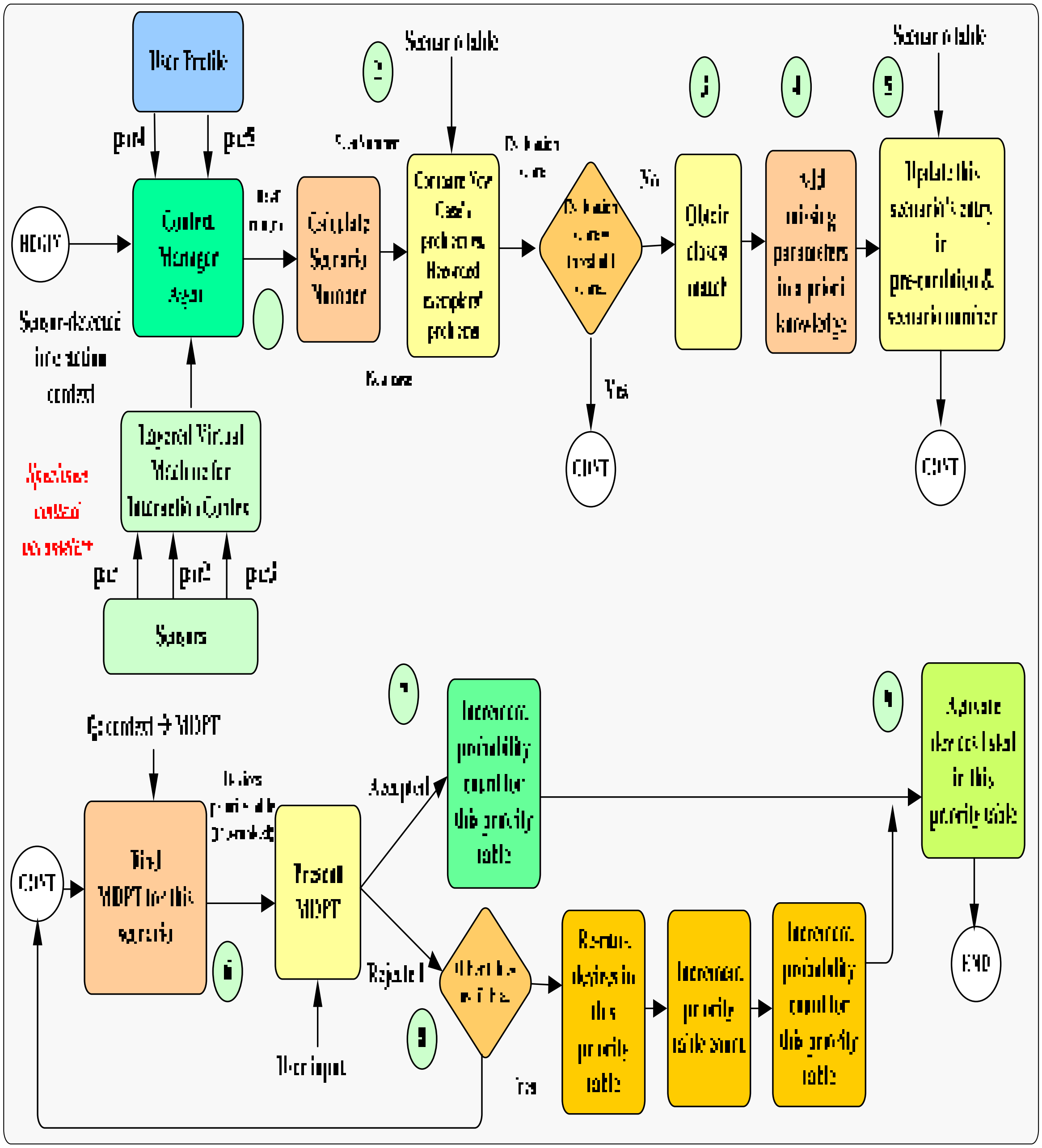


Figure 1.16 The training for choosing appropriate MDPT for a specific context.

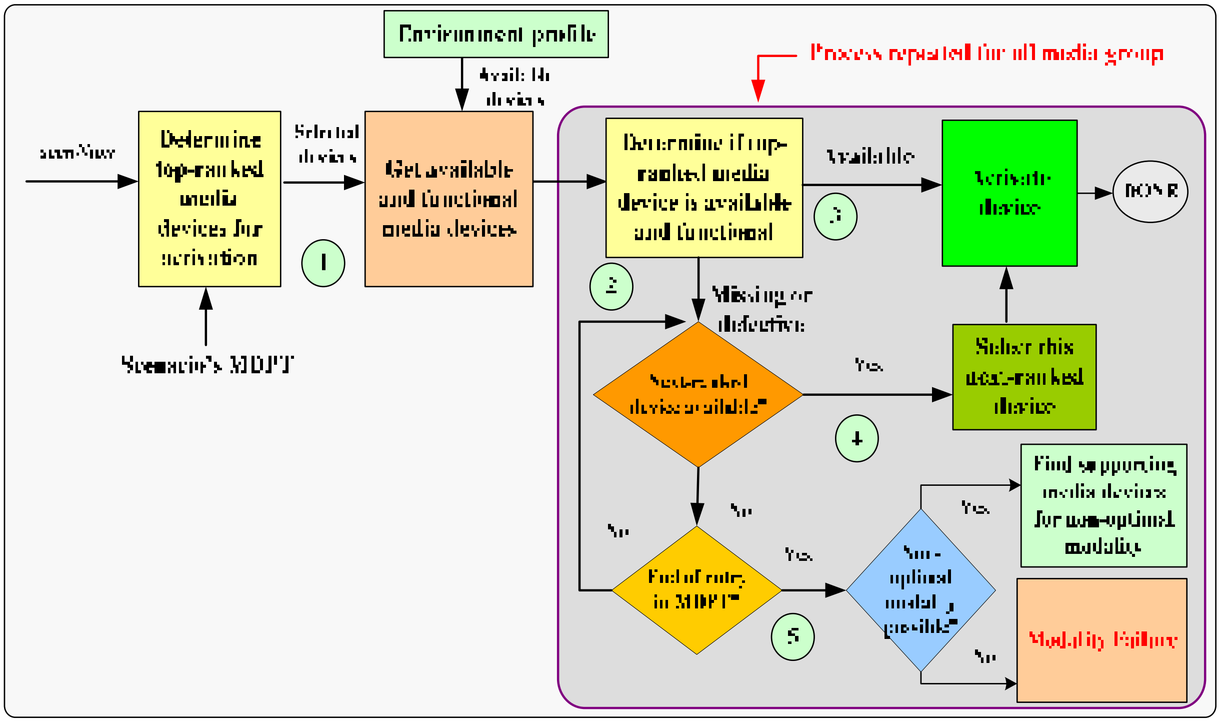


Figure 1.17 The process of finding replacement to a failed or missing device.

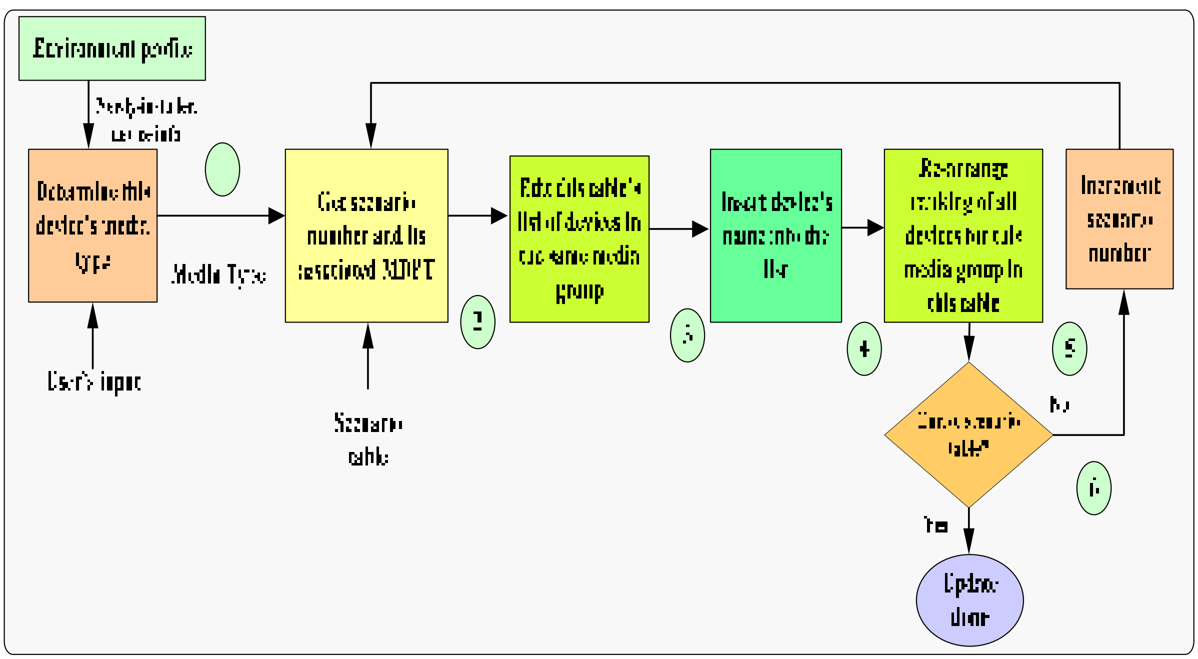


Figure 1.18 The process for updating MDPT due to a newly-installed device.

1.3.6 Mechanism/Paradigm 2: Detection of Applications – Needed to Perform User's Task and Appropriate to the Interaction Context

For the user to work on his computing task, applications need to be instantiated, preferably using the suppliers preferred by the user. These suppliers themselves need to be configured so as the QoS dimensions preference of the user is satisfied, whenever possible. In this mechanism, HKA is trained and learned the following functions:

1. Function **f_1: Data format → Application**. This function maps *data format* (i.e. of form *filename.extension*, such as *file1.doc*, *mymusic.wvm*, *www.mysite.html*, etc.) with its corresponding application. Every data pair belonging to this function is given a relevance score which can either be H (high), L (low) or I(inappropriate). For example, *(.doc, Text Editor)* has a score of H, *(.html, Video Player)* has a score of I, and *(.txt, Web Browser)* has a score of L.

2. Function **f_2: Application → (Preferred Supplier, Priority)**. This function maps an application to a user's preferred supplier and that supplier's priority ranking. In general, the implementation or execution of an application can be done using more than one supplier. For example, *Text Editing* can be implemented using *Microsoft Word*, *Emacs*, *NotePad*, *WordPad*, etc. in the same way that *Web Browsing* can be done using *Internet Explorer*, *Google Chrome*, *Opera*, etc. The priority of these suppliers, with regards to the applications, is based solely on user's preference. Sample elements of function **f_2** can be f_2 = *{(Text Editor, (MSWord, 1)), (Text Editor, (WordPad, 2)), etc.}*

3. Function **f_3: Application i → (QoS dimension j, Priority).** This function maps a specific application i to its QoS dimension parameter j where $1 \leq i \leq$ app_max (max. no. of applications) and **Application** $i \in$ user task. (Recall that in this work, *task* is a computing work the user needs to do. To accomplish this task, the user runs one or more computing applications. For example, a user wishing to shop for a second-hand car may access a web browser, a text editor and a video player). Also, $1 \leq j \leq$ qos_max (max. no. of QoS

dimensions) and QoS dimension $j \in$ **Application** i. Priority is of type £1 (i.e. integer greater than zero). Since there are many possible values for each QoS dimension, the user arranges these values by their priority ranking. Possible elements that may be associated with this function may be:

f_3 = *{(Text Editor, (40 characters per line, 1)), (Web Browser, (low page loading, 3)), (Audio/Video Player, (medium volume, 1)), (Audio/Video Player, (high volume, 2)), etc.}*

Given a user task, one or more applications may be instantiated. For an application, however, there are some suppliers and QoS dimensions selections that can be invoked. Respecting the user's preferences is the way to instantiate an application, but if it is not possible, the dynamic reconfiguration mechanism must look upon various configuration spaces and determine the one that is feasible to the user's needs. Given a user's task that can be implemented using various applications, we have:

- The task's QoS dimension space is given by

$$QoS\ Dimension\ space = \bigotimes_{i=1}^{qos\_\max} D_i \tag{1.27}$$

Given two applications s and t, their QoS dimension space is $D_i(s) \otimes D_i(t)$.

- An application i has its own set of suppliers, called "*supplier space*":

$$Supplier\ space = \bigotimes_{i=1}^{app\_\max} Supp_i \tag{1.28}$$

A *feasible configuration* is a set-up that tries to satisfy the user's preferences given the user's context, and the resources' constraints. When the configuration is feasible, it is said that the user's satisfaction is achieved. Let the *user's satisfaction* to an outcome be within the

Satisfaction space. It is in the interval of [0, 1] in which 0 means the outcome is totally unacceptable while a 1 corresponds to a user's satisfaction. Whenever possible, the system strives to achieve an outcome that is closer to 1.

Given an application, the user's satisfaction is enhanced if his preferences are enforced. The supplier preferences in instantiating an application are given by:

$$\textit{Supplier preferences} = h_s^{x_s} \bullet f_s \, c_s \quad (1.29)$$

where s ∈ Supplier space is an application supplier.

- The term $c_s \in [0, 1]$ reflects how the user cares about supplier *s*. Given an application, if it has *n* suppliers which are arranged in order of user's preference, then $c_{supplier1} = 1$, $c_{supplier2} = 1 - 1/n$, $c_{supplier3} = 1 - 1/n - 1/n$, and so on. The last supplier therefore has a value of c_s close to zero which means that the user cares not to have it if given a choice. In general, in each application, the c_s assigned to supplier i, $1 \leq i \leq n$, is given by:

$$c_{supplier\ i} = 1 - \sum_{1}^{i-1} (1/n) \quad (1.30)$$

- The term f_s: dom(s) → [0, 1] denotes the expected features present in supplier *s*. The supplier *features* are those that are important to the user, other than the QoS dimensions. For example, in a text editor application, the user might prefer a supplier that provides *spelling and grammar checking*, or *equation editor* or feature to *build a table*, etc. For example, if the user listed $n = 3$ preferred features for an application, and the selected supplier supports them, then $f_s = 1$. If, however, one of these features is missing (either because the feature is not installed or the supplier does not have such feature), then the number of *missing* feature $m = 1$ and $f_s = 1 - m/(n + 1) = 1 - ¼ = 0.75$. In general, the user satisfaction with respect to application features is given by:

$$f_{supplier} = 1 - \frac{m}{n + 1} \tag{1.31}$$

- The term $h_s^{x_s}$ expresses the user's satisfaction with respect to the change of the supplier, and is specified as follows: $h_s \in (0, 1]$ is the user's tolerance for a change in the supplier. If this value is close to 1, then the user is fine with the change while a value close to 0 means the user is not happy with the change. The optimized value of h_s is:

$$h_s = arg\ max\,(c_s + c_{rep}) / 2 * c_s \tag{1.32}$$

where c_{rep} is a value obtained from equation (4) for replacement supplier. x_s indicates if change penalty must be considered. $x_s = 1$ if the supplier exchange is due to the dynamic change of environment, while $x_s = 0$ if the exchange is instigated by the user.

Similarly, a user's preferences for QoS dimensions of his applications as given by:

$$QoS\ preference\,s = h_q{}^{x_q} \bullet c_q \tag{1.33}$$

where and $q \in$ QoS dimension space is a QoS dimension of an application. Note that equations (3) and (7) are almost identical except for the differences in the subscripts and the absence of feature in QoS dimensions. The algorithms for finding the optimized QoS and supplier configuration of any application are given in Figure 1.19. In each algorithm, the default configuration is compared with other possible configurations until the one yielding the maximum value of user's satisfaction is found and is returned as result of each algorithm.

A *feasible configuration* is achieved if the user's task can be realized by appropriate applications that are instantiated using the user's preferred suppliers and QoS dimensions. The feasible configuration is given by:

$$\underset{\substack{app(a)\in task \\ s\in supplier(app(a)) \\ q\in QoS\ dim\,(s)}}{\arg\max} = \prod_{a=1}^{app\_max} Supplier\ \ preferences\,(a) \bullet QoS\ \ preferences\,(a) \tag{1.34}$$

The algorithm for finding the feasible configuration of applications within the user's task is shown in Figure 1.20. It finds the feasible configuration in every application.

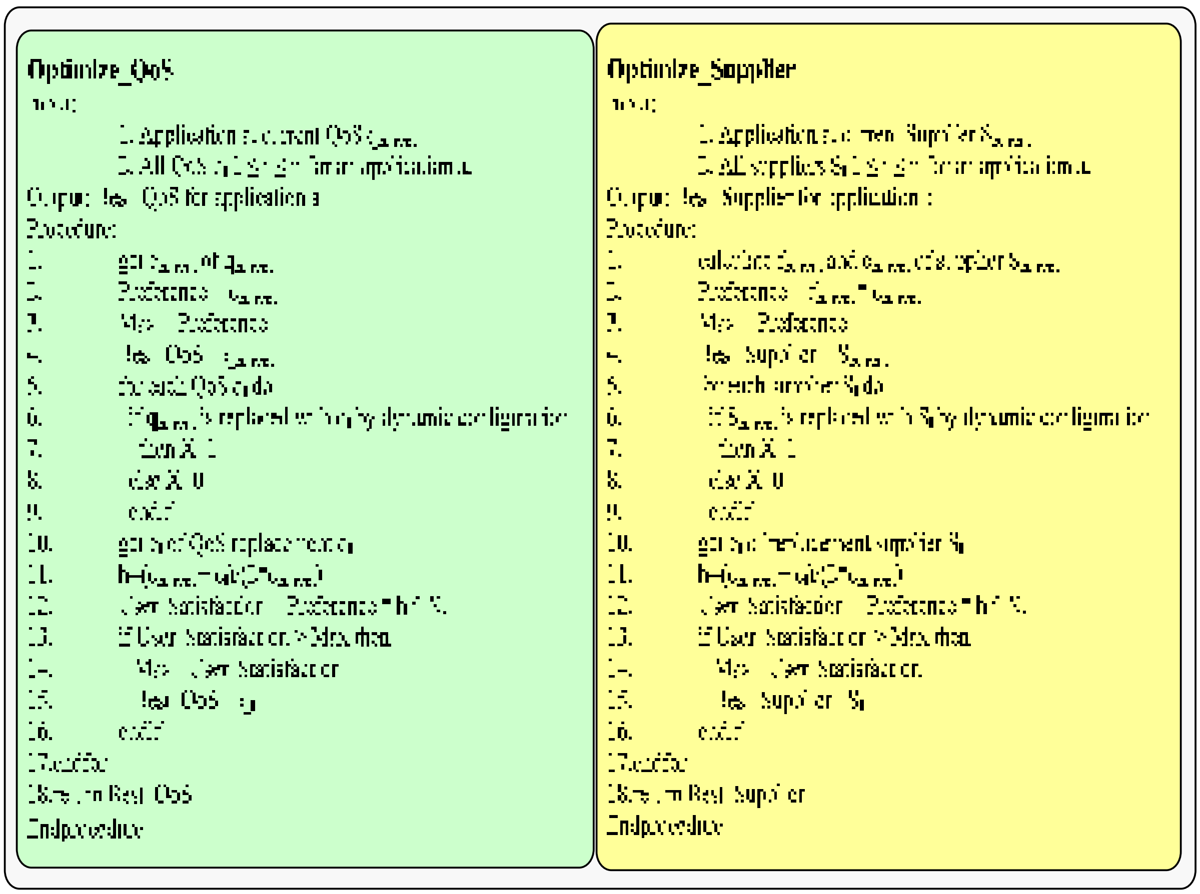


Figure 1.19 Algorithms for optimized QoS and supplier configuration of an application.

The work of (Poladian, Sousa et al. 2006) has positively influenced our work. Equation 1.27, Equation 1.28 and Equation 1.34 were taken from such work. Although previously defined in the same reference, Equation 1.29 and Equation 1.33 have since evolved that their final forms in this paper have become ours. The rest of the other equations are all ours.

```
Optimize task
Input:
            1. Application ai, 1 <= i <= max app
            2. Current Supplier for each application ai.
            3. Current QoS for each application ai
Output: Appropriate suppliers and QoS for the task
Procedure
for i [illegible] 1 to max app
 optimize supplier()
 optimize QoS()
 configuration ai [illegible] Best QoS * Best Supplier
endfor
Endprocedure
```

Figure 1.20 The algorithm for optimizing user's task configuration.

1.3.7 Simulation

As shown in Figure 1.21, the application parameters as well as QoS dimensions used in instantiating an application is based on user's preferences as obtained from his user profile, which itself constitutes an integral part of interaction context. The Petri Net (Jensen, Kristensen et al. 2007) network shows all possible variations of the four specimen input parameters which through simulation produces the resulting media devices that will be activated by the system.

In Figure 1.22, the simulation demonstrates that each combination of interaction context parameters produces different possibilities of implementing modality. The result is that the most appropriate modality is selected for activation.

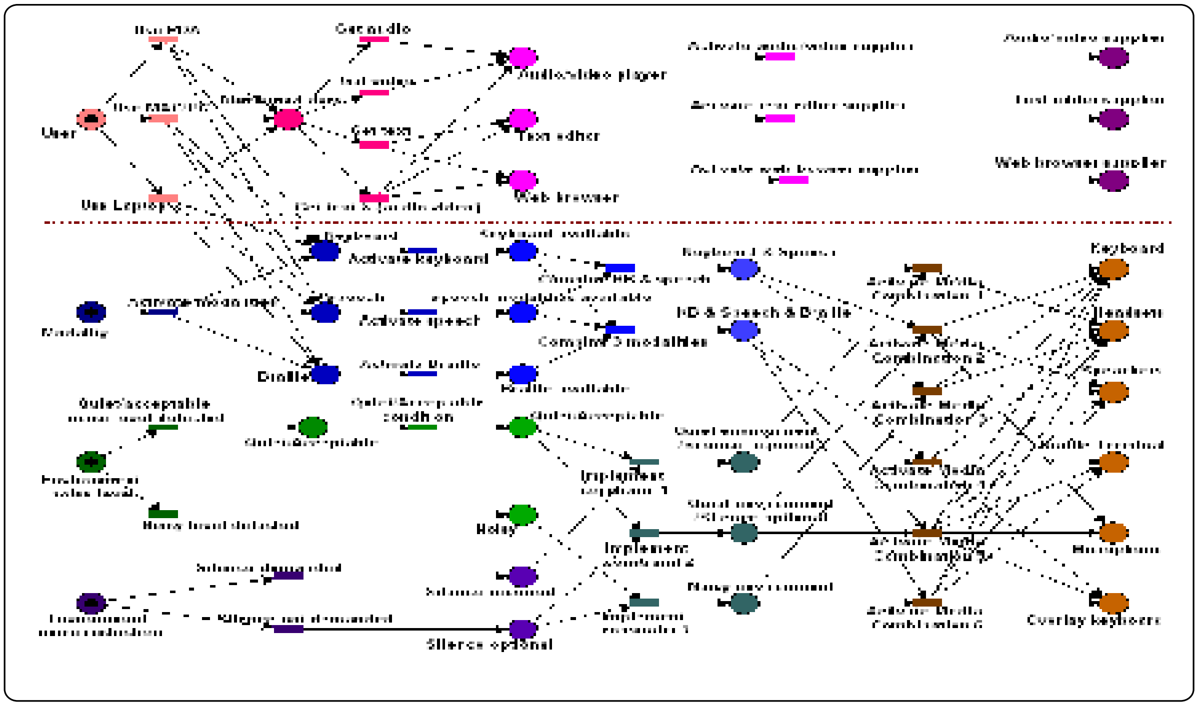


Figure 1.21 Specification using Petri Net showing different pre-conditions scenarios yielding their corresponding post-condition scenarios.

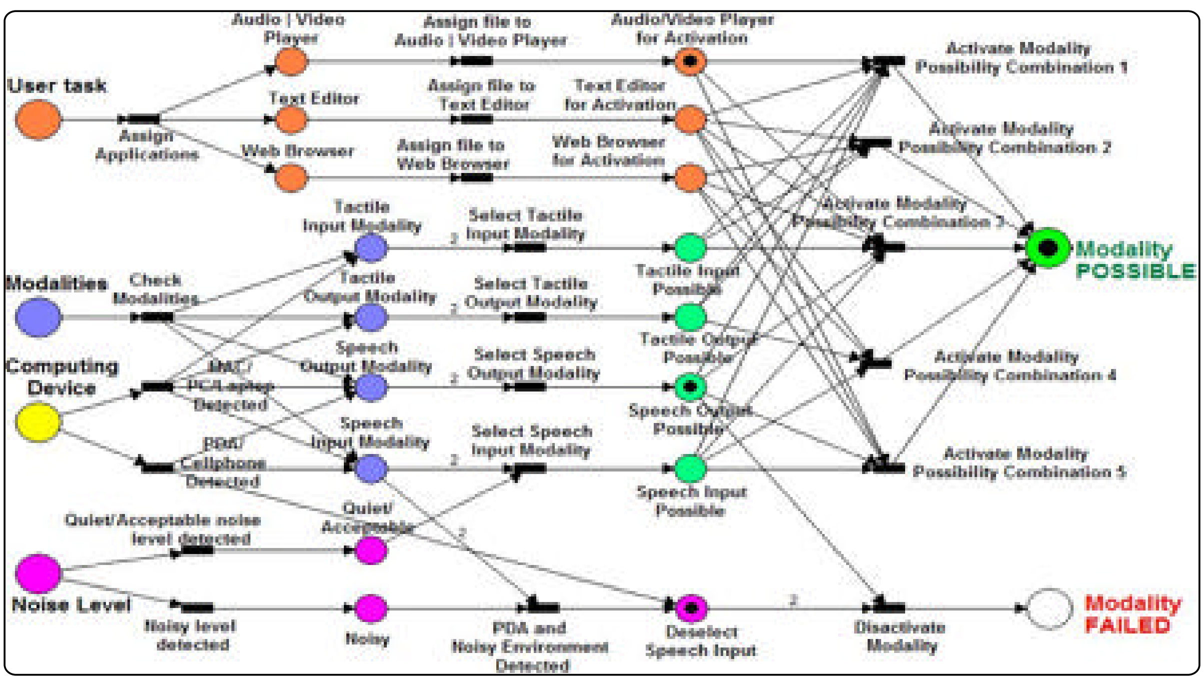


Figure 1.22 Petri Net diagram showing failure of modality as a function of specimen parameters noise level, availability of media devices and user's task.

In Figure 1.23, the simulation shows that various combinations of interaction context parameters (such as user's task, the modalities, the available media devices and noise restrictions in user's workplace, just to cite some as specimen) yield the corresponding modalities as output. The Petri Net of Figure 1.24 shows how an automatic selection of modality may succeed or fail based upon the status of specimen parameters, such as availability of media devices and noise restriction imposed in the workplace.

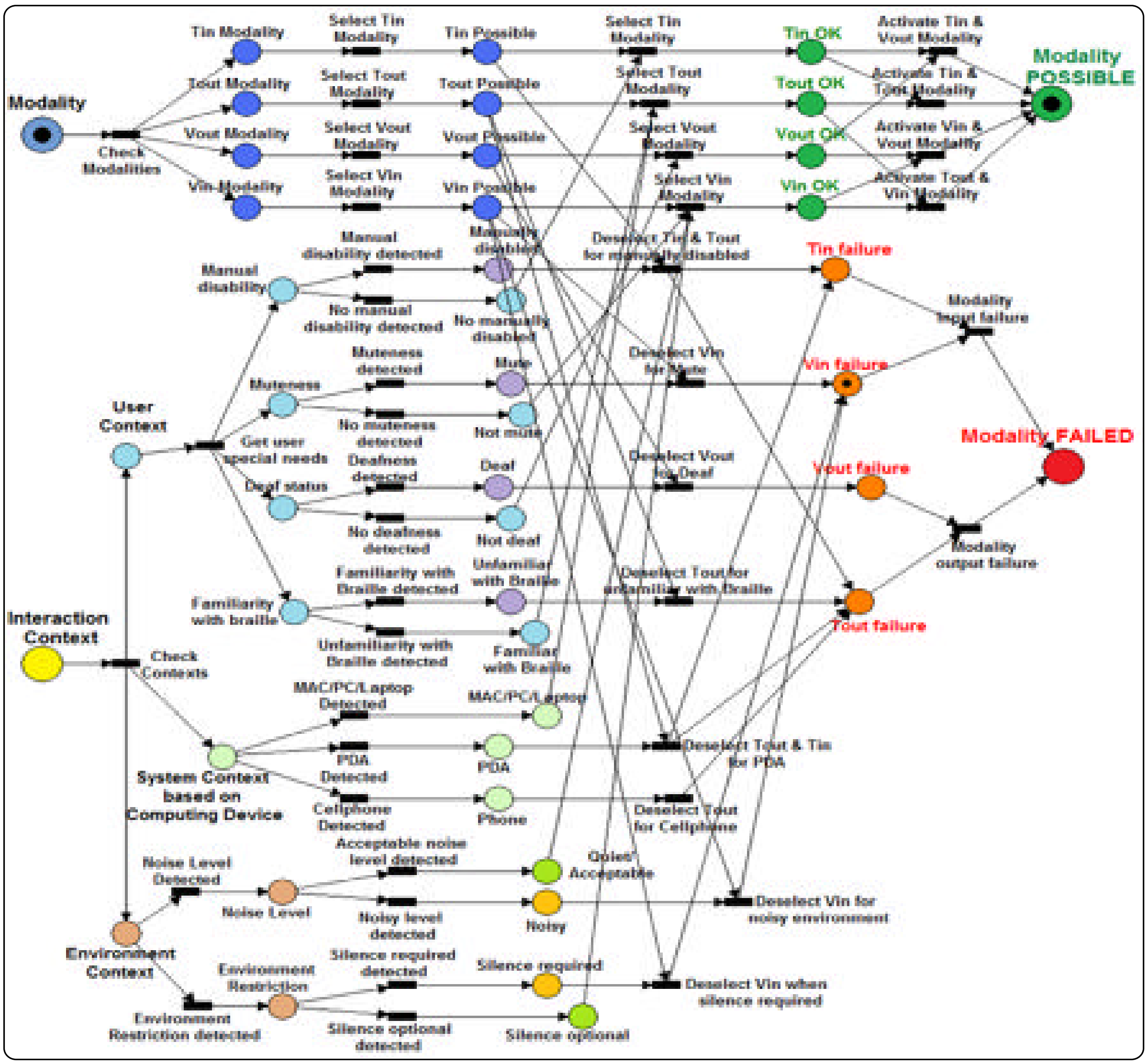


Figure 1.23 Detection if modality is possible or not based on the specimen interaction context.

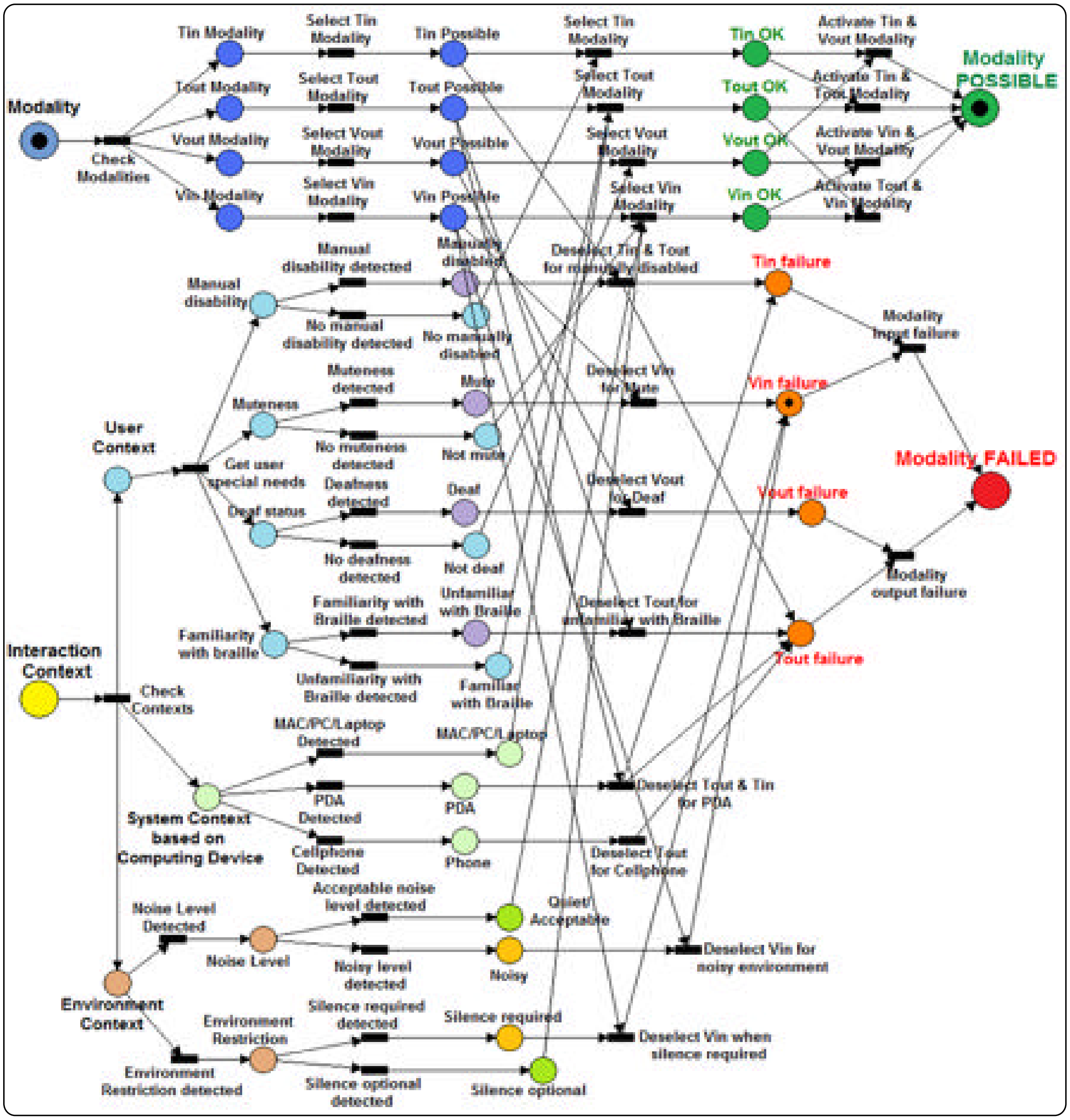


Figure 1.24 Petri Net showing the possibility of failure of modality based on the specimen parameters availability of media devices, and noise restriction within the user's working environment.

The simulation in Figure 1.25 demonstrates the variations in the satisfaction of user when his preferences, registered in the user's profile, are modified through dynamic configuration. The first graph shows the variations of user satisfaction with regards to the supplier and its features; the second graph shows variations in user satisfaction as a function of current supplier and its features and its alternative supplier replacement. The last graph shows the user's satisfaction as a function of current QoS parameters and its alternative replacements.

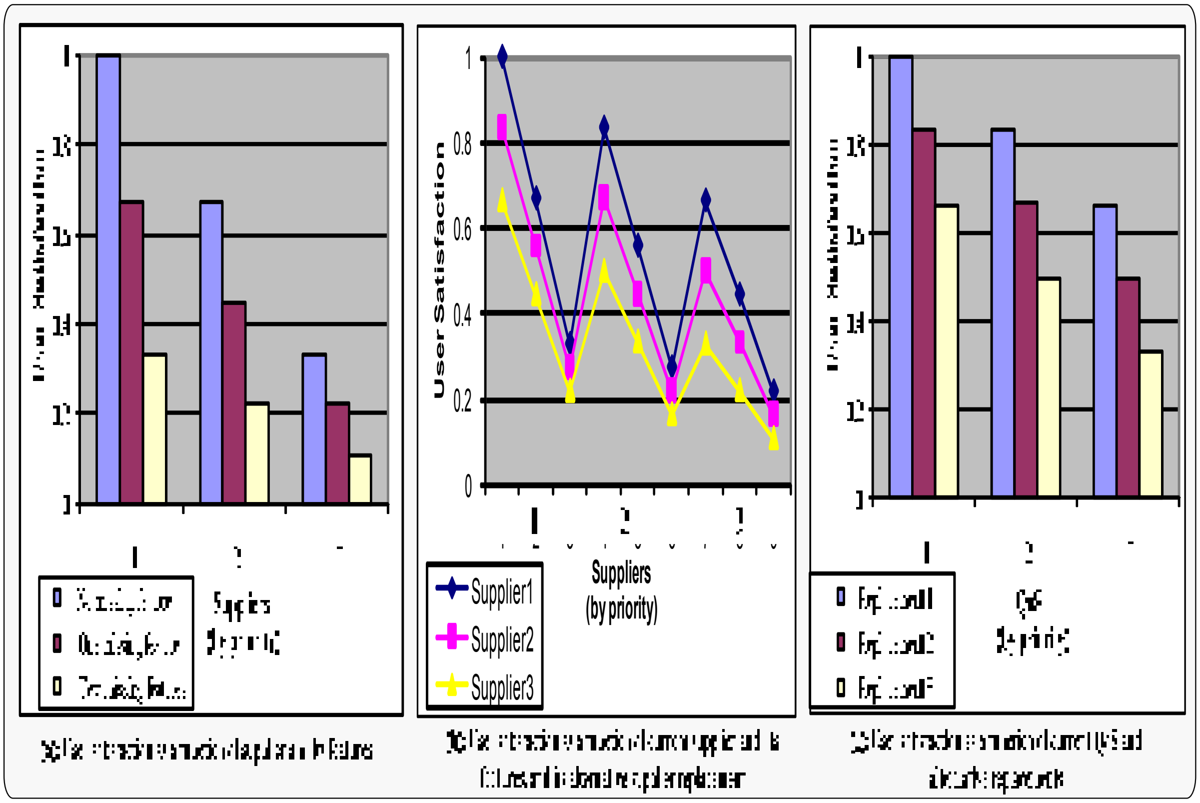


Figure 1.25 Variations of user satisfaction based on user's preferences (suppliers, QoS, and available features of the supplier).

1.3.8 User Interaction Interface

A close look on latest research in user interaction interface reveals that multimodal interface is still far from what it should be – that users are given interaction interface in which they can use the appropriate modality as deemed appropriate to the user's context of interaction. Instead, what is available are user interfaces by which mouse, keyboard and screen are still the impoverished tools used for communication. For instance, the Harvard's *SUPPLE* project for automatically generating user interface adapted to the user's motor and vision abilities (Gajos, Wobbrock et al. 2007; Gajos 2008; Gajos 2010) is promising yet it still uses mouse, keyboard and screen and lacks the flexibility that multimodality provides – the use of other modalities and media. In this regard, we contend that the advancement in pervasive multimodality is closely associated with an adaptive user multimodal interaction interface. Though this thesis is not an in-depth research on this matter, we provide a futuristic vision of what a multimodal user interface should be.

This discussion on user interaction interface is taken into consideration after the proposed system has detected there is no failure of modality in the given context of interaction and that multimodality is possible for implementation.

1.3.8.1 Media Groups and Media Devices

Using commonly-used media devices of our time, the association between media group and media devices can be specified as follows:

$$VIM = eye\ gaze \vee gesture\ interpreter \tag{1.35}$$

$$OIM = speech\ recognition \wedge microphone \tag{1.36}$$

$$MIM = keyboard \vee Braille \vee pen \tag{1.37}$$

$$TIM = mouse \vee virtual\ mouse \vee touch\ screen \tag{1.38}$$

$$VOM = terminal\ screen \vee printer \tag{1.39}$$

$$HOM = speech\ synthesis \wedge (speaker \vee headset) \tag{1.40}$$

$$TOM = tactile\ keyboard \vee Braille \tag{1.41}$$

It is to be noted that these relationships are limited to commonly-used media devices. Some of these relationships can be modified with ease by the user to include other media devices of his preference.

Let **INPUT_MEDIA_GROUP** = {VIM, OIM, TIM, MIM}, then the power set (i.e. the set of all subsets) of this group is given by **¡(INPUT_MEDIA_GROUP)** = {{VIM}, {OIM}, (Chen, Finin et al.), {MIM}, {VIM, OIM}, {VIM, TIM}, {VIM, MIM}, {VIM, OIM, TIM}, {VIM, OIM, MIM}, {VIM, TIM, MIM}, {VIM, OIM, TIM, MIM}, {OIM, TIM}, {OIM, MIM}, {OIM, TIM, MIM}, {TIM, MIM}, {}}. These results indicate that there can be four types of user interface. Note that, by definition, a human-machine interface is generally considered as a function of input modalities only. Hence, the possible types of human-machine interaction interfaces are:

- **unimodal interface** – media devices (and supporting software) belonging to VIM, OIM, TIM and MIM can be used, but there is no fusion of data generated by one media with the data generated by another media (ex. speech, pen, vision)

- **bimodal interface** – there are 6 possible combinations of fusion of data generated by two media devices – that of {VIM, OIM}, {VIM, TIM}, {VIM, MIM}, {OIM, TIM}, {OIM, MIM}, and {TIM, MIM}. The current state-of-the-art multimodal interfaces fall in this category.

- **trimodal interface** – this interface allows the combination of data that are generated by three media devices into a new meaningful data; there are 4 possible selections, namely: {VIM, OIM, TIM}, {VIM, OIM, MIM}, {VIM, TIM, MIM}, and {OIM, TIM, MIM}

- **quadmodal interface** – this one would combine all types of input media altogether, {VIM, OIM, TIM, MIM}.

As far as research advancement (i.e. year 2010) is concerned, a user interface can only be unimodal or bimodal. There is no evidence that a trimodal interface, let alone a quadmodal one, exists, at least not yet.

1.3.8.2 User-Preferred Interface

Given that a unimodal or bimodal user interface is possible, we believe that the selection of user interface should be based on (i) the modalities and media groups that suit the given context (ii) the availability of media devices (and their supporting software) that support the chosen modalities, and (iii) the availability of the preferred interface system or middleware within the user's computing system, and (iv) the user's preference on these interface as given by their priority rankings.

In determining if a unimodal or multimodal interface is to be implemented, let there be a *human-machine interaction interface priority table* (HMIIPT). This table contains information related to the user's preferences, such as: (i) the priority ranking of multimodal and unimodal interface, (ii) the priority ranking of modalities within the interface, and (iii) the priority ranking of media devices that support a modality. See Tableau 1.2 for a sample HMIIPT.

Suppose that the user prefers a unimodal interface over a multimodal (actually, bimodal) one, using HMIIPT, the system determines the ranking assigned to each of the input modalities. Then the priority ranking of media devices supporting a preferred modality are taken from the *media devices priority table* (see Tableau 1.2). For multimodal interface, the user is also consulted in the priority rankings of all modality combinations/fusion. In the same manner, the priority of media devices supporting the multimodal fusion is also indicated in HMIIPT.

The selection process for optimal user interface modality uses the following functions to determine the score of each user interface mode. We take the result yielding the highest score as the optimal user interface modality:

Tableau 1.2 A sample human-machine interaction interface priority table (HMIIPT)

User Interface Modality by Priority			
Interaction Interface Priority	Priority of Modalities in Unimodal Interface	Priority of Modalities in Bimodal Interface	Priority of Media Devices in Bimodal Interface
1. Bimodal Interaction 2. Unimodal Interface	1. MIM 2. OIM 3. TIM 4. VIM	1. VIM and OIM	1. eye gaze – speech, 2. gesture - speech
		2. VIM and TIM	1. eye gaze – mouse, 2. eye gaze – virtual mouse, 3. gesture – mouse, etc.
	Priority of Media Devices in Unimodal Interface	3. VIM and TIM	1. eye gaze – keyboard, 2. eye gaze – Braille, 3. gesture – keyboard, etc
		4. OIM and TIM	1. speech – mouse, 2. speech – touch screen, 3. speech – virtual mouse, etc.
	Same Priority Ranking as MDPT	5. OIM and MIM	1. speech – keyboard, 2. speech – pen, 3. speech – joystick, etc.
		6. TIM and MIM	1. mouse – keyboard, 2. mouse-Braille, 3. mouse-pen, etc.

$$\textit{User Interface Modality Score} = \textit{User Interface Priority} \; x \; \textit{Modality Priority} \; x \; \textit{Media Devices Priority} \tag{1.42}$$

$$\textit{User Interface Priority} = (m + 1 - p)/m \tag{1.43}$$

$$\textit{Modality Priority} = (q + 1 - p)/q \tag{1.44}$$

$$\textit{Media Device}_i \; \textit{Priority} = 1 - \sum_{1}^{i-1} (1/n) \tag{1.45}$$

such that for user interface priority, the variable m = number of types of user interface (i.e. unimodal, bimodal, etc. available in HMIIPT) and p = priority ranking as obtained from HMIIPT, $1 \leq p \leq m$. For modality's priority, the variable q = number of modality selections (i.e. available in HMIIPT) and p = priority ranking of the specified modality as obtained from HMIIPT, $1 \leq p \leq q$. For media devices priority, n = available media devices supporting the chosen modality and media group (see Equations (1.28) through (1.34)). Also, given the i^{th} device, where $1 \leq i \leq n$, then d_i = priority ranking obtained from MDPT and n = number of media devices supporting the same modality as the i^{th} device.

1.4 Summary

A few years after the "*third wave of computing*" and "*calm technology*" (Weiser and Brown 1996), was envisioned, we are all experiencing what Weiser envisioned with regards to ubiquitous computing: (i) that an end user is connected to many computers in day-to-day life, (ii) the computer becomes a quiet and invisible servant, and (iii) the computing is calm and the computer does not require to be a user's focus of attention. Indeed, ubiquitous computing is here within our midst and it has become an essential part of our day-to-day existence.

Several *characteristics* and *scientific terms* are associated with *ubiquitous computing*, among them *ambient intelligence*, *smart devices*, and *context awareness*, just to cite a few. In this work, we are particularly interested in context awareness. Various research efforts were exerted in defining what context is; in fact, *there is not even a general consensus of what context really means* as individual researcher defines context differently. In this regard, we differentiate ourselves from other researchers in the sense that we only wish that the parameters of the context that need to be considered are those that are *important to the user* in the course of his undertaking of an activity. And for us, no one knows better than the end user as far as what parameters constitute context and what are not. This leads us to an *incremental definition of context* – defining one context parameter at a time. Our desire for a more conclusive context also leads us to a redefinition of context, for it to become *interaction context* – the combination of the *context of the user*, the context of his *environment* and the context of his *computing system*. Each of these subcomponents of interaction context is itself composed of various parameters (e.g. environment context can be a function of noise level, time of the day, brightness/darkness of the workplace, etc.).

A closer look at the current state-of-the-art informs that there is a *lack of ongoing research* in *pervasive multimodality*. The same can be said about the adaptation of an application with regards to the given instance of interaction context. Instead, most, if not all of the researches in the domain were mostly delegated to the definition of context, the capture of context and how to use context within an application. Ours is different – a context-adaptive pervasive multimodal multimedia computing system adapts itself or reconfigures itself dynamically with respect to the given instance of interaction context in order to provide the end user the environment, the modality, the media devices and the configuration of his applications that will allow such end user to continue working on his computing task anytime and anywhere he wishes.

In this thesis, we specify *two distinct contributions* – a mechanism (i.e. paradigm) that selects the modality and its supporting media devices based on the given interaction context and another mechanism that configures the system dynamically based on the given user task and instance of interaction context. We also presented concepts behind the selection and implementation of user interface that suits the user with regards to the suitable modalities as decided by the system.

In summary, here is what we have contributed to the domain:

1. A design and implementation of layered virtual machine for implementing incremental context. Its design is robust in the sense that a big chunk of this tool can be associated with whatever system that adds, modifies, and deletes one context parameter at a time. It is implemented using Java.

2. Various formulas that we ourselves designed and conceptualized in order to calculate and determine the optimal modalities for any given instance of interaction context.

3. Various formulas that we have developed, others being inspired by other researchers and eventually modified to suit our needs, in order to determine the dynamic configuration mechanism that suits the user's preferences as well as the given instance of interaction context.

4. Training and learning mechanisms for our machine learning component – learning one experience at a time that, with time, will render the machine intelligent enough that it will be able to react to whatever possible variation of pre-condition scenario it will encounter.

5. Various simulations that demonstrate that our concept and idea of interaction context-adaptive pervasive multimodal multimedia computing system is possible and feasible.

1.5 Conclusion of Chapter 1

The very essence of this thesis work is to be able to incorporate *dynamic components* into a pervasive multimodal multimedia system that will allow the system to respond to the new requirements of adaptability based on the evolution of context. This could never been truer than in the case of multimodal applications which largely take into account adaptability to a much larger context called interaction context.

To attain this objective, we designed a global architecture called context-sensitive which has the capability of perceiving the user's condition and situation in his workplace and consequently adapt all the behavior of the system to suit the situation in consideration. Afterwards, we started developing in details the components that are the most important in our architecture. The first one permits the incremental definition of context of interaction. A generic tool was built; it was robust that it can be applied to application of whatever domain that uses context. The second component is capable of learning and acquiring knowledge. It takes instance of interaction context as input and produces output that may be the selection of modalities (and activation of its supporting media devices) or the instantiation of applications needed for the user to work on his computing task. It uses *case-based reasoning with supervised learning* as a technique for learning. For this reason, the system component is able to render system's adaptation to the given situation automatic. Two automation mechanisms/paradigms were proposed within this component: (i) an automation mechanism for the selection of modalities and their supporting media devices that are considered most suitable to the given context of interaction of the user, and (ii) an automation mechanism for the selection of applications and the configuration of such applications' parameters, taking into account the user's preferences and the given instance of interaction. In both mechanisms, we strive to come up with the optimal configurations and its quantification. The concepts and ideas in this research work can be easily implemented using a multi-agent system and each of these components and mechanisms in this architecture can be implemented as agents themselves within the platform of dynamic reconfiguration.

CHAPITRE 2

TOWARDS A CONTEXT-AWARE AND ADAPTIVE MULTIMODALITY

Manolo Dulva Hina [1,2], Chakib Tadj [1], Amar Ramdane-Cherif [2], Nicole Levy[2]
[1] LATIS Laboratory, Université du Québec, École de technologie supérieure,
1100, rue Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines,
45, avenue des États-Unis, Versailles Cedex, 78035 France

This chapter is published as an article in Research in Computing Science Journal, Special Issue: Advances in Computer Science and Engineering, Vol. 29, 2007, ISSN: 1870-4069, Mexico.

Résumé

La *multimodalité diffuse* peut réaliser une informatique de n'importe où et n'importe quand en utilisant les différents modes d'interaction personne-machine supportés par des dispositifs multimédias non traditionnels comme le souris, l'écran et le clavier. Pour une meilleure efficacité, les modalités doivent être sélectionnées en fonction de leur pertinence ou conformité avec le contexte d'interaction donné (c.-à-d. le contexte de l'utilisateur, celui de son environnement et du système informatique) et la disponibilité des dispositifs multimédias. Pour être tolérant à la faute, le système doit être capable de trouver un remplacement aux dispositifs défectueux ou non disponibles. Cet article présente le paradigme d'un tel système. Nous présentons des solutions aux défis techniques incluant l'établissement du lien entre le contexte d'interaction, la modalité et les dispositifs multimédias. Nous avons également formulé un mécanisme permettant l'apprentissage incrémental associant le contexte d'interaction avec les modalités appropriées.

Mots clés : multimodalité diffuse, système sensible au contexte, système multimodal multimédia, informatique diffuse, système adaptatif.

Abstract

Pervasive multimodality would realize anytime, anywhere computing using various modes of man-machine interaction and supported by media devices other than the traditional keyboard-mouse-screen for data input/output. For utmost efficiency, modalities must be chosen based on their suitability to a given interaction context (i.e. combined user, environment and system contexts) and availability of media devices. To be fault-tolerant, the same system must be capable of finding replacement to a failed supporting media device. This paper presents a paradigm of such computing system. Proposed solutions to some technical challenges including establishment of the relationships among interaction context, modalities and media devices, and finding a mechanism for incremental learning related to interaction context and its suitable modalities are presented.

Keywords: pervasive multimodality, context-aware system, multimodal multimedia system, pervasive computing, adaptive system.

2.1 Introduction

In the very near future, we shall be living in a society wherein pervasive computing (also known as *ubiquitous computing*) (Weiser 1991; Vasilakos and Pedrycz 2006) 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 (Satyanarayanan 2001) will be available and the applications seamlessly adapting accordingly to the user's context (Coutaz, Crowley et al. 2005) and available resources. Multimodality (Ringland and Scahill 2003), 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, depending on their availability and context suitability, may be selected to support a chosen modality. Hence, pervasive multimodality shall be a computing trend in the future, a computing that adapts to the needs of the users, including those with disabilities. To further enhance its functionalities, a system may be designed with machine learning (Mitchell 1997;

Alpaydin 2004) capability, that intelligent mechanism in which the machine "learns" from previous experiences, improves system performance, encourages autonomy and promotes fault-tolerance. 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 of pervasive multimodality. The rest of this paper is structured as follows. Works related to ours are listed in section 2.2. Section 2.3 lists down some of the technical challenges and our proposed solutions. The main contents, namely matters related to interaction context and multimodality, and the context learning and adaptation are discussed in sections 2.4 and 2.5. The paper is concluded in section 2.6.

2.2 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. Sample recent works in this domain include an interface for wireless user interface (Ringland and Scahill 2003) and the static user interface (Oviatt and Cohen 2000). Multimodality also refers to the fusion of two (or more) modalities. Some sample works in this area include the combined speech and pen inputs (Oviatt 2000) and the combined speech and lips movements (Rubin, Vatikiotis-Bateson et al. 1998). Multimodality as a domain in human-computer interface provides increase usability to the user, such that a modality that is weak for a given setting may be replaced by another but more appropriate modality. Comparatively, our goal on research of pervasive multimodality is to provide anytime, anywhere computing using modalities that are suitable to the given context.

In (Coutaz, Crowley et al. 2005), Coutaz explained the importance of context. Research have gone a long way since Dey provided the basic definition of context (Dey and Abowd 1999) as applied in context-aware computing (Dey 2001). Rey and Coutaz updated the definition in (Rey and Coutaz 2002) 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 a very active ongoing research in pervasive and mobile computing. The Prism model in Project Aura (Garlan, Siewiorek et al. 2002) demonstrates a user's moving aura (profile and task). Our work has extended the same concept by considering an incremental learning system and is part of the pervasive information, along with the user's profile, task and preferences. We also strive to incorporate adaptability and autonomy into our system using machine learning.

2.3 Technical Challenges

Here, we list down some software engineering challenges by posing specific technical challenges that need to be addressed and describe our approach.

Our goal is to model a pervasive computing system that senses its current IC and accordingly chooses the appropriate modalities and media devices that support the chosen modalities. 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? Which media devices are selected to support a suitable modality?

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

Requirement 3: *Provide a mechanism for the system to be fault-tolerant on matter concerning failed selected 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 input/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 whether a specific modality is suitable to an IC parameter and by how much (i.e. high, medium, low or inappropriate). All media devices must be grouped such that a relationship of modalities and media group would be established.

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

Proposed solution to requirement 3: We strive to design an autonomous, adaptive and fault-tolerant system. In matters concerning faulty media device, a replacement is search for its replacement. 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.

2.4 Interaction Context and Multimodality

An *interaction context*, **IC** = $\{IC_1, IC_2, \ldots, IC_{max}\}$, is a set of all possible interaction contexts. At any given time, a user has a specific interaction context *i* denoted $\mathbf{IC_i}$, $1 \leq i \leq max$, which is a set of variables that are present in the conduct of a user's activity. Each variable is a function of the system's 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:

$$IC_i = UC_k \otimes EC_l \otimes SC_m \quad (2.1)$$

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 context, environment context and system context, respectively. The Cartesian product (symbol: $\otimes$) 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 the conduct of his activity. Any specific user context k is given by:

$$UC_k = \bigotimes_{x=1}^{max_k} ICParam_{kx} \quad (2.2)$$

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

$$EC_l = \bigotimes_{y=1}^{max_l} ICParam_{ly} \quad (2.3)$$

$$SC_m = \bigotimes_{z=1}^{max_m} ICParam_{mz} \quad (2.4)$$

Multimodality refers to the selections of 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 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 (plus 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 is possible if there is at least one modality for data input and at least one modality for data output:

$$Modality = (VI_{in} \vee VO_{in} \vee M_{in}) \wedge (VI_{out} \vee VO_{out} \vee M_{out}) \quad (2.5)$$

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 Figure 2.1.

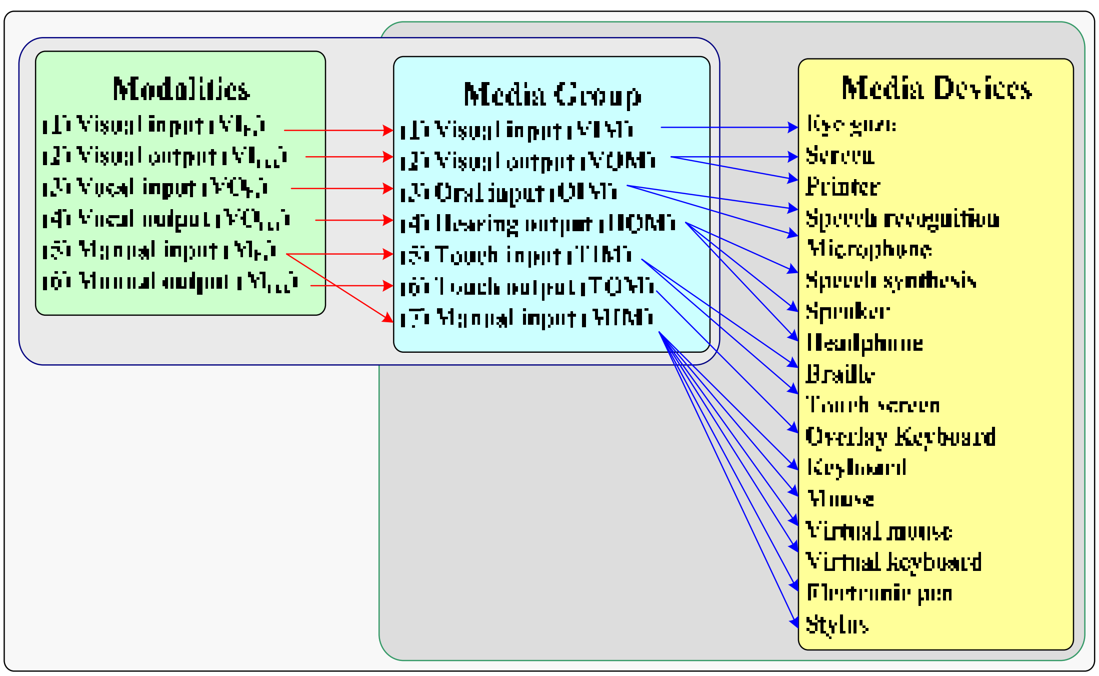


Figure 2.1 The relationship among modalities, media groups and physical media devices.

For the relationship between modalities and media devices, let there be a function g1 that maps a modality to a media group, given by **g_1: Modality → Media Group**. This is shown in Figure 2.1. There is often a case when two or more media devices belong to one media group. In such a case, devices selection is determined through their priority rankings. Hence, let there be a function g2 that maps a media group to a media device and its priority rank, denoted by **g_2: Media group → (Media Device, Priority)**. Sample elements of these functions are:

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

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 parameter. Suitability measures are *high*, *medium*, *low* and *inappropriate*. *High suitability* means that the modality in consideration is the preferred mode for computing; *medium suitability* means the modality is simply an alternative mode for computing, 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 in consideration has *n* suitability scores. We adopt the following conventions:

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:

$$SuitabilityScore_{modality} = \sqrt[n]{\prod_{i=1}^{n} context\_parameter_i} \qquad (2.6)$$

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

$$Suitability_{modality} = \begin{cases} H \text{ if } SuitabilityScore_{modality} = 1.00 \\ M \text{ if } 0.75 \le SuitabilityScore_{modality} < 1.00 \\ L \text{ if } 0.50 \le SuitabilityScore_{modality} < 0.75 \\ I \text{ if } SuitabilityScore_{modality} < 0.50 \end{cases} \qquad (2.7)$$

Figure 2.2 shows the algorithm for determining the suitability of modalities to a given IC and if multimodality is possible (i.e. equation 2.5). The possibility of having multimodality is done by checking that not all of input modalities (i.e. specified by indexes 1, 2 and 3) are scored "inappropriate", and so does for output modalities (i.e. specified by indexes 4, 5 and 6). The *optimal input modality* is chosen from the group of input modalities with the highest IC suitability score. The same principle applies to the selection of the *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 that has the next highest score. The process is repeated until the system is able to find a replacement modality that can be supported by currently available media devices.

When multimodality is found possible and optimal modalities are chosen, then supporting media devices are checked for availability. Using function g_1, the media group that support the chosen modality is 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 **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 **g_1,** that is $\forall$p: **Media Group,** $\exists$q: **Modality** $|\, p \rightarrow q \in g_1$. Using function **$g_2$**, the top-ranked media devices that belong to such media group are also identified. Given function $\mathbf{g_2}$, a media device **d**, priorities $\mathbf{p_1}$ and $\mathbf{p_2}$ where Priority: $£_1$ (positive numbers excluding zero), then finding the top-ranked device for a media group *m* is obtained as $\exists$**m:** Media group, $\forall$**d:** Media device, $\exists \mathbf{p_1}$**:** Priority, $\forall \mathbf{p_2}$**:** Priority $\vert$ d ● m $\rightarrow$ (d, p_1) $\in$ $g_2 \wedge$ (p1 < p2).

```
//Initialization
// Assumption: index i = 1 to 6 represent modalities
// VI_in, VO_in, M_in, VI_out, VO_out and M_out respectively
 for i = 1 to modality_max
    modality[i] = Null
end for
 //Evaluate IC suitability of individual modality
for i = 1 to modality_max do
     // Calculate modality's IC suitability score
     temp := 1.0;
     for j = 1 to parameter_max do
         // read suitability level of a modality with respect to parameter j
        case suitabilityLevel(j) of
              suitabilityLevel(j) = High:              score = 1.00
              suitabilityLevel(j) = Medium :         score = 0.75
              suitabilityLevel(j) = Low:               score = 0.50
              suitabilityLevel(j) = Inappropriate: score = 0.0
        end case
        temp = temp * score;
     end for
     finalScore := temp ^ (1/parameter_max)
     case finalScore of
          finalScore = 1.00:                 Suitability = High
          0.75 <= finalScore < 1.00:  Suitability = Medium
          0.50 <= finalScore < 0.75:  Suitability = Low
          finalScore < 0.50:                 Suitability = Inappropriate
     end case
    modality[i] = Suitability
end for
//check if multimodality is possible
if ((modality[1] != Inappropriate) OR (modality[2] != Inappropriate) OR
   (modality[3] != Inappropriate)) AND ((modality[4] != Inappropriate) OR
   (modality[5] != Inappropriate) OR (modality[6] != Inappropriate)) 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])
 else
 //multimodality is not possible
end if
```

Figure 2.2 Algorithm to determine modality's suitability to IC and if modality is possible.

Let there be a *media devices priority table* (MDPT) (see Tableau 2.1) which tabulates all media groups, and its set of media devices arranged by priority ranking. **T** = {T_1, T_2… T_{max_table}} is the set of MDPT's. The elements of table $T_n \in$ **T**, n = 1 to *max_table*, are similar to elements of function g_2. 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 media device may be different in each MDPT. In general, it is possible that two or more different context scenarios may be assigned to one common MDPT.

Tableau 2.1 Sample media devices priority table

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			

2.5 Context Learning and Adaptation

In concept, *machine learning* (ML) is about programming that optimizes an entity's performance from using sample data or past experiences. ML is important when human expertise does not exist hence learning rule is formulated from acquired data (Alpaydin

2004). A machine is said to have "learned" if its performance in doing a task improves with its experience (Mitchell 1997). In this work, the objective of adopting ML is that given an IC, the system determines the appropriate modalities and supporting media devices. Such knowledge is stored in a repository (called an exemplar) such that when the same case reoccurs, the system automatically implements multimodality with little or no human intervention.

System knowledge begins with the establishment of a priori knowledge which is related to an IC parameter. An example of such knowledge is shown in Tableau 2.2. As shown, the "user location" parameter is deduced from a sensor (i.e. a GPS). Initially, specific values of latitude and longitude are given specific meanings (a.k.a. conventions). When sample sensor readings are taken, the system then knows if the user is "at home", "at work" or "on the go". Also, the expert (i.e. end user) is required to supply the suitability score of each modality for each user location convention (see Tableau 2.2(b)). Hence, based on user location, the system can easily retrieve the suitability score of each modality.

In general, if a system is to become reliable in its detection of the suitability of modalities given a specific 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 the layered virtual machine for incremental definition of IC. When all the a priori knowledge are collected and grouped together, it forms a tree-like IC structure, as shown in Figure 2.3. Every new parameter is appended as a branch of either UC or EC or SC. Accordingly, the values or conventions of the parameter are identified along with the suitability scores of all types of modalities.

There are cases, however, when a certain parameter's value could nullify the importance of another 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 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 a case. A *case* is composed of these elements: (1) **the problem** – the IC in consideration, composed of UC, EC and SC parameters and their values, (2) **the solution** – the final IC suitability of each modality, and (3) **the evaluation** – the rate of relevance of the solution.

Tableau 2.2 A sample user context parameter – conventions and modalities selections

(a) User location convention table using GPS values

Convention No.	Latitude	Longitude	Meaning
1	<value11>	<value12>	At home
2	<value21>	<value22>	At work
3	not <value11> AND not <value21>	NOT <value12> AND NOT <value22>	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

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 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 it. In such a case, a new case with supervised learning is produced. The ML component adds the new case in its knowledge database. This whole learning mechanism is called *case-based reasoning with supervised learning*.

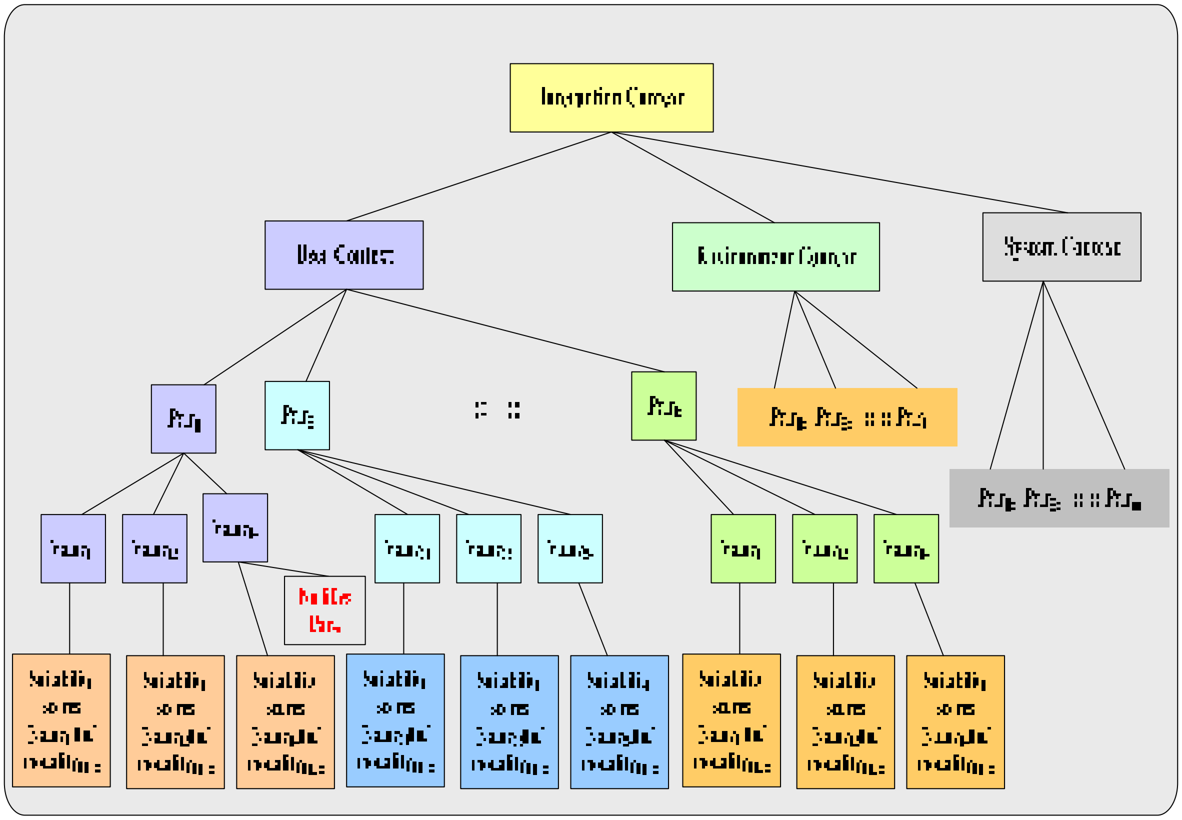


Figure 2.3 The structure of stored IC parameters.

Inspired by (Lajmi, Ghedira et al. 2007), we modify the 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 (i.e. subscript indicates which case is considered) is equal to their similarity in **UC**, **EC** and **SC** and is given by:

$$\begin{aligned} Sim(NC, MC) = {} & \frac{1}{3} Sim(UC_{NC}, UC_{MC}) + \\ & \frac{1}{3} Sim(EC_{NC}, EC_{MC}) + \frac{1}{3} Sim(SC_{NC}, SC_{MC}) \end{aligned} \tag{2.8}$$

The similarity between the **UC** of NC vs. the **UC** of MC is given by:

$$Sim(UC_{NC}, UC_{MC}) = \frac{\sum_{i=1}^{max_{NC}} Sim(UC\_Par_{i_{NC}}, UC_{MC})}{max(UC\_Par_{NC}, UC\_Par_{MC})} \tag{2.9}$$

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 $Sim(UC\_Par_{i_{NC}}, UC_{MC}) = max_{j=1\ to\ max_{MC}}\ Sim(UC\_Par_{i_{NC}}, UC\_Par_{j_{MC}})$ where $UC\_Par_{j_{MC}} \in$ UC_{MC} and $Sim(UC\_Par_{i_{NC}}, UC\_Par_{j_{MC}}) \in [0, 1]$ is the similarity between a specific UC parameter *i* of NC and parameter *j* of MC.

For the similarity measures of **EC** and **SC** of NC vs. MC, the same principle as Equation 2.9 must be applied, with the formula adjusted accordingly to denote **EC** and **SC**, respectively, yielding:

$$Sim(EC_{NC}, EC_{MC}) = \frac{\sum_{i=1}^{max_{NC}} Sim(EC\_Par_{i_{NC}}, EC_{MC})}{max(EC\_Par_{NC}, EC\_Par_{MC})} \tag{2.10}$$

$$Sim(SC_{NC}, SC_{MC}) = \frac{\sum_{i=1}^{max_{NC}} Sim(SC\_Par_{i_{NC}}, SC_{MC})}{max(SC\_Par_{NC}, SC\_Par_{MC})} \tag{2.11}$$

Equation 2.8 assumes that the weights of **UC**, **EC** and **SC** are equal (i.e. each is worth 33.3%). This figure can be easily adjusted to suit the need of the expert.

An ideal case match is a perfect match. However, a 90% match means that a great deal of context 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 search. Using MDPT, the media device that is next in priority is searched. The process is repeated until a replacement is found (see Figure 2.4). Formally, given a failed device **d** of priority $\mathbf{p_1}$, the specification for finding the replacement media device $\mathbf{d_{rep}}$ is $\exists m$**:** Media Group, $\forall \mathbf{d_{rep}}$**:** Media Device, $\exists \mathbf{p_1}$**:** Priority, $\forall \mathbf{p_2}$**:** Priority $\mid (p_1 = p_1 + 1) \wedge (p_1 < p_2) \wedge m \rightarrow (d_{rep}, p_1) \in g_2 \bullet d_{rep}$.

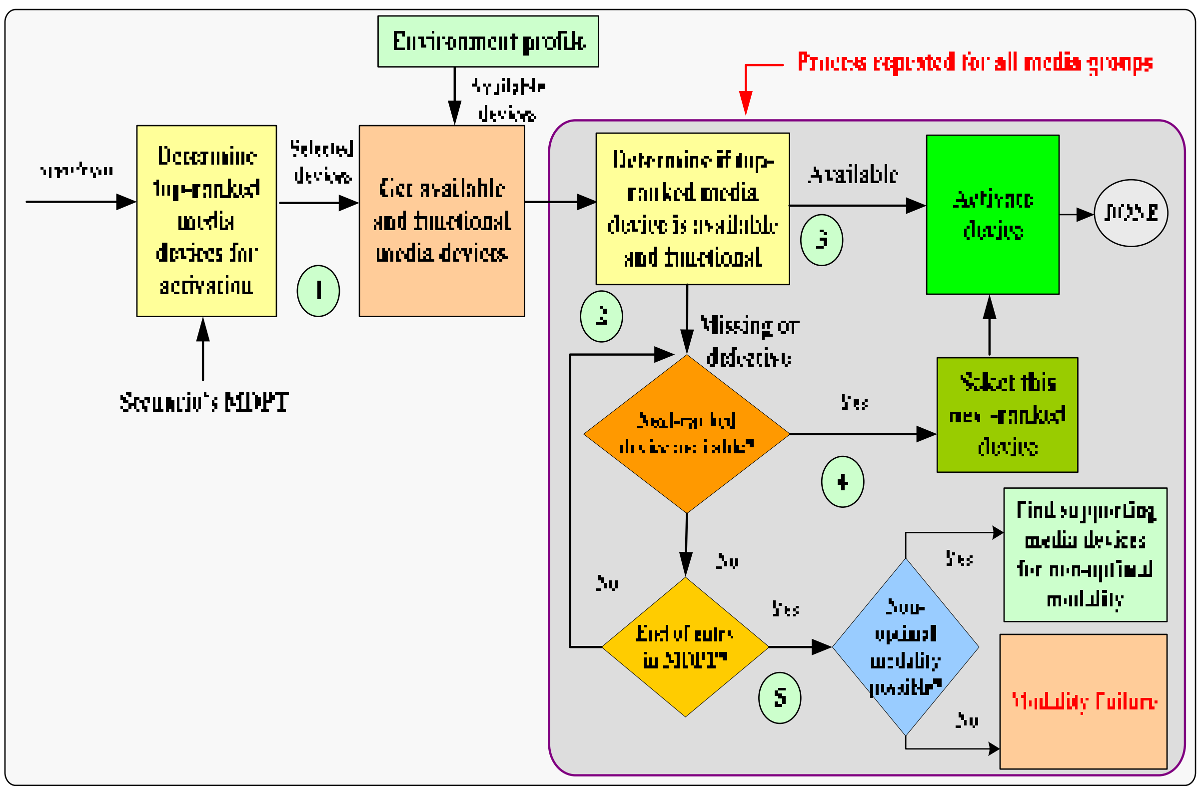


Figure 2.4 Algorithm for a failed device's replacement.

2.6 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.

2.7 References

Alpaydin, E. (2004). *Introduction to Machine Learning*. Cambridge, Massachusetts, MIT Press.

Coutaz, J., Crowley, J. L., et al. (2005). "*Context is key*." Communications of the ACM 48(3): pp. 49-53.

Dey, A. K. (2001). "*Understanding and Using Context* " Springer Personal and Ubiquitous Computing 5(1): pp. 4 - 7.

Dey, A. K. and Abowd, G. D. (1999). *Towards a Better Understanding of Context and Context-Awareness*. 1st Intl. Conference on Handheld and Ubiquitous Computing, Karlsruhe, Germany, Springer-Verlag, LNCS 1707.

Garlan, D., Siewiorek, D., et al. (2002). "*Project Aura: Towards Distraction-Free Pervasive Computing*." IEEE Pervasive Computing, Special Issue on Integrated Pervasive Computing Environments 21(2): pp. 22 - 31.

Lajmi, S., Ghedira, C., et al. (2007). "*Une méthode d'apprentissage pour la composition de services web*." L'Objet 8(2): pp. 1 - 4.

Mitchell, T. (1997). *Machine Learning*, McGraw-Hill.

Oviatt, S. L. (2000). "*Designing the User Interface for Multimodal Speech and Gesture Applications: State-of-the-art Systems and Research Directions*." Human Computer Interaction 15(4): pp. 263-322.

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

Rey, G. and Coutaz, J. (2002). *Le contexteur: une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte* Proceedings of the 14th French-speaking Conference on Human-Computer Interaction (Conférence Francophone sur l'Interaction Homme-Machine) IHM '02 Poitiers, France, ACM Press.

Ringland, S. P. A. and Scahill, F. J. (2003). "*Multimodality - The future of the wireless user interface*." BT Technology Journal 21(3): pp. 181-191.

Rubin, P., Vatikiotis-Bateson, E., et al. (1998). "*Audio-Visual Speech Processing*." Speech Communications 26(1-2): 1998.

Satyanarayanan, M. (2001). "*Pervasive Computing: Vision and Challenges*." IEEE Personal Communications 8(4): pp. 10-17.

Vasilakos, A. V. and Pedrycz, W. (2006). *Ambient Intelligence, Wireless Networking, Ubiquitous Computing*. USA, ArtecHouse.

Weiser, M. (1991). "*The computer for the twenty-first century*." Scientific American 265(3): pp. 94 - 104.

CHAPITRE 3

INFRASTRUCTURE OF A CONTEXT ADAPTIVE AND PERVASIVE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM

Manolo Dulva Hina [1,2], Amar Ramdane-Cherif [2], Chakib Tadj [1], and Nicole Levy[2]
[1] LATIS Laboratory, Université du Québec, École de technologie supérieure,
1100, rue Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, 45, avenue des États-Unis, Versailles Cedex, 78035 France

This article is published in the Journal of Information, Intelligence and Knowledge, Vol. 1, Issue 3, 2008, pp. 281-308, ISSN: 1937-7983.

Résumé

L'informatique multimodale multimédia diffuse (IMMD) a pour but de réaliser une informatique de n'importe où et n'importe quand en utilisant plusieurs formes d'interaction personne-machine. L'état de l'art sur les systèmes diffus et les solutions ne comprend pas d'applications liées à IMMD. Les interfaces multimodales courantes sont conçues avec des modes d'interaction personne-machine prédéfinis. Ces modes ne sont pas sélectionnés selon le contexte de l'utilisateur, son environnement et son système informatique. Suite à ce constat, cet article propose un système multimodal multimédia qui sélectionne la modalité basée sur le contexte d'interaction donné. Le système choisit une interface multimodale (ou unimodale) en fonction de ce contexte, les dispositifs multimédias disponibles et les préférences de l'utilisateur. Ce travail de recherche présente les défis associés au design de l'infrastructure de ce système et montre comment nous avons adressé ces défis. Ce travail est notre contribution qui a pour but de réaliser la multimodalité diffuse.

Mots clés: informatique multimodale multimédia; informatique diffuse; interaction personne-machine, système sensible au contexte.

Abstract

The aim of pervasive multimodal multimedia computing is to realize anytime, anywhere computing using various modes of human-computer interaction. The current state-of-the-art on pervasive systems and solutions, however, do not include applications that are related to pervasive multimodality. Also, the current multimodal interfaces were designed with pre-defined modes of human-machine interaction that were not chosen based on the given context of the user, of his environment and of his computing system. This paper addresses these weaknesses by proposing a pervasive multimodal multimedia computing system in which its modalities are chosen based on their suitability to the given interaction context. This same system chooses a multimodal (or unimodal) interface based on the given context, available media devices and user preferences. This paper discusses the challenges in designing the infrastructure of such computing system and illustrates how we addressed those challenges. This work is our contribution to the ongoing research that aims at realizing pervasive multimodality

Keywords: multimodal multimedia computing; pervasive computing; human-computer interaction; context-aware system

3.1 Introduction

Pervasive computing (also known as *ubiquitous computing*) aims at providing anytime, anywhere computing to a user working on a computing task using various software applications. This has been made possible because the infrastructure of pervasive computing (Satyanarayanan 2001) (Pahlavan and Krishnamurthy 2002), (Satyanarayanan 1990; Satyanarayanan 1996) does exist, an infrastructure that allows both wired and wireless computing and communications. *Pervasive multimodal multimedia computing*, on the other hand, aims to provide the infrastructure that would realize anytime, anywhere computing using various modes of human-computer interaction. Context awareness is an integral characteristic of pervasive computing systems. Context-awareness implies that the system is capable of adapting its operations to the most current context without explicit user intervention. Some context-aware systems have been developed to deliver pervasive healthcare, education, and communication, just to cite a few. Noticeably, however, there is something missing in the current state-of-the-art pervasive applications – one that would permit pervasive multimodal multimedia computing.

A *multimodal system*, in the context of human-computer interface, refers to the fusion of two (or more) input modes – such as speech, pen, gesture, gaze and head and body movements (Oviatt 2002). In contrast, a *unimodal recognition system* or *interface* involves only a single recognition-based technology, such as speech, pen and vision. Some of the current multimodal interfaces do fusion of speech and pen inputs, speech and lip movements, speech and manual gesturing, and gaze tracking and manual input. This implies, however, that the modes of human-machine interaction are already pre-defined from the very beginning. In most of these interfaces, there is an assumption that the setting is ideal (i.e. that there is barely a change in environment's context) and that the user is stationary. For mobile computing, a new conflicting requirement arises – that is, mobility requires that the computing terminal be light and small yet the system is required to deliver more advanced multimedia features. Such requirement suggests that keypads should possibly shrink or even vanish. To this end, the suitability of manual input modality also shrinks while the others – specifically, the vocal input modality – augments. Hence, the necessity for a wireless user interface (Ringland and Scahill 2003) in which speech is the mode for data input interaction. Most of the multimodal interfaces are not suitable to mobile users. Most of their modes for data input are pre-defined from the beginning and not selected based on their suitability to the environment's context. The drawback of such non-adaptive system is that if a context parameter changes (e.g. the environment becomes noisy) then the effectiveness of the interface is compromised (e.g. speech, as mode for data input, is not effective in a noisy workplace). To this end, we believe that an ideal pervasive system for multimodal application must have a wide-range selection of multimodal interfaces (aside from the regular unimodal interfaces) and at any given time, one particular interface is selected based on its suitability to the context of the user, of his environment and of computing system (henceforth called the *interaction context*).

Modality, in the context of multimodal multimedia computing, refers to the mode of human-computer interaction for data input and output. Given the current state-of-the-art systems and solutions, we have noted that the infrastructure to realize pervasive multimodality is missing. Such infrastructure is important as it is meant to be the backbone that (i) *implements either stationary or mobile computing*, (ii) *allows the invocation of modalities based on context suitability and availability of supporting media devices*, and (iii) *appropriately selects a unimodal or multimodal interface based on the given interaction context*. This paper, therefore, is intended to present the design of such infrastructure, the challenges involved in the design and our proposed solutions to address these challenges.

Apart from this introductory section, the rest of this paper is structured as follows. Section 2 discusses the works related to pervasive computing, multimodal system and the shortcomings of the current state-of-the-art systems and solutions, and presents our idea of addressing them. In section 3, we list down software engineering challenges related to this work and how we address them, and in the process present our contribution. In section 4, we explain the concepts of context and its relationship with modality and media devices. In section 5, we explain our proposed method for selecting appropriate modalities for the user interaction interface. Sample cases are cited in section 6 and the architectural framework of our proposed system is explained in section 7. Finally, we conclude this paper in section 8 and provide future works that we intend to do.

3.2 Related Work

Pervasive computing (aka ubiquitous computing) (Weiser 1991; Vasilakos and Pedrycz 2006) realizes anytime, anywhere computing; In doing so, a user's productivity increases as he can continue working on an interrupted computing task whenever and wherever he wishes. Context awareness, along with context management, heterogeneity, scalability, mobility, transparent user interaction, dependability and security are some software infrastructural issues for ubiquitous computing (da Costa, Yamin et al. 2008). Several applications of pervasive computing have been developed and implemented, among them are one for pervasive *healthcare* (Varshney 2003), *education* (Garlan, Siewiorek et al. 2002) and *communication* (Vallejos and Desmet 2007), just to cite a few. Missing, however, in the current state-of-the-art pervasive applications is the one that is related to multimodal multimedia computing.

In human-computer interface, multimodality refers to the fusion of two (or more) modes for data input. Since Bolt's original "*Put that there*" concept demonstration (Bouhuys 1995), which processed speech and manual pointing during object manipulation, some significant achievements in multimodal interface have surfaced, such as the one that combines *speech and pen* (Oviatt 2000), *speech and gestures* (Oviatt and Cohen 2000), *speech and lips movements* (Rubin, Vatikiotis-Bateson et al. 1998), *gaze and speech* (Zhang, Imamiya et al. 2004), *speech and mouse* (Djenidi, Ramdane-Cherif et al. 2002), *interface for Internet* (Dong, Xiao et al. 2000), and *wireless user interface* (Ringland and Scahill 2003). But *why build multimodal interface*? It is because it supports more transparent, flexible, efficient and expressive means of human-computer interaction. Multimodal interfaces are expected to be easier to learn and use, and are expected to accommodate more adverse user conditions than in the past (Oviatt 2002). The drawback to the current state-of-the-art multimodal interfaces, however, is that they are all designed with pre-defined modes for data input and without consideration to the varying conditions in the user's workplace, such as a workplace that becomes noisy. Most of these existing systems are also meant for users who are in stationary locations, and hence would become ineffective the moment the user becomes mobile.

Given the limitations cited above, we then envision a pervasive multimodal multimedia computing system. This new computing paradigm's infrastructure is characterized by the following features: (1) it is adaptive to the given interaction context – that is, the modalities for data input and output between the user and the machine are chosen based on their suitability to the given context, (2) that the modalities of interaction are chosen because they can be supported by available media devices, (3) that the chosen user interface is selected based on its suitability to the given context, the availability of supporting media devices and of user's preferences, (4) that the infrastructure supports both stationary and mobile computing, and (5) that the infrastructure itself is autonomic – specifically, it is self-optimizing, self-adaptive, self-configurable, self-optimizing (Horn 2001; Kephart and Chess 2001; Salehie and Tahvildari 2005). Due to space constraints, however, the design of the system's infrastructure as presented in this paper demonstrates only its *self-adaptive* features. This work is our contribution to the ongoing research in making anytime, anywhere computing using the most suitable form of human-computer interaction possible.

3.3 Requirements Analysis and Contribution

Here, we list down some software engineering challenges by posing specific technical challenges that need to be addressed. By answering these challenges, we do explain our novel contribution to the software engineering domain.

Our goal is to model a pervasive multimodal multimedia computing system. The design of such a system needs to address some key requirements cited below:

Requirement 1: Provide a generic representation of context. Provide a methodology that allows the incremental definition of context (i.e. add, delete, modify a context parameter).

Requirement 2: Provide the relationship between modality and context. 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 a context parameter and to the overall interaction context?

Requirement 3: Given a modality that is suitable to the given interaction context, provide a mechanism that chooses its supporting media devices. Then, given the modality and media devices selections, provide the mechanism that will determine the appropriate (unimodal or multimodal) user interface. What factors should be considered in the selection of a user interface?

The technical challenges are addressed by our proposed solutions given below:

Proposed solution to requirement 1: The term context, in this work, refers to *interaction context* (*IC*) which is the combined contexts of the user, his environment and his computing system. We provide a mathematical model that defines each parameter of interaction context and a virtual machine model for its implementation.

Proposed solution to requirement 2: The modalities for human-machine interaction are *manual*, *visual* and *vocal*, both to input and output data (details in next section). All context parameters are related to its domain of application, which in this case is multimodality. The relationship to consider is whether a specific modality is suitable to each *IC* parameter and if so, to what extent.

Proposed solution to requirement 3: We establish a relationship between modality and media devices. Given that any selected modality is deemed appropriate to the given context, it also follows that the selected media devices supporting the modality are also suitable to the given context. The interface of user-machine interaction is either unimodal or multimodal. When more than one interface is found suitable, then another factor to consider is the user's preference vis-à-vis user interface. Hence, we propose a priority ranking being assigned to the user's preference. The same priority ranking applies to the user's preferred modality and preferred media devices. The selection of user's interface, therefore, is based on appropriately selected modalities, available media devices and user's preferences.

3.4 Context, Multimodality and Media Devices

Here, we define context and provide its mathematical representation. We also illustrate how we can implement an incremental definition of context. Then, we derive the relationships that exist between context and multimodality and between multimodality and media devices.

3.4.1 Context Definition and Representation

In chronological order, the early definition of context includes that of (Schilit and Theimer 1994) in which context means the answers to the questions "*Where are you?*", "*With whom are you?*", and "*Which resources are in proximity with you?*". Schilit defined context as the changes in the physical, user and computational environments. This idea is taken afterwards by Pascoe (Pascoe 1998) and later on by Dey (Dey, Salber et al. 1999). Brown considered context as "*the user's location, the identity of the people surrounding the user, as well as the time, the season, the temperature, etc.*" (Brown, Bovey et al. 1997). Ryan defined context as the environment, the identity and location of the user as well as the time involved (Ryan, Pascoe et al. 1997). Ward viewed context as the possible environment states of an application (Ward, Jones et al. 1997). In Pascoe's definition, he added the pertinence of the notion of state: "*Context is a subset of physical and conceptual states having an interest to a particular entity*". Dey specified the notion of an entity: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves*" (Dey, Salber et al. 2001). This definition became the basis for Rey and Coutaz to coin the term interaction context: "*Interaction context is a combination of situations. Given a user U engaged in an activity A, then the interaction context at time t is the composition of situations between time t_0 and t in the conduct of A by U*" (Rey and Coutaz 2004).

We adopted the notion of "*interaction context*", but define it in the following manner: An interaction context, **IC** = $\{IC_1, IC_2, \ldots, IC_{max}\}$, 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 in 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:

$$IC_i = UC_k \otimes EC_l \otimes SC_m \tag{3.1}$$

where $1 \le k \le max_k$, $1 \le l \le max_l$, and $1 \le m \le max_m$, 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** itself is composed of parameters that describe the state of the user during the conduct of his activity. A specific user context k is a tuple composed of ($ICParam_{k1}$, $ICParam_{k2}$, … $ICParam_{kmaxk}$) and is given by:

$$UC_k = \bigotimes_{x=1}^{max_k} ICParam_{kx} \tag{3.2}$$

where $ICParam_{kx}$ = parameter of UC_k, k = the number of **UC** parameters, $k \in 1 .. max_k$. Using similar convention as that of **UC**, a specific instance of environment context EC_l and a specific instance of system context SC_m can be specified as follows:

$$EC_l = \bigotimes_{y=1}^{max_l} ICParam_{ly} \tag{3.3}$$

$$SC_m = \bigotimes_{z=1}^{max_m} ICParam_{mz} \tag{3.4}$$

3.4.2 Incremental Definition of Interaction Context

As stated, an instance of ***IC*** is composed of specific instances of ***UC**, **EC***, and ***SC***, which themselves are composed of one or more parameters. To realize the incremental definition of ***IC***, each of these parameters is introduced into the system, one at a time.

In our work, a virtual machine is designed to add, modify or delete one context parameter at a time, making ***IC*** parameters a reflection of the system's dynamic needs.

A *virtual machine* (VM) is software that creates a virtualized environment on computer platform so that the end user can operate the software. *Virtualization* is the process of presenting a group or subset of computing resources so that they can be accessed collectively in a more beneficially manner than their original configuration. In effect, a VM is an *abstract computer*; it accepts input, has algorithms and steps to solve the problem related to the input, and yields an output. The steps taken by the VM are its "*instructions set*" which is a collection of functions that the machine is capable of undertaking. A *layered VM* is a group of VM's wherein interaction takes place only between adjacent layers. *Layering* is a design choice to limit the propagation of errors within the concerned layer only during a modification of its functionality. Generally, in layered VM, the top layer refers to the interface that interacts with the end users while the bottom layer interacts with the hardware. Hence, Layer 0 is the bottom layer composed of sensors that generate some raw data representing the value needed by the topmost VM layer.

Figure 3.1 shows the functionality of such "*machine*". In general, the transfer of instruction command is top-down (steps 1 to 4). At Layer 0, the raw data corresponding to the *IC* parameters are collected for sampling purposes. The sampled data are then collated and interpreted, and the interpretation is forwarded to different layers bottom-up (steps 5 to 8).

The VM Layer 4 acts as the human-machine interface; its "*instruction set*" are the four functions found in Layer 3. The "*add parameter*", "*modify parameter*", and "*delete parameter*" are basic commands that manipulate the sensor-based context parameters while "*determine context*" yields the values of currently-defined parameters. VM Layer 2 is a "*library of functions*" that collectively supports Layer 3 instructions while Layer 1 is another "*library of functions*" that acts as a link between Layer 2 and Layer 0.

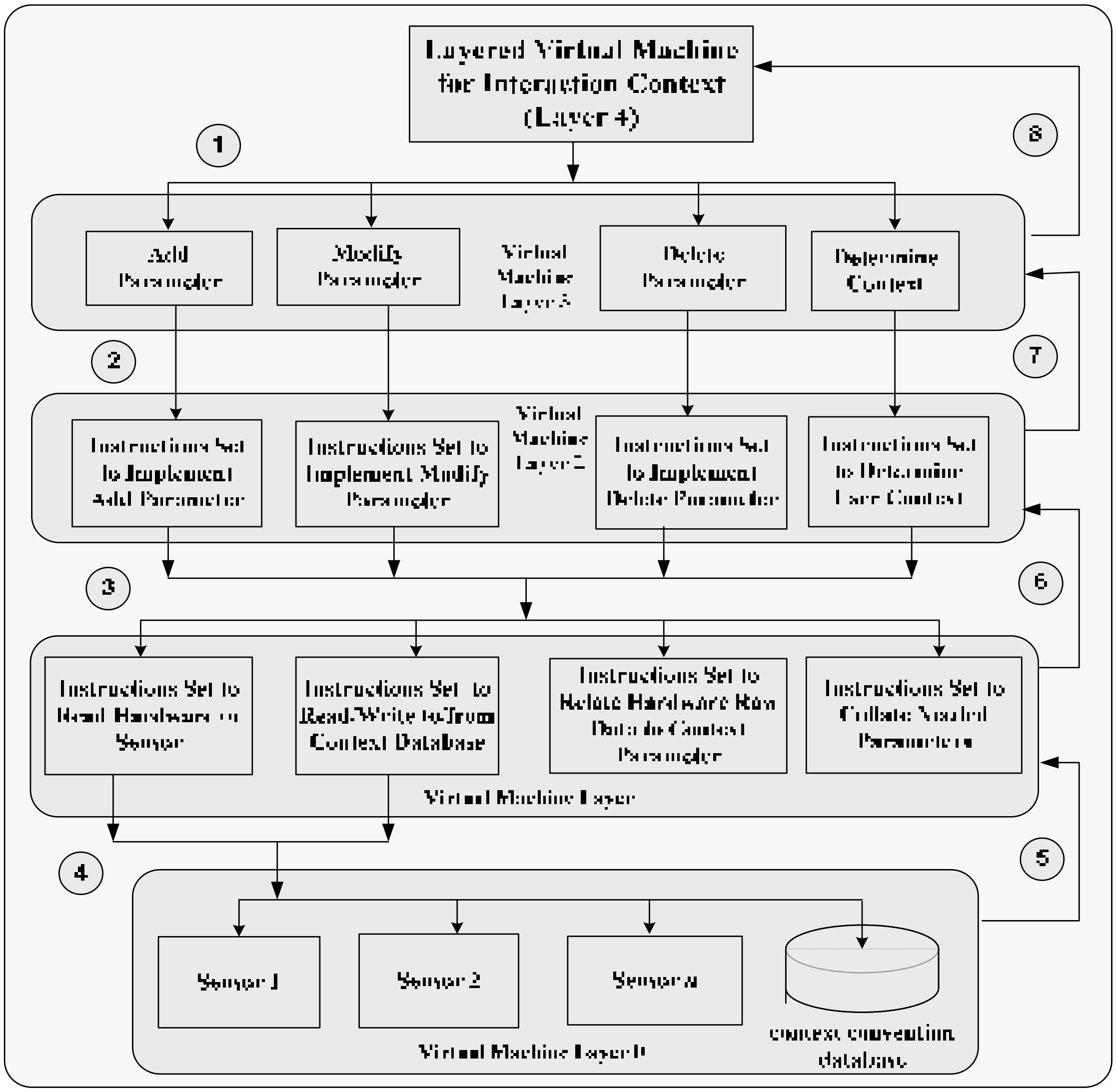


Figure 3.1 The design of a layered virtual machine for incremental interaction context.

3.4.2.1 Adding a Context Parameter

Consider using VM to add a specimen context parameter: the "*noise level*". See the design of VM's user interface in Figure 3.2.

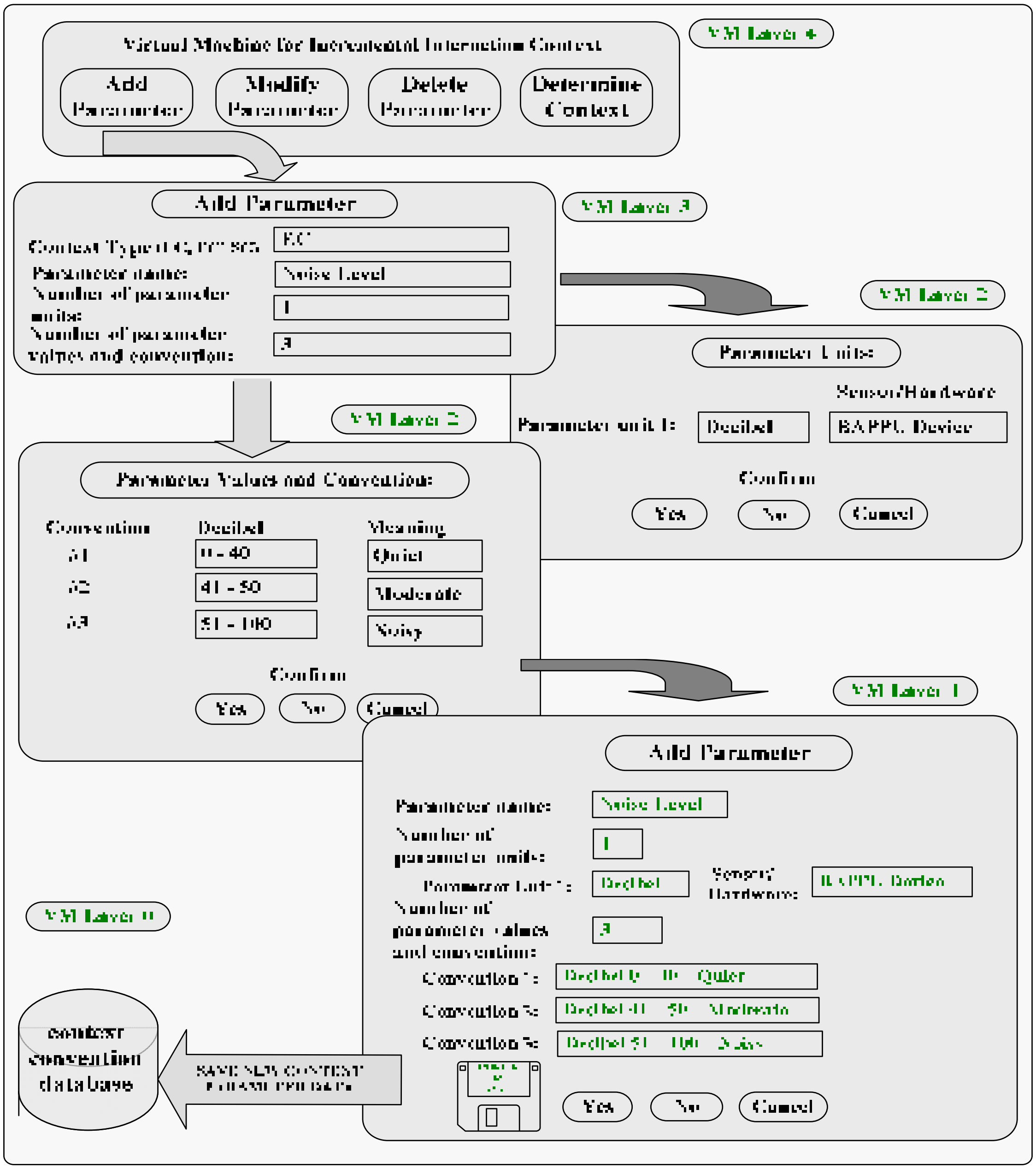


Figure 3.2 The interactions among layers to add a new (specimen only) context parameter: "Noise Level".

As shown in the diagram, upon invoking the VM user interface (i.e. Layer 4), the user chooses the "*Add Parameter*" menu. A window opens up which transfers the execution control to Layer 3. Then data entry takes place. To realize adding a new context parameter, at least four data entry functions must exist, namely: (i) *getting context type of the parameter*,

(ii) *getting name of the parameter*, (iii) *getting the parameter's number of units*, and (iv) *getting number of parameter values and conventions*. In Layer 3, the user inputs "*Noise level*" as parameter name, itself an EC parameter, "*1*" as parameter unit, and "*3*" as parameter values and conventions. When done, two new windows open up, one window at a time, that brings up the functionalities of Layer 2. For each parameter's unit, the VM receives input for the parameter's unit name and the sensor (or hardware) that supplies its raw data. As shown, the unit for "Noise level" is specified as "*decibel*" and the BAPPU noise measuring device (http://www.bappu.com/) (or any sensor for that matter that measures noise and supplies data to the computer) as the sensor supplying the data. When done, another Layer 2 window opens up for data entry of "*Parameter values and conventions*". In the diagram, the user specifies the value (range of decibels) that he considered is equivalent to "*quiet*", "*moderate*" and "*noisy*". When done, a window for Layer 1 opens up to save the newly-added parameter information. This function interacts directly with the hardware (i.e. the context convention database).

3.4.2.2 Modifying and Deleting a Context Parameter

The VM layers interaction involved in "*Modify parameter*" is almost identical to that of "*Delete Parameter*" function. The only thing extra in the former is a procedure that allows user to select the context parameter that should be modified. Other than that, everything else is the same. The processes involved in "*Delete Parameter*" menu are shown in Figure 3.3.

Upon menu selection, the execution control goes to Layer 3, demanding the user to specify the parameter for deletion (e.g. "*Noise level*" is chosen for deletion). Upon confirmation, the information about the parameter for deletion is extracted and read from database (transfer of control from Layer 2 to Layer 1 then to Layer 0). When the information for deletion is read, the control goes back to Layer 2 where such information is presented and a re-confirmation of its deletion is required. When parameter deletion is done, the control goes back to Layer 3 which presents the updated list of context parameters. An "*OK*" button click transfers the control back to Layer 4.

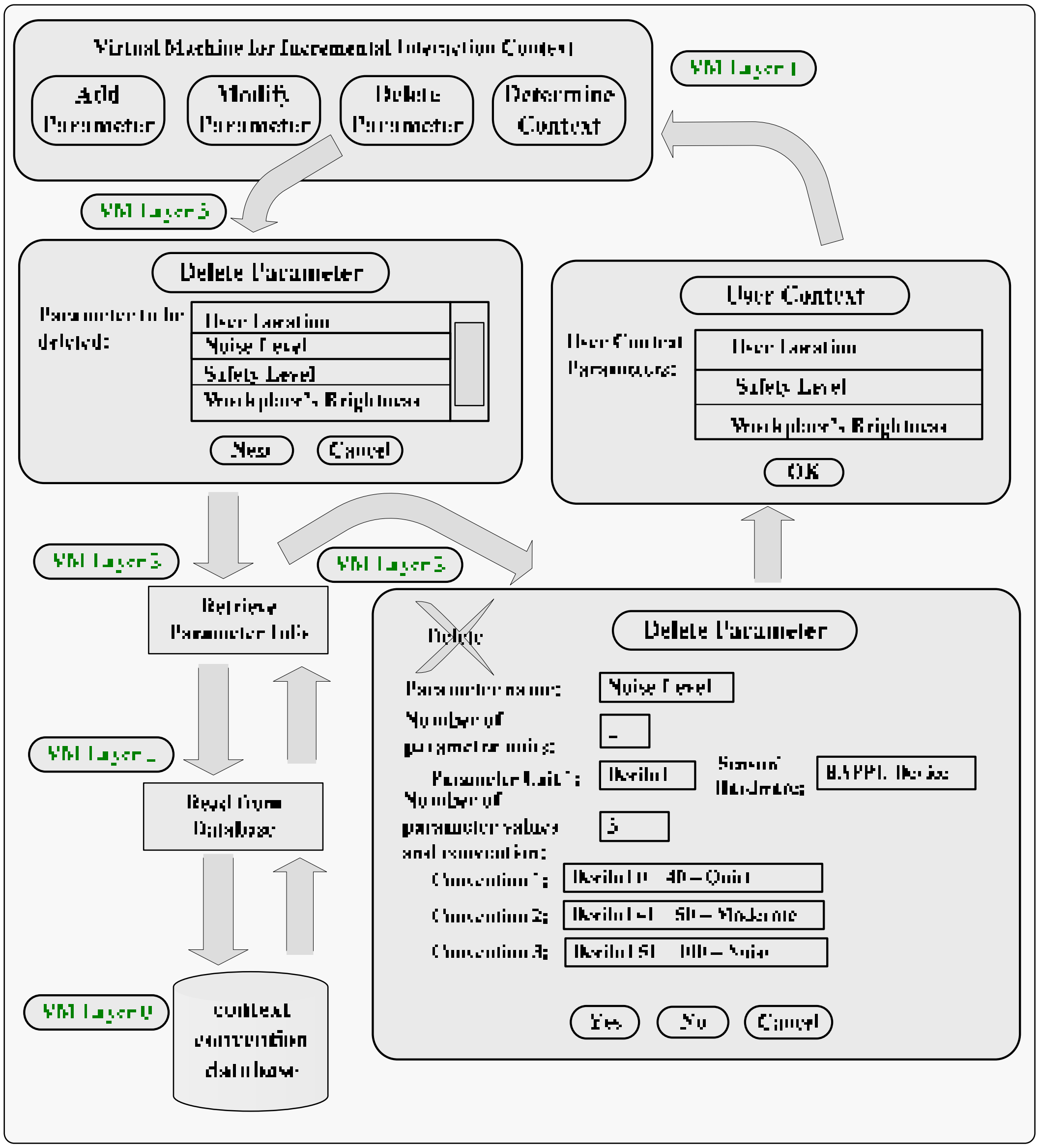


Figure 3.3 The VM layers interaction to realize "deleting a user context parameter".

3.4.2.3 Capturing the User's Current Context

The interactions of VM layers to "*Determine Context*" are shown in Figure 3.4. This is simulated using three specimen context parameters, namely (i) *the user location*, (ii) *the safety level*, and (iii) *the workplace's brightness*. When the user opts for this menu, the VM execution control goes to Layer 3. The function "*get user context*" creates threads equal to the number of parameters. Hence, this process produces thread "*get parameter 1*", assigned to detect user location, thread "*get parameter 2*" assigned to get the user's safety level, and the thread "*get parameter 3*" for the user's workplace's brightness (i.e. light intensity). The concepts involved are identical for each thread. Consider the case of "*user location*". The thread passes control to Layer 1 where a function takes sample data from a sensor (e.g. global positioning system, GPS, at http://www.rayming.com), attached to the user computer's USB port or value transmitted to the computer via wireless communication. In the VM design, the user can specify the number of raw data that need to be sampled and in what frequency (*n* samples per *m* unit of time). These *n* samples are then collated, normalized and interpreted.

For example, a specimen GPS data of 5 samples, taken 1 sample per minute, is shown in Figure 3.5. The data are then normalized (averaged), hence, the user's computer is located at 14°11' latitude and -120°57' longitude. Then, this value is interpreted using the convention values for user location parameter. Tableau 3.1 shows the format of the convention values of the specimen parameters. (Recall that the convention value of a parameter is created during the "*Add Parameter*" process.) Using Tableau 3.1-a, the interpretation identifies if the user (who uses the computer equipped with a GPS) is at home, at work or on the go.

Specimen parameter 2 *(the workplace's safety level*) is a function of (1) the person sitting in front of the computer, and (2) the presence of other people in the user's workplace. A camera with retinal recognition (http://www.informatik.uniaugsburg.de/~kimjongh/biometrics/retinal.pdf) may be used to identify the person sitting in the user's seat. The identification process would yield three values: (1) *User* – if the legitimate user is detected, (2) *Other* – if another person is detected, and (3) *Empty* – if no one is detected. Also, an infrared detector (http://www.globalsources.com/manufacturers/InfraredDetector.html) may be used to identify the presence of other person in front or in either side of the user. The identification process would yield two values: (1) *Image* – if at least one person is detected, and (2) *No Image* – if nobody is detected. (Note that the image and pattern recognition is not the subject of this work; hence, the detection process is not elucidated further in this paper.). The VM takes n = 5 samples, normalizes them and compares the result against the convention values in Tableau 3.1-b. The interpretation yields a result indicating if user's workplace is *safe*, *sensitive* or *risky*. This specimen parameter is useful for people working on sensitive data (e.g. bank manager) but can be irritating to a person working with teammates (e.g. students working on a project). Hence, this specimen parameter can be added or deleted on the user's discretion.

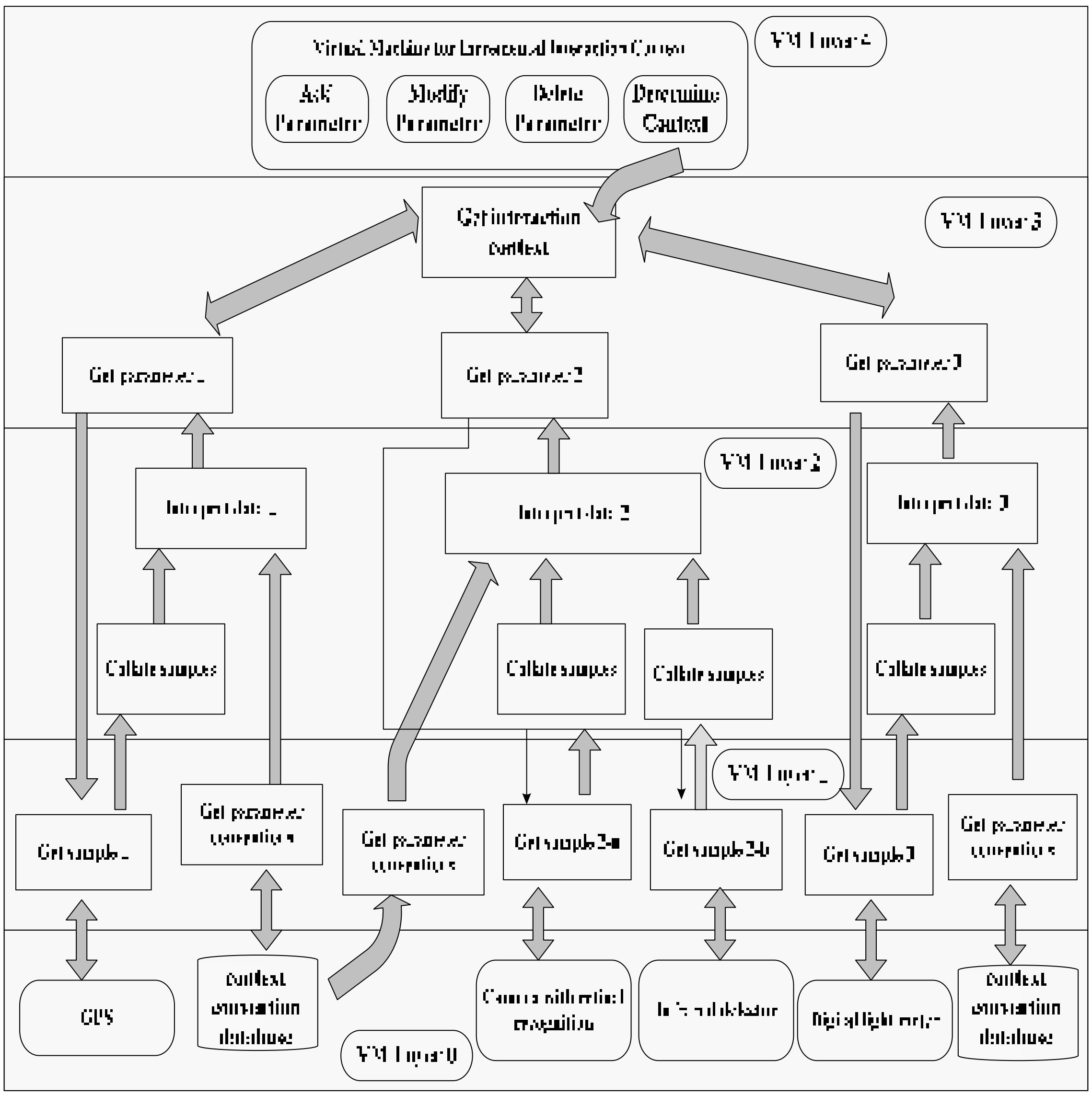


Figure 3.4 VM layers interaction in detecting the current interaction context.

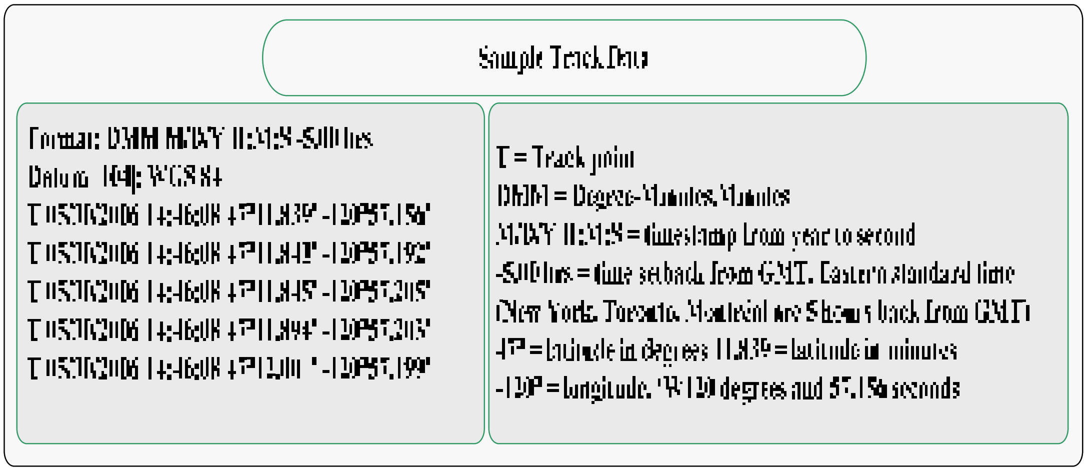


Figure 3.5 Sample GPS data gathered from Garmin GPSIII+.

Tableau 3.1 Sample conventions of the specimen sensor-based context parameters

(a) Convention format for user's location

Convention No.	Latitude	Longitude	Meaning
1	<value11>	<value12>	At home
2	<value21>	<value22>	At work
3	not <value11> AND not <value21>	NOT <value12> AND NOT <value22>	On the go

(b) Convention format for safety level in user's workplace

Convention No.	Detected in user's seat	Detected in user's workplace	Meaning
2	User	Image	Sensitive
1	User	No Image	Safe
3	Empty	Image	Sensitive
1	Empty	No Image	Safe
3	Other	Image	Risky
3	Other	No Image	Risky

(c) Convention format for light intensity in user's workplace

Convention No.	Foot-Candle	Meaning
1	<value-range1>	Bright
2	<value-range2>	Moderate
3	<value-range3>	Dark

The third specimen parameter (i.e. *workplace's brightness* in Tableau 3.1-c) detects the workplace's light intensity. Here, we can assume that a sensor measuring the light's intensity (http://www.gossen-photo.de/english/lichtmess_produkte.html) is attached to the computer's USB port. Its measurement unit, the *foot-candle*, is the number of "*lumens*" falling on a square foot of an inch; lumen is a unit of light used to rate the output of a bulb. For example, we may assume the following conventions in a user's workplace: (1) 0 – 9 foot candles = *dark,* (2) 10 – 20 foot-candles = *moderate*, and (3) 21 – 100 foot-candles = *bright.* The processes involved in sampling, collating and interpreting sensor data for parameter 3 is identical with the other 2 parameters mentioned above. Given the specimen parameters, when "*determine context*" is done, the output indicates (1) *if the user is at home, at work or on the go*, (2) *if user's workplace is safe, sensitive or risky*, and (3) *if the workplace's light intensity is bright, moderate or dark.*

3.4.3 Context Storage and Dissemination

In general, if a system is to obtain an accurate representation of the user's interaction context, then the system must be introduced to the most number of possible context parameters. As a context parameter is added to the system, the VM's context convention database forms a tree-like IC structure, as shown in generic format in Figure 3.6. Every new IC parameter is first classified as either UC or EC or SC parameter and is then appended as a branch of UC or EC or SC. Then, the conventions of the parameter are identified.

For the IC information to be propagated in a pervasive system, the data representation used is XML Schema which is based on XML (Hunter, Ayers et al. 2007). Figure 3.7(Left) illustrates the general XML format of a context parameter (i.e. name, units, source of raw data, and conventions) and Figure 3.7(Right) shows the various snapshots of windows involved in adding a parameter in the VM as implemented using Java programming language (Liang 2010).

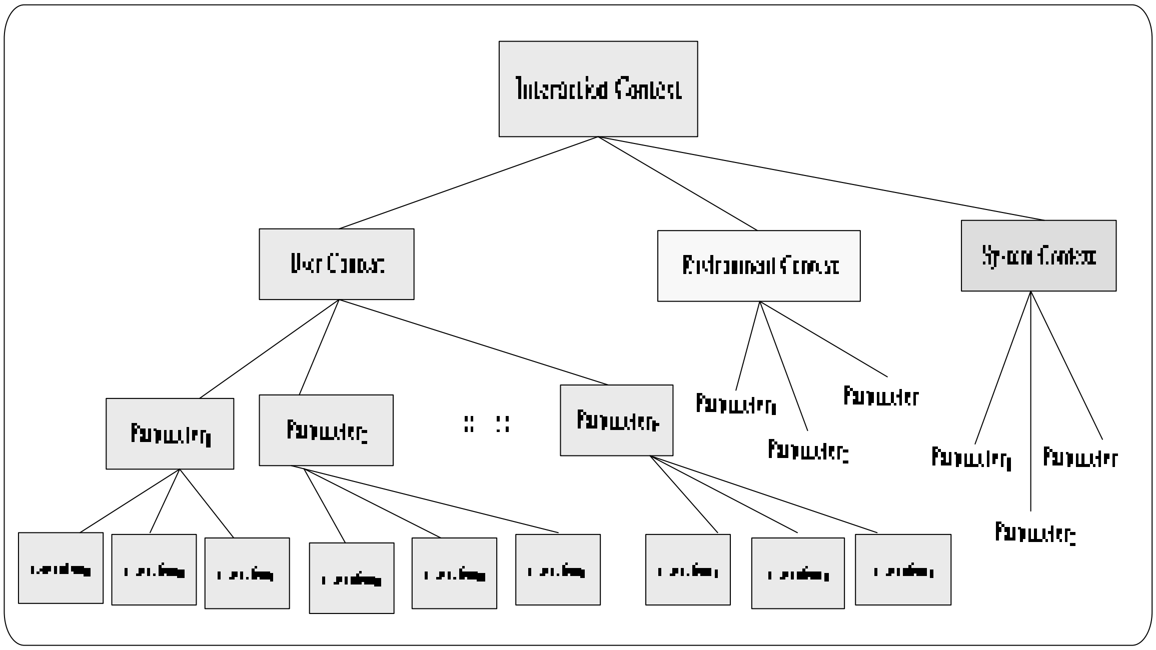


Figure 3.6 The structure of stored IC parameters.

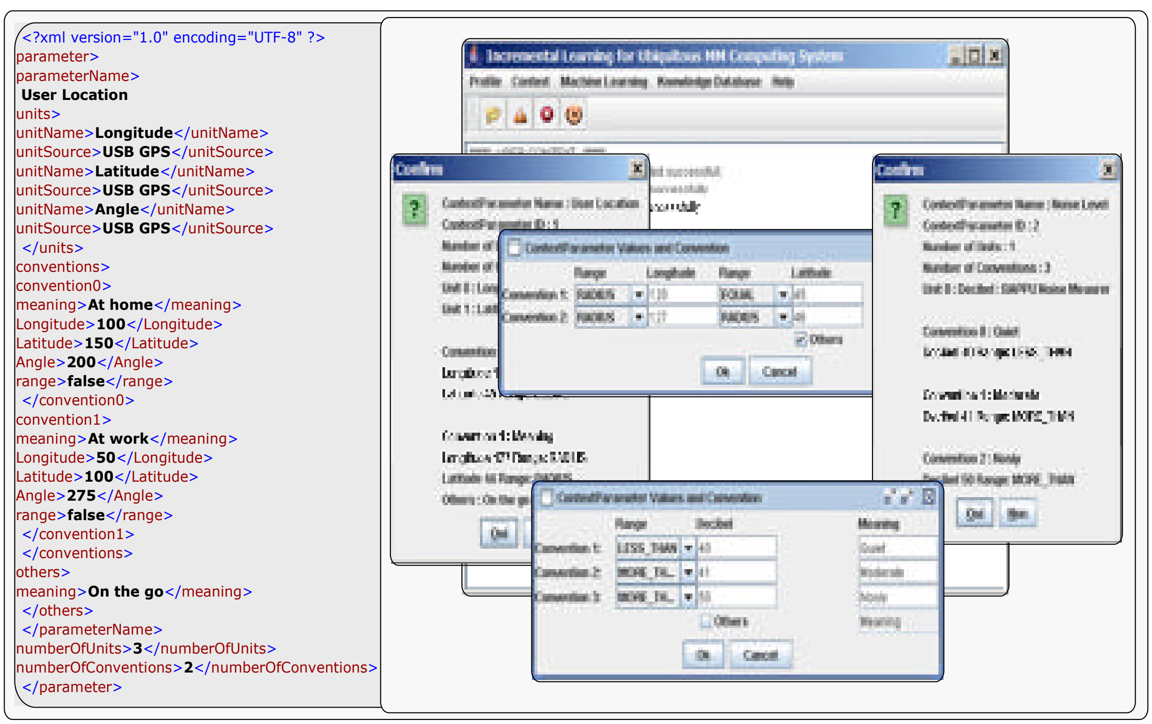


Figure 3.7 (Left) Sample context parameter in XML, (Right) snapshots of windows in adding a context parameter.

3.4.4 Measuring a Modality's Context Suitability

Multimodality refers to the selection of modalities 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 may only be realized if there is/are media devices that would support it. In this work, media refers to a set of physical interaction devices (plus some software supporting the physical devices). Using natural language processing as basis, we group modalities as follows: (i) Visual Input (VI_{in}), (ii) Vocal Input (VO_{in}), (iii) Manual/Tactile Input (M_{in}), (iv) Visual Output (VI_{out}), (v) Vocal Output (VO_{out}), and (vi) Manual/Tactile Output (M_{out}). Multimodality is possible if there is at least one modality for data input and at least one modality for data output.

Using Z language specification (Lightfoot 2001), let there be a set of input modalities and output modalities, as given by **INPUTMODE** ::= VI_{in} | VO_{in} | M_{in} and **OUTPUTMODE** ::= VI_{out} | VO_{out} | M_{out}. Let the relationship multimodality be a set of pairs of input and output modalities, as denoted by multimodality: **¡ (INPUTMODE ⊗ OUTPUTMODE)** where ¡ = power set which is the set of all subsets denotes power set and ⊗ = Cartesian product. At any given time, we can test if multimodality is possible by getting an instance of input and output modalities. Assume that x: **¡ INPUTMODE**, y: **¡ OUTPUTMODE**. Multimodality is possible if x and y forms a pair within the relationship multimodality and that neither x nor y is an empty set, that is, Possible((x,y)) ⇔ (x,y) ∈ multimodality $\wedge x \neq \emptyset \wedge y \neq \emptyset$.

Accordingly, media devices themselves are grouped as follows: (i) Visual Input Media (VIM), (ii) Visual Output Media (VOM), (iii) Oral Input Media (OIM), (iv) Hearing Output Media (HOM), (v) Touch Input Media (TIM) (vi) Manual Input Media (MIM), and (vii) Touch Output Media (TIM). The relationships that map modalities with media group and then the media group with media devices are shown in Figure 3.8.

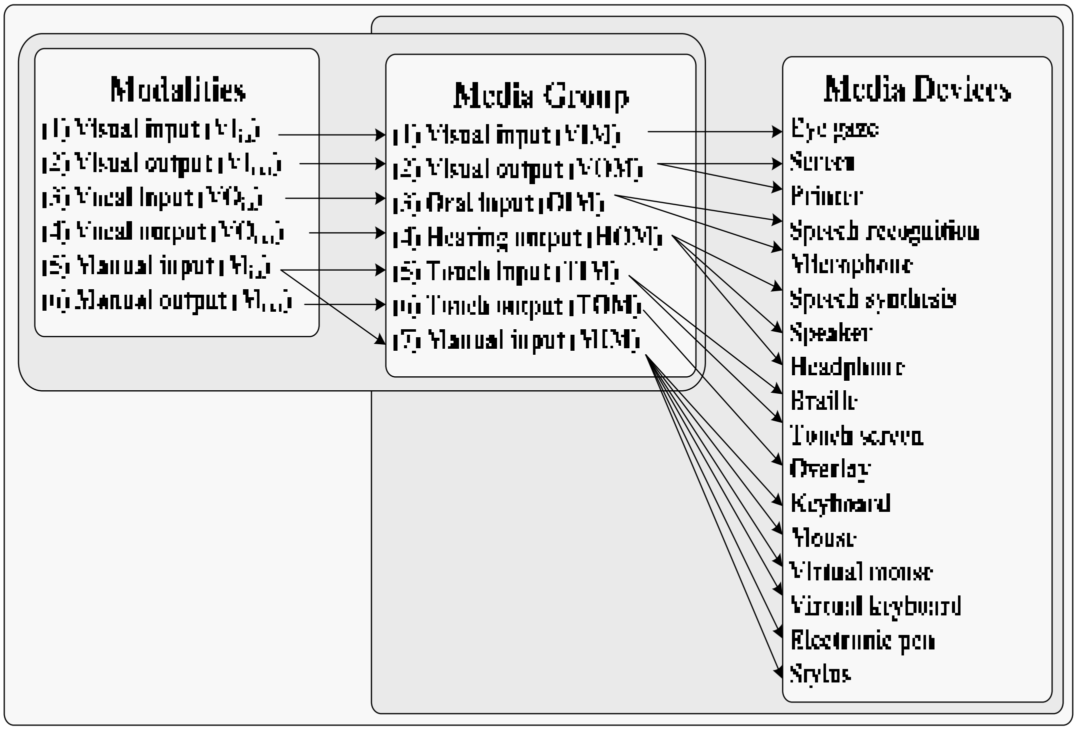


Figure 3.8 The relationship among modalities, media group and physical media devices.

To build a relationship between modalities and media devices, let there be a function g1 that maps a modality to a media group, given by **g1: MODALITY → MEDIAGROUP**. This is shown in Figure 3.8. There is often a case, however, when two or more media devices both belong to one media group. In such a case, devices selection is determined through their priority rankings. Hence, let there be another function g2 that maps a media group to a media device and its priority rank, denoted by **g2: MEDIAGROUP → (MEDIADEVICE**, £$_1$) where £$_1$ denotes an integer value greater than zero. Sample elements of these functions are:

g$_1$ = {(VI_{in},VIM), (VI_{out},VOM), (VO_{in},OIM), (VO_{out},HOM), (M_{in},TIM), (M_{in},MIM), (M_{out},TOM)}

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 the IC's individual parameters. Suitability measure is not binary, not just suitable or not suitable, because there are some cases wherein the extent of suitability lies in between. Hence, our suitability measures are *high*, *medium*, *low* and *inappropriate*. High suitability means that the modality in consideration is the preferred mode for computing; medium suitability means the modality is simply an alternative mode for computing, 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 the modality in consideration has n suitability scores. We then adopt the following conventions:

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:

$$SuitabilityScore_{modality} = \sqrt[n]{\prod_{i=1}^{n} context\_parameter_i} \tag{3.5}$$

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

$$Suitability_{modality} = \begin{cases} H \text{ if } SuitabilityScore_{modality} = 1.00 \\ M \text{ if } 0.75 \leq SuitabilityScore_{modality} < 1.00 \\ L \text{ if } 0.50 \leq SuitabilityScore_{modality} < 0.75 \\ I \text{ if } SuitabilityScore_{modality} < 0.50 \end{cases} \tag{3.6}$$

3.4.5 Selecting Context-Appropriate Modalities

Figure 3.9 shows the algorithm for determining the suitability of modalities to a given IC and if multimodality is possible (i.e. sub-section 3.4.4). Checking that multimodality is possible is done by determining that not all of input modalities (i.e. specified by indexes 1, 2 and 3) are scored "inappropriate", and so does for output modalities (i.e. specified by indexes 4, 5 and 6). The optimal input modality is chosen from the group of input modalities and is one with the highest IC suitability score. The same principle applies to the selection of the optimal output modality. Subject to the availability of media devices, an optimal modality is ought to be implemented; all others are considered optional. In the absence of supporting media devices, an alternative modality is chosen and is one that has the next highest score. Again, any alternative modality must be supported by available media devices. This process is repeated until the system is able to find a replacement modality that can be supported by currently available media devices.

```
//Initialization
// Assumption: index i = 1 to 6 represent modalities
// VIin, VOin, Min, VIout, VOout and Mout respectively
 for i = 1 to modality_max
    modality[i] = Null
end for
 //Evaluate IC suitability of individual modality
for i = 1 to modality_max do
      // Calculate modality's IC suitability score
      temp := 1.0;
      for j = 1 to parameter_max do
          // read suitability level of a modality with respect to parameter j
         case suitabilityLevel(j) of
                suitabilityLevel(j) = High:               score = 1.00
                suitabilityLevel(j) = Medium:          score = 0.75
                suitabilityLevel(j) = Low:                score = 0.50
                suitabilityLevel(j) = Inappropriate: score = 0.0
         end case
         temp = temp * score;
      end for
      finalScore := temp ^ (1/parameter_max)
      case finalScore of
            finalScore = 1.00:              Suitability = High
            0.75 ≤ finalScore < 1.00: Suitability = Medium
            0.50 ≤ finalScore < 0.75: Suitability = Low
            finalScore < 0.50:              Suitability = Inappropriate
      end case
     modality[i] = Suitability
end for
//check if multimodality is possible
if ((modality[1] != Inappropriate) OR (modality[2] != Inappropriate) OR
    (modality[3] != Inappropriate)) AND ((modality[4] != Inappropriate) OR
    (modality[5] != Inappropriate) OR (modality[6] != Inappropriate)) 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])
 else
  //multimodality is not possible
end if
```

Figure 3.9 Algorithm to determine a modality's suitability to IC and if multimodality possible.

3.4.6 Selecting Media Devices Supporting Modalities

When multimodality is possible and optimal modalities have been chosen, then supporting media devices are checked for availability. Using function g_1, the media group that supports the chosen modality can be identified. Given that **MODALITY** = {VI_{in}, VO_{in}, M_{in}, VI_{out},

VO_{out}, M_{out}} and **MEDIAGROUP** = {VIM, OIM, MIM, TIM, VOM, HOM, TOM} and that **g_1: MODALITY → MEDIAGROUP**, then formally, using Z language, we can specify that for every p which is a selected modality, there corresponds a media group q, wherein neither p nor q is an empty set, such that the ordered pair (p, q) is a member of set g_1, that is $\forall$ p: **MODALITY**; $\exists$ q: **MEDIAGROUP** $|$ $x \neq \emptyset \wedge y \neq \emptyset \bullet (p, q) \in g_1$.

By using function g_2, the top-ranked media devices that belong to the specified media group can also be identified. Given function g_2, a media device d, priorities p_1 and p_2 of type $£_1$, then the specification for finding the top-ranked device within the media group m is that the mapping between m and media device d with a priority ranking of d_1 does exist in function g_2 and that the numerical value of device's priority p_1 is less than p_2, (i.e. the lesser the numerical value, the higher is its priority ranking), that is, $\exists$ m: **MEDIAGROUP**; $\forall$ d: **MEDIADEVICE**; $\exists\, p_1$: $£_1$; $\forall\, p_2$: $£_1$ $|$ $d \bullet m \rightarrow (d, p_1) \in g_2 \wedge (p_1 < p_2)$.

Let there be a *media devices priority table* (MDPT) (see Tableau 3.2) which tabulates all media groups, and its set of media devices arranged by priority ranking. **T** = {T_1, T_2… T_{max_table}} is the set of MDPT's. The elements of table $T_n \in$ **T**, where $n \in 1 .. max\_table$, are similar to elements of function $\mathbf{g_2}$. 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 media device may be different in each MDPT. In general, it is possible that two or more different context scenarios may be assigned to one common MDPT.

When a new media device d_{new} is added or introduced to the system for the *first time*, the device is associated to a media group and is given a priority ranking r by the user. What happen to the rankings of other devices d_i, ($i \in 1 .. n$, and n = number of media devices) which are in the same media group as d_{new} in the MDPT? Two things may happen, depending on the user's selection. The first possibility is after having the new device's priority Priority(d_{new}) set to r then the priority of the other device i, ($1 \leq i \leq n$) denoted Priority(d_i), remains the same.

Tableau 3.2 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			

The second possibility is the priority rankings of all media devices (d_i) ranked r or lower are adjusted such that their new priority rankings are one lower than their previous rankings. Formally, this is specified as: $\forall i, \exists r: £_1; \forall d_i, \exists d_{new}$: **MEDIADEVICE** | (Priority($\mathbf{d}_{\mathbf{new}}$) = r $\wedge$ Priority($\mathbf{d}_i$) $\geq$ r) $\Rightarrow$ Priority($\mathbf{d}_i$)' = Priority($\mathbf{d}_i$) + 1.

3.5 Modalities in User Interaction Interface

Here, we wish to determine the selections of modality to be used in the user interaction interface, given that it is already known that multimodality is possible for implementation.

3.5.1 Media Groups and Media Devices

In general, the association between media group and media devices can be specified as:

$$VIM = eye\ gaze \vee gesture\ interpreter \tag{3.7}$$

$$OIM = speech\ recognition \wedge microphone \tag{3.8}$$

$$MIM = keyboard \vee Braille \vee pen \tag{3.9}$$

$$TIM = mouse \vee virtual\ mouse \vee touch\ screen \tag{3.10}$$

$$VOM = terminal\ screen \vee printer \tag{3.11}$$

$$HOM = speech\ synthesis \wedge (speaker \vee headset) \tag{3.12}$$

$$TOM = tactile\ keyboard \vee Braille \tag{3.13}$$

Note that the relationships cited above list down only limited number of commonly-used media devices. That said, these relationships can be modified easily and accordingly to include other media devices.

Given that **INPUT_MEDIA_GROUP** = {VIM, OIM, TIM, MIM}, then the power set (i.e. the set of all subsets) of this group is given by **¡(INPUT_MEDIA_GROUP)** = {{VIM}, {OIM}, {MIM}, {VIM, OIM}, {VIM, TIM}, {VIM, MIM}, {VIM, OIM, TIM}, {VIM, OIM, MIM}, {VIM, TIM, MIM}, {VIM, OIM, TIM, MIM}, {OIM, TIM}, {OIM, MIM}, {OIM, TIM, MIM}, {TIM, MIM}, {}}. These results indicate that as far as human-machine interaction interface is concerned, there can be four types of user interface. Note that, by definition, an interface is a function of input modalities only. Hence, the possible types of human-machine interaction interfaces are:

- **unimodal interface** – media devices (and supporting software) belonging to VIM, OIM, TIM and MIM can be used, but there is no fusion of data generated by one media with the data generated by another media (ex. speech, pen, vision)

- **bimodal interface** – there are 6 possible combinations of fusion of data generated by two media devices – that of {VIM, OIM}, {VIM, TIM}, {VIM, MIM}, {OIM, TIM}, {OIM, MIM}, and {TIM, MIM}. The current state-of-the-art multimodal interfaces fall in this category.

- **trimodal interface** – this interface allows the combination of data that are generated by three media devices into a new meaningful data; there are 4 possible selections, namely: {VIM, OIM, TIM}, {VIM, OIM, MIM}, {VIM, TIM, MIM}, and {OIM, TIM, MIM}

- **quadmodal interface** – this one would combine all types of input media altogether, {VIM, OIM, TIM, MIM}.

As far as research advancement (i.e. year 2008) is concerned, a user interface can only be unimodal or bimodal. There is no evidence that a trimodal interface, let alone a quadmodal one, exists, at least not yet.

3.5.2 The User Interface

Given that a unimodal or bimodal user interface is possible, then the system, in consultation with the user, decides the most suitable user interface. We believe that the selection of user interface that suits the user should be based on (i) the modalities and media groups that suit the given context (ii) the availability of media devices (and their supporting software) that would support the chosen modalities, and (iii) the availability of the preferred interface system or middleware within the user's computing system, and (iv) the user's preference on these interface as given by their priority rankings.

In order to determine whether the system will implement a unimodal or multimodal interface, let there be a *human-machine interaction interface priority table* (HMIIPT). This table contains important information related to the user's preferences, such as: (i) the priority ranking of multimodal and unimodal interface, (ii) the priority ranking of modalities within the interface, and (iii) the priority ranking of media devices that support a modality. See Tableau 3.3 for a sample HMIIPT.

Suppose that the user prefers a unimodal interface over a multimodal (actually, bimodal) one. Using HMIIPT, the system then determines the ranking assigned to each of the input modalities. Then the priority ranking of media devices supporting a preferred modality are taken from the media devices priority table (see Tableau 3.1). For multimodal interface, the user is also consulted in the priority rankings of all modality combinations/fusion. In the same manner, the priority of media devices supporting the multimodal fusion is also indicated in HMIIPT.

The selection process for optimal user interface modality uses the following functions to determine the score of each user interface mode. We take the result yielding the highest score as the optimal user interface modality:

$$\begin{aligned} \text{User Interface Modality Score} &= \text{User Interface Priority} \\ &\quad \times \text{Modality Priority} \times \text{Media Devices Priority} \end{aligned} \tag{3.14}$$

$$\text{User Interface Priority} = (m + 1 - p)/m \tag{3.15}$$

$$\text{Modality Priority} = (q + 1 - p)/q \tag{3.16}$$

$$\text{Media Device}_i \text{ Priority} = 1 - \sum_{1}^{i-1} (1/n) \tag{3.17}$$

Tableau 3.3 A sample human-machine interaction interface priority table (HMIIPT)

User Interface Modality by Priority			
Interaction Interface Priority	Priority of Modalities in Unimodal Interface	Priority of Modalities in Bimodal Interface	Priority of Media Devices in Bimodal Interface
1. Bimodal Interaction 2. Unimodal Interface	1. MIM 2. OIM 3. TIM 4. VIM	1. VIM and OIM	1. eye gaze – speech, 2. gesture – speech
		2. VIM and TIM	1. eye gaze – mouse, 2. eye gaze – virtual mouse, 3. gesture – mouse, etc.
	Priority of Media Devices in Unimodal Interface	3. VIM and TIM	1. eye gaze – keyboard, 2. eye gaze – Braille, 3. gesture – keyboard, etc.
		4. OIM and TIM	1. speech – mouse, 2. speech – touch screen, 3. speech – virtual mouse, etc.
	Same Priority Ranking as MDPT	5. OIM and MIM	1. speech – keyboard, 2. speech – pen, 3. speech – joystick, etc.
		6. TIM and MIM	1. mouse – keyboard, 2. mouse-Braille, 3. mouse-pen, etc.

such that for user interface priority, the variable m = number of types of user interface (i.e. unimodal, bimodal, etc. available in HMIIPT) and p = priority ranking as obtained from HMIIPT, $1 \leq p \leq m$. For modality's priority, the variable q = number of modality selections (i.e. available in HMIIPT) and p = priority ranking of the specified modality as obtained from HMIIPT, $1 \leq p \leq q$. For media devices priority, n = available media devices supporting the chosen modality and media group (see Equations (3.7) through (3.13)). Also, given the i^{th} device, where $1 \leq i \leq n$, then d_i = priority ranking obtained from MDPT and n = number of media devices supporting the same modality as the i^{th} device.

3.6 Sample Cases

Here, we simulate sample cases and accordingly apply the principles discussed in the previous sections.

3.6.1 Sample Case Using Specimen Interaction Context

Suppose that we are given the following interaction context: (i) *user context*: user location = at home, user handicap = none, (ii) *environment context:* noise level = quiet, safety factor = safe, (iii) *system context:* computing device = PDA. Suppose that the context convention database contains the conventions and suitability score of different modalities as shown in Tableau 3.4 through Tableau 3.8.What will be the optimal modality?

Tableau 3.4 User location conventions and suitability scores

(a) User location convention table using GPS values

Convention No.	Latitude	Longitude	Meaning
1	<value11>	<value12>	At home
2	<value21>	<value22>	At work
3	not <value11> AND not <value21>	NOT <value12> AND NOT <value22>	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

The given interaction context is **IC** = (c_1, c_2, c_3, c_4, c_5) = (1, 1, 1, 1, 3). The calculated final suitability scores of each type of modality are given below:

Visual Input= $[(H)(H)(H)(H)(L)]^{1/5} = [(1)(1)(1)(1)(0.50)]^{1/5} = 0.87$ = Medium suitability

Vocal Input = $[(H)(H)(H)(H)(H)]^{1/5} = [(1)(1)(1)(1)(1)]^{1/5} = 1$ = High suitability

Manual Input = $[(H)(H)(H)(H)(H)]^{1/5} = [(1)(1)(1)(1)(1)]^{1/5} = 1$ = High suitability

Visual Output = $[(H)(H)(H)(H)(H)]^{1/5} = [(1)(1)(1)(1)(1)]^{1/5} = 1$ = High suitability

Vocal Output = $[(H)(H)(H)(H)(H)]^{1/5} = [(1)(1)(1)(1)(1)]^{1/5} = 1$ = High suitability

Manual Output = $[(H)(H)(H)(H)(L)]^{1/5}=[(1)(1)(1)(1)(0.50)]^{1/5}= 0.87$ = Medium suitability

Tableau 3.5 User disability conventions and suitability scores

(a) User profile/disability convention

Convention No.	User Profile
1	Regular User
2	Deaf
3	Mute
4	Visually Impaired
5	Manuall Impaired

(b) Modality selection based on user profile/handicap

Type of Modality	User = Regular User	User = Deaf	User = Mute	User = Visually Impaired	User = Manually Impaired
Visual Input	H	H	H	L	H
Visual Output	H	H	H	L	H
Vocal Input	H	M	L	H	H
Vocal Output	H	L	M	H	H
Manual Input	H	H	H	H	L
Manual Output	H	H	H	H	L

Tableau 3.6 Workplace safety conventions and suitability scores

(a) The safety/risk factor convention table

Convention No.	Detected in user's seat	Detected in user's workplace	Meaning
2	User	Image	Sensitive
1	User	No Image	Safe
2	Empty	Image	Sensitive
1	Empty	No Image	Safe
3	Other	Image	Risky
3	Other	No Image	Risky

(b) Modality selection based on workplace safety level

Type of Modality	Safety level = Safe	Safety level = Sensitive	Safety level = Risky
Visual Input	H	M	L
Visual Output	H	M	L
Vocal Input	H	M	L
Vocal Output	H	M	L
Manual Input	H	M	L
Manual Output	H	M	L

Tableau 3.7 Noise level conventions and suitability scores

(a) Sample noise level convention

Convention No.	Decibel	Meaning
1	Less than 40	Quiet
2	40 to 50	Moderate
3	Greater than 50	Noisy

(b) Modality selection based on noise level

Type of Modality	Noise level = Quiet	Noise level = Moderate	Noise level = Noisy
Visual Input	H	H	H
Visual Output	H	H	H
Vocal Input	H	M	L
Vocal Output	H	H	M
Manual Input	H	H	H
Manual Output	H	H	H

Tableau 3.8 Computing device conventions and suitability scores

(a) Computing device convention

Convention No.	Computing Device
1	Desktop PC
2	Laptop
3	PDA
3	Cellular phone

(b) Modality selection based on user's computing device

Type of Modality	Computing device = Desktop PC	Computing device = Laptop	Computing device = PDA/Cellphone
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	L

Given this case, multimodality is possible (see section 3.4.4). The preferred input modality is either Vocal Input (VOin) or Manual Input (Min). The preferred output modality is Visual Output (VIout) or Vocal Output (VOout). All non-optimal modalities are considered optional. Using Figure 3.9 as visual reference, the media groups that suit the given interaction context are OIM, MIM, VOM, and HOM.

3.6.2 Sample Media Devices and User Interface Selection

Consider that the same user has the following media devices: OIM: speech recognition system, microphone, MIM: mouse, keyboard and electronic pen, VOM: screen and keyboard, and VOM: speech synthesis, speaker. Question: what is the most suitable human-computer interaction interface for the user?

To answer this question, we need to know the user preferences concerning media devices and user interface. Assuming that the data in the specimen MDPT and HMIIPT apply, then:

1. Bimodal Interface Priority = (2 + 1 – 1)/2 = 1
2. Modality Priority: OIM and MIM = (6 + 1 - 5)/6 = 0.33
3. Media Devices Priority: speech-keyboard = 1, speech-pen = 2/3
4. Unimodal Interface Priority = (2+1-2)/2 = 0.5
5. Modality Priority: MIM = 1, OIM = 0.75
6. Media Devices Priority: OIM: speech = 1 MIM: keyboard = 1, pen = 3/5

The calculation for bimodal interface follows:

1. speech-keyboard = 1*0.33*1 = 0.33
2. speech-pen = 1* 0.33*0.67 = 0.22

For, unimodal interface, the result is:

MIM: (a) keyboard = 0.5 * 0.75 * 1 = 0.375, and (b) pen = 0.5 * 0.75 * 0.6 = 0.225, and
OIM: speech = 0.5 * 0.75 * 1 = 0.375.

In this case, the system determines that the optimal user interface is a unimodal one in which the optimal input device/modality is the keyboard and speech.

3.7 Our Multimodal Multimedia Computing System

Here, we present the architectural framework of the pervasive multimodal multimedia computing system as well as the concept of ubiquity of system knowledge that is to be propagated to the network.

3.7.1 Architectural Framework

Our proposed system is conceived for two purposes: (1) to contribute to multimodal multimedia computing research and (2) to further advance self-adaptive computing system. To achieve the first goal, we develop the model that relates modality with user context, and associate media devices to support the implementation of the chosen modality. In the second goal, we advocate the propagation of knowledge, acquired through training, into the user's computing environment so that such knowledge can be used for system adaptation based on user's requirements and system's restrictions. The major components of our multimodal multimedia computing system are shown in Figure 3.10. The functionality of each component is given below:

- **The Task Manager Agent (TMA)** – manages user's profile, task and pertinent data and their deployment from a server to the user's computing device, and vice versa.

- **The Context Manager Agent (CMA)** – detects user context from sensors and user profile, and selects the modality and media apt for the context.

- **The History and Knowledge-based Agent (HKA)** – responsible for ML training and knowledge acquisition.

- **The Layered Virtual Machine for Interaction Context (LVMIC)** – detects sensor-based context and allows the incremental definition of context parameters.

- **The Environmental Manager Agent (EMA)** – detects available and functional media devices in the user's environment.

In the diagram, the user (Manolo) can work at home, log out, and still continue working on the same task at anytime and any place. Due to user's mobility, the variation in user's context and available resources is compensated by a corresponding variation in modality and media devices, and user interface selections.

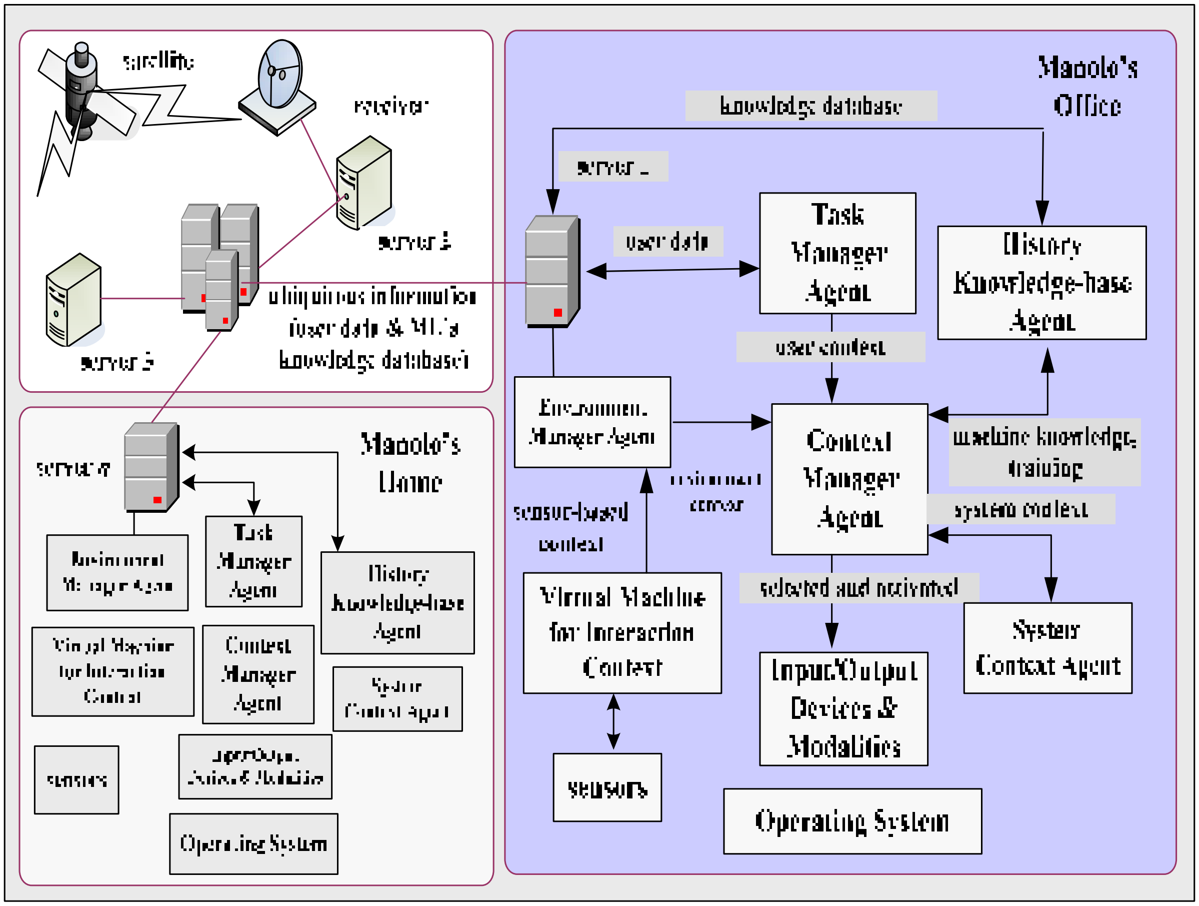


Figure 3.10 The architecture of a context-aware ubiquitous multimodal computing system.

3.7.2 Ubiquity of System Knowledge and Experience

The HKA is the component that incorporates incremental learning to the system. Its knowledge, for now, is concentrated on the system's ability to configure modalities, media devices and user interface based on the given context.

In the beginning, we assume that our system would have no knowledge whatsoever. Its initial knowledge is related to context parameter and its conventions, obtained incrementally as every relevant context parameter gets added using our VM. The system will acquire extra knowledge when HKA interacts with CMA (see Figure 3.11).

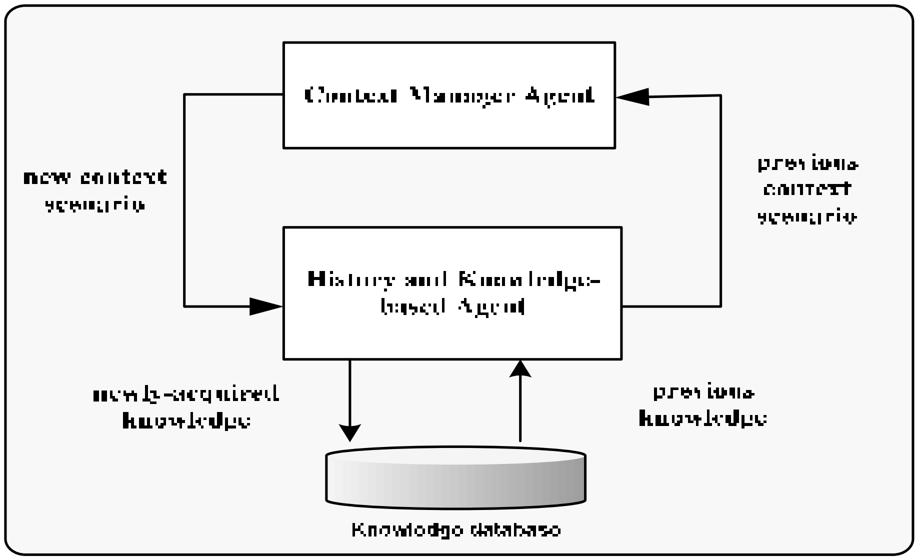


Figure 3.11 The History and Knowledge–based Agent at work

Through CMA, the HKA obtains information containing the user’s pre-condition scenario (i.e. the instance of interaction context), and accordingly determines the corresponding post-condition scenario (i.e. the selected modalities, media devices, and user interface). Each unique instance of interaction context (pre and post conditions) forms an entry in the knowledge database.

The system adds newly-acquired knowledge onto the database. Whenever a situation arrives that the system needs to do some decision or calculation based on the given instance of IC, the system first consults the database for any previous knowledge. If an exact match exists, then the system would simply implement the applicable set-up or post-condition scenario. If no match is found, then the system would have to do all the calculations as this is a new case. Afterwards, the result of the calculations becomes a newly acquired knowledge that is appended onto the database. Over time, the system would have enough knowledge to deal with almost all conceivable IC situations. Ideally, when the system could react automatically for almost every conceivable computing condition with minimum or no human intervention, we then say that the machine is “*intelligent*”.

To implement a pervasive system, the system's knowledge database needs to be transportable from one computing environment to another. A model of migration of user's task and of machine's knowledge is already demonstrated in our previous work, as applied to visually-impaired users (Awde, Hina et al. 2007). We intend to apply the same main principle into this system.

3.8 Conclusion and Future Works

In this paper, we noted that present-day applications of pervasive system do not include that of pervasive multimodality. The current state-of-the-art multimodal interfaces themselves are not apt for pervasive application since they are conceived and developed with pre-defined modalities from the very beginning. Indeed, there is a need for a pervasive multimodal multimedia computing system that would serve both stationary and mobile users, chooses modalities (i.e. mode of human-computer interaction to input and output data) that are appropriate to the given interaction context (i.e. user context + environment context + system context), chooses media devices to support the selected modality and chooses the optimal unimodal/multimodal interface based on user's preferences. This paper enumerates some software engineering challenges in designing the system's infrastructure and explains some details on how those challenges are addressed. We then show the architectural framework of our system and explain briefly our vision on the system's incremental knowledge acquisition.

Our future works include the system's dynamic configuration of its applications whenever computing resources become scarce. Also, more knowledge acquisition and machine learning algorithms need to be developed to make our system exhibits more autonomic computing system features (i.e. self-optimization, self-protection and self-healing).

3.9 Acknowledgement

This research work has been made possible through the funding provided by the *Natural Sciences and Engineering Research Council of Canada* (NSERC), the *Décanat des études* of

École de technologie supérieure and *La bourse de cotutelle* (Programme Frontenac) provided by the governments of Quebec and France.

3.10 References

Awde, A., Hina, M. D., et al. (2007). *Task Migration in a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users*. GPC 2007 - 2nd International Conference on Grid and Pervasive Computing, Paris, France, Springer Verlag.

Bouhuys, A. (1995). " *Induction of depressed and elated mood by music influences - the Perception of Facial Emotional Expressions in Healthy Subjects*." Journal of Affective Disorders 33(1): pp. 278 - 282.

Brown, P. J., Bovey, J. D., et al. (1997). "*Context-Aware Applications: From the Laboratory to the Marketplace*." IEEE Personal Communications 4(1): pp. 58 - 64.

Chen, H., Finin, T., et al. (2004). "*An Ontology for Context Aware Pervasive Computing Environments.*" Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review 18(3): pp. 197 - 207.

da Costa, C. A., Yamin, A. C., et al. (2008). "*Toward a General Infrastructure for Ubiquitous Computing.*" IEEE Pervasive Computing, Special Issue on Integrated Pervasive Computing Environments **7**(1): pp. 64 -73.

Dey, A. K., Salber, D., et al. (2001). "*A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications*." Human- Computer Interaction Journal 16(2-4): pp. 97-166.

Dey, A. K., Salber, D., et al. (1999). *The Conference Assistant: Combining Context-Awareness with Wearable Computing*. Digest of Papers, The 3rd International Symposium on Wearable Computers.

Djenidi, H., Ramdane-Cherif, A., et al. (2002). *Dynamic Multi-agent Architecture for Multimedia Multimodal Dialogs*. IEEE Workshop on Knowledge Media Networking.

Dong, S. H., Xiao, B., et al. (2000). "*Multimodal user interface for internet*." Chinese Journal of Computers 23(12): pp. 1270-1275.

Garlan, D., Siewiorek, D., et al. (2002). "*Project Aura: Towards Distraction-Free Pervasive Computing*." IEEE Pervasive Computing, Special Issue on Integrated Pervasive Computing Environments 21(2): pp. 22 - 31.

Hina, M. D., Ramdane-Cherif, A., et al. (2005). *A Context-Sensitive Incremental Learning Paradigm of a Ubiquitous Multimodal Multimedia Computing System*. WiMob 2005, IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Montreal, Canada.

Horn, P. (2001). *Autonomic Computing: IBM's Perspective on the State of Information Technology*, IBM Research.

Hunter, D., Ayers, D., et al. (2007). *Beginning XML*, Wrox Printing Inc.

Kephart, J. O. and Chess, D. M. (2001). *The Vision of Autonomic Computing*, IBM Thomas J. Watson Research Centre.

Liang, Y. D. (2010). *Introduction to Java Programming, 8th ed.*, Prentice Hall.

Oviatt, S. (2002). *Multimodal Interfaces: Handbook of Human-Computer Interaction*. New Jersey, USA, Lawrence Erbaum.

Oviatt, S. L. (2000). "*Designing the User Interface for Multimodal Speech and Gesture Applications: State-of-the-art Systems and Research Directions*." Human Computer Interaction 15(4): pp. 263-322.

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

Pahlavan, K. and Krishnamurthy, P. (2002). *Principles of Wireless Networks*.

Pascoe, J. (1998). *Adding Generic Contextual Capabilities to Wearable Computers*. 2nd IEEE Intl. Symposium on Wearable Computers Pittsburg, IEEE Computer Society.

Rey, G. and Coutaz, J. (2004). *The Contextor Infrastructure for Context-Aware Computing*. 18th European Conference on Object-Oriented Programming (ECOOP 04), Workshop on Component-Oriented Approach to Context-Aware Systems, Oslo, Norway.

Ringland, S. P. A. and Scahill, F. J. (2003). "*Multimodality - The future of the wireless user interface.*" BT Technology Journal 21(3): pp. 181-191.

Rubin, P., Vatikiotis-Bateson, E., et al. (1998). "*Audio-Visual Speech Processing.*" Speech Communications 26(1-2): 1998.

Ryan, N., Pascoe, J., et al. (1997). *Enhanced Reality Fieldwork: the Context-aware Archaeological Assisstant*. Computer Applications in Archaeology. V. Gaffney, M. v. Leusen and S. Exxon.

Salehie, M. and Tahvildari, L. (2005). *Autonomic Computing: Emerging Trends and Open Problems*. 2005 Workshop on Design and Evolution of Autonomic Application Software, St. Louis, Missouri, USA, ACM Press.

Satyanarayanan, M. (1990). "*Scalable, secure, and highly available distributed file access*." Computer 23(5): pp. 9-18.

Satyanarayanan, M. (1996). "*Mobile information access*." IEEE Personal Communications 3(1): pp. 26-33.

Satyanarayanan, M. (2001). "*Pervasive Computing: Vision and Challenges*." IEEE Personal Communications 8(4): pp. 10-17.

Schilit, B. and Theimer, M. (1994). "*Disseminating Active Map Information to Mobile Host*." IEEE Network 8(5): 22 - 32.

Vallejos, J. and Desmet, B. (2007). *Pervasive Communication: The Need for Distributed Context Adaptations*. ECOOP 2007 Workshop on Object Technology for Ambient Intelligence and Pervasive Systems. Berlin, Germany.

Varshney, U. (2003). "*Pervasive Healthcare*." Computer 36(12): pp. 138 - 140.

Vasilakos, A. V. and Pedrycz, W. (2006). *Ambient Intelligence, Wireless Networking, Ubiquitous Computing*. USA, ArtecHouse.

Ward, A., Jones, A., et al. (1997). "*A New Location Technique for the Active Office*." IEEE Personal Communications: pp. 42 - 47.

Weiser, M. (1991). "*The computer for the twenty-first century*." Scientific American 265(3): pp. 94 - 104.

Zhang, Q., Imamiya, A., et al. (2004). *A gaze and speech multimodal interface*, Hachioji, Japan, Institute of Electrical and Electronics Engineers Inc., Piscataway, USA.

3.11 Websites

Noise Measuring Device, ELK Company, at http://www.bappu.com/

Rayming Corp. USB GPS Receiver, at http://www.rayming.com

HTG Advance System's retinal recognition by a camera, at http://www.informatik.uniaugsburg.de/~kimjongh/biometrics/retinal.pdf

Infrared Detector, at http://www.globalsources.com/manufacturers/ InfraredDetector.html

USB Digital Light Meter", at http://www.gossen-photo.de/english/ lichtmess_produkte.html.

CHAPITRE 4

AUTONOMIC COMMUNICATION IN PERVASIVE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM

Manolo Dulva Hina [1,2], Chakib Tadj [1], Amar Ramdane-Cherif [2,3], and Nicole Levy[2]
[1] LATIS Laboratory, Université du Québec, École de technologie supérieure,
1100, rue Notre-Dame Ouest, Montréal, Québec, Canada H3C1K3
[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,
10-12 avenue de l'Europe, 78140 Vélizy, France

This article is published as a chapter in the book "Autonomic Communication", Vasilakos, A.V.; Parashar, M.; Karnouskos, S.; Pedrycz, W. (Eds.), 2009, XVIII, pp. 251- 283, ISBN: 978-0-387-09752-7.

Résumé

La communication autonomique dans un système informatique analyse l'élément individuel du système. Dans la communication personne-machine, le système autonomique assure ses services de façon autonome, ajustant le comportement de ses services pour s'adapter à la requête implicite ou explicite de l'utilisateur. Un système multimodal multimédia diffus vise à réaliser un calcul informatique n'importe où, n'importe quand. La communication autonomique inclut les protocoles qui sélectionnent les modalités et les dispositifs multimédias qui sont appropriés à un contexte d'interaction (IC) donné. Les modalités sont les modes de l'interaction (c.-à-d. les données d'entrée et de sortie) entre l'utilisateur et l'ordinateur. Les dispositifs multimédias sont les dispositifs physiques qui sont utilisés pour soutenir les modalités choisies. IC est une combinaison des contextes de l'utilisateur, de l'environnement et du système informatique. Dans cet article, nous présentons les protocoles de la communication autonomique impliqués dans la détection de l'IC et l'adaptation correspondante du système. Les défis techniques impliqués en formulant l'infrastructure de ce système incluent, notamment : (1) l'établissement du rapport entre l'IC et ses modalités appropriées, (2) la classification des dispositifs multimédias et son rapport avec la modalité, (3) la modélisation de l'IC et sa définition incrémental, et (4) l'établissement du mécanisme de la tolérance à la faute du système lorsque les dispositifs sont défectueux ou non disponibles. Deux aspects important sont mis de l'avant lors du design du paradigme de notre système : L'acquisition des connaissances de la machine ainsi que l'utilisation d'une machine virtuelle en couche pour la définition et la détection de l'IC.

Mots clés : système autonomique, système multimodal multimédia, système sensible au contexte, contexte incrémental, informatique diffuse.

Abstract

Autonomic communication in a computing system analyzes the individual system element as it is affected by and affects other elements. In the human-machine communication aspect, the autonomic system performs its services autonomously, adjusting the behavior of its services to suit what the user might request implicitly or explicitly. The pervasive multimodal multimedia (MM) computing system aims at realizing anytime, anywhere computing. Its autonomic communication includes the protocols that selects, on behalf of the user, the modalities and media devices that are appropriate to the given interaction context (IC). The modalities are the modes of interaction (i.e. for data input and output) between the user and the computer while the media devices are the physical devices that are used to support the chosen modalities. IC, itself, is the combined user, environment, and system contexts. In this paper, we present the autonomic communication protocols involved in the detection of IC and the MM computing system's corresponding adaptation. The technical challenges involved in formulating this system's infrastructure include, among others: (1) the establishment of relationship between IC and its suitable modalities, and the quantification of this suitability, (2) the classification of media devices and its relationship to modality, (3) the modeling of IC and its incremental definition, and (4) the establishment of the system's fault-tolerance mechanism concerning failed or missing media device. The heart of this paradigm's design is the machine learning's knowledge acquisition and the use of the layered virtual machine for definition and detection of IC.

Keywords: autonomic system, multimodal multimedia computing, context-aware system, incremental context, pervasive computing

4.1 Introduction

In autonomic communication, a system element learns, in every moment of its existence, about other elements and the world where it belongs through sensing and perception. In the human-machine aspect of autonomic communication, the system performs services autonomously. At the same time, it adjusts the behaviour of its services based on its learned perception of what the user might request, either implicitly or explicitly. This principle is applied in a specific application domain – the pervasive MM computing system. In such a system, various forms of modality for data input and output exist. Also, various media devices may be selected to support these modalities. Multimodality is possible if the mechanism for data input and output exists. Multimodality is important because it provides increased usability and accessibility to users, including those with handicaps. With multimodality, the strength or weakness of a media device is decided based on its suitability to a give*n* context[1]. For example, to a user in a moving car, an electronic pen and speech are more appropriate input media than that of a keyboard or a mouse. Multimodality can be further enhanced if more media devices (other than the traditional mouse-keyboard-screen combination) and their supporting software are made available. Socially, offering basic services using multimodality (e.g. a multimodal banking services) is not only wise but also contributes to the creation of a more humane, inclusive society as the weak, the old and the handicapped are given participation in using new technology.

Slowly, *pervasive computing*, also known as *ubiquitous computing*, (Weiser 1991; Satyanarayanan 2001; Vasilakos and Pedrycz 2006) which advocates anytime, anywhere computing is no longer a luxury but is becoming a way of life. For instance, healthcare is

1 Here, the word *context* signifies a generic meaning. Later, context will evolve to become an *interaction context*. Unless explicitly specified, context and interaction context may be used interchangeably.

adopting it. Soon, our personal and computing information would "*follow*" us and become accessible where and when we want them. This shall increase our productivity as we can continue working on an interrupted task as we desire. This has been made possible because the infrastructures for wired, wireless and mobile computing and communications (Satyanarayanan 1990; Satyanarayanan 1996; Pahlavan and Krishnamurthy 2002) do already exist.

Multimodality also involves fusion of two distinct data or modalities. For instance, the fusion of two or more temporal data, such as from a mouse and speech as in simultaneously clicking the mouse and uttering "*Put that there*" (Djenidi, Ramdane-Cherif et al. 2002; Djenidi, Benarif et al. 2004), is full of promise, further advancing multimodality. The fusion process, however, is still static – that is, the media and modality in consideration are pre-defined rather than dynamically selected. Also, the fusion process is not adaptive to the changes occurring in the environment (e.g. as in environment becomes noisy); hence over time, the effectiveness of a modality (e.g. vocal input) in the fusion process becomes unreliable. In general, it is unwise to predefine a chosen modality. A modality – whatever it may be – should be chosen only based on its suitability to a given context.

Context changes over time. Hence, context cannot be viewed as fixed nor should it be pre-defined. Instead, it should be defined dynamically based on the needs and requirement of a system. Our approach, therefore, is to define context by considering one context parameter at a time; a parameter may be added, deleted or modified as needed. This leads us to an *incremental context* where context becomes an attribute that is adaptive to the needs and requirements of a system. Context parameters may or may not be based on sensors data. For sensor-based context, we propose the adoption of *virtual machine* (VM). In this approach, the real-time interpretation of a context parameter is based on sampled data from sensor(s). The design of our layered VM for incremental user context is robust that it can be adopted by almost any system that uses sensor-based context.

Machine learning (ML) (Mitchell 1997) involves acquisition of knowledge through training or past experiences; this knowledge, when adopted, improves the system's performance. ML is the heart of this work. Our system's ML component is given:

1. Functions that (a) define the relationship between context and multimodality, and (b) define the relationships between modality and media group, and between media group and media devices,

2. Rules and algorithms that (a) determines the media device(s) that would replace the faulty one(s), and (b) the re-adaptation of the *knowledge database* (KD) when a new media device is introduced into the system. The acquired knowledge are then used to optimize configurations and for the system to exhibit fault-tolerance characteristics.

3. Case-based reasoning and supervised learning to find the appropriate solution to a new case, in consultation with the system's stored knowledge.

The rest of this paper is structured as follows: Section 2 surveys related works and highlights the novelty of this work. Section 3 essays on the technical challenges and our approach to address them. Section 4 is all about context – its definition, representation, storage and dissemination. Section 5 is about modalities, media devices and their context suitability. Section 6 is about our system's knowledge acquisition and the use of such knowledge to adapt to a given interaction context. The paper is concluded in Section 7.

4.2 Related Works

Some research works on multimodality include an interface for wireless user interface (Ringland and Scahill 2003), the static user interface (Oviatt and Cohen 2000), text-to-speech synthesis (Schroeter, Ostermann et al. 2000), and a ubiquitous system for visually-challenged user (Awdé, Hina et al. 2006). Some works on multimodality data fusion are the combined speech and pen inputs (Oviatt 2000), the combined speech and gestures inputs (Oviatt and

Cohen 2000) and the combined speech and lips movements (Rubin, Vatikiotis-Bateson et al. 1998). These are proofs that multimodality is possible, doable, and feasible. Compared with them, however, our work is one step further: it provides the infrastructure in which those above-mentioned works can be invoked anytime, anywhere.

Context counts heavily in determining the appropriate modalities for the user. Indeed, "*context is the key*" (Coutaz, Crowley et al. 2005). The evolution of context definitions, including Rey's definition for context-aware computing in (Dey and Abowd 1999; Dey 2001) and that of *contextor* (Rey and Coutaz 2002), is described in Section 4. The federation of context-aware perceptions (Coutaz, Crowley et al. 2005), and context-awareness in wearable computing (Dey and Abowd 1999) are some context-aware systems. Our contribution to the domain, however, is we take user's context and relate it to multimodality. While contextor is an interactive context-sensitive system, it does not, however, provide the mechanism to realize an ever-changing context. Our layered VM approach is more adaptable to an ever-changing environment. It has been proven that a layered VM/object-oriented design is an effective paradigm, as in Hughes Aircraft Company (Shumate 1988).

The user profile constitutes an integral part of user's context. Sample works on user profile analysis include (Antoniol and Penta 2004) and (Bougant, Delmond et al. 2003). Our work, however, differs because we consider user handicap as part of user's profile. This allows our work to cover a much larger spectrum of users. Finally, our objective is to assemble all these beneficial concepts to form a package for ubiquitous computing consumption. In Project Aura (Garlan, Siewiorek et al. 2002), the Prism model shows a user's moving aura (profile and task). In comparison, ours include not only the user's ubiquitous profile and task but also an acquired ML knowledge that goes with a moving user. Such knowledge is used in the detection of changes in IC and resources, and the system's adaptation to these changes by selecting the appropriate modalities and media devices. This work is intended to contribute to designing paradigms that explores the challenges in technologies that realize that vision wherein devices and applications seamlessly interconnect, intelligently cooperate and autonomously manage themselves, a.k.a. *autonomic communication*.

4.3 Contribution and Novel Approaches

Our vision is to enhance the use of multimodality through an infrastructure that realizes pervasive MM computing – intelligent, fault-tolerant, rich in media devices, and adaptive to a given context, acting on behalf of the user. To realize this, a system solution must address the key requirements given below.

Requirement 1: *Determine the suitability of various modalities to a given context.* It is first necessary to group modalities, and later determine the modalities that will allow user to input data and receive results, given a certain IC. What are these types of modality? What is the relationship between modalities and context?

Proposed Solution: Modality can be grouped into two classes: input modality and output modality. Within input modality, there exists the visual input, the vocal input, and the manual input. Similarly, within output modality, possible options are visual output, vocal output and manual output. There must be an input modality and an output modality if modality is to be realized. Given an IC, a modality has some degree of suitability to it. Such suitability is not only binary (very suitable or not suitable at all) but also includes something in between – medium and low suitability. Numerical value for suitability can be assigned as follows: High suitability = 100%, Inappropriate = 0. Medium and low suitability should have value in between this range. To relate modality to IC, each type of modality gets a suitability score for every context parameter. The final suitability to an IC is the normalized product of suitability scores on individual parameter.

Requirement 2: *Provide a relationship between modality and media devices that are invoked to realize modality. Given that various media devices exist, then provide a classification of media where all devices could fit.* What should be a generic media classification so that all media devices – presently known and all those that will come in the future – would fit in?

What would be the basis of such classification? In which category should, for example, a mouse belongs? What about the eye gaze, etc.?

Proposed Solution: *Media devices may be grouped in the same way as modalities.* The media classification is based on man's *natural language processing;* man transmits and receives information through his five senses (e.g. hearing, tasting, etc.) and voice. Therefore, media category is based on what body part uses the media device to generate data input, and the body part that consumes data output from the device. For example, a mouse is a manual input device, and so is the keyboard. A Braille terminal for the visually-impaired user is an example of a touch output device. An eye gaze is a visual input device.

Requirement 3: *Determine the parameters that would constitute a context, but since context changes over time then provide a mechanism that allows user to modify (add, change, delete) parameters on the fly.* A mobile user who changes environment over time does not have a fixed context; hence defining a fixed set of parameter that forms the context is incorrect. How do we declare the parameters of a context? Also, if modification of parameters is necessary, what mechanism should be used to effect such modification without producing a ripple effect into the system?

Proposed Solution: *An IC is the combined user, environment and system contexts, each of which is composed of one or more parameters. Our layered VM for incremental IC is a robust "machine" that can be adapted to any system and in which parameter modification can be done on the fly with minimum system ripple effect.* Also, the context parameter consideration in our layered VM is gradual or incremental. In effect, IC is defined based on the needs of the user.

Requirement 4: *Provide a self-healing mechanism that provides replacement to a faulty media device, and an orderly re-organization if a new device is introduced into the system for the first time.* If two or more media devices are classified as members of the same media group, which one would be given priority in a specific context? What are the guidelines for such priority ranking? If the chosen media device is faulty (missing or defective), how do we determine its replacement? If a new media device is introduced for the first time, how would it affect the priority ranking of other media devices in the same group?

Proposed Solution: *Through training, our ML system acquires knowledge for user context detection, determining the suitable modality, determining the appropriate media group and devices. The same system includes knowledge on which devices could replace the defective ones.* The policy of replacement is based on the media devices availability and priority rankings. For example, the devices that are used in usual configuration are given higher priority than those that are not used regularly. The ML training includes user participation that guides the system to recognize positive examples which form system knowledge.

4.4 The Interaction Context

This section discusses context – the evolution of its definition, its representation, capture, storage and dissemination.

4.4.1 Context Definition and Representation

In chronological order, early definition of context includes that of Schilit's (Schilit and Theimer 1994) in which context is referred to the answer to the questions "*Where are you?*", "*With whom are you?*", and "*Which resources are in proximity with you?*". Schilit defined context as the changes in the physical, user and computational environments. This idea is taken again by Pascoe (Pascoe 1998) and later on by Dey (Dey, Salber et al. 1999). Brown considered context as "*the user's location, the identity of the people surrounding the user, as well as the time, the season, the temperature, etc.*" (Brown, Bovey et al. 1997). Ryan defined context as the environment, the identity and location of the user as well as the time (Ryan, Pascoe et al. 1997). Ward viewed context as the possible environment states of an application (Ward, Jones et al. 1997). In Pascoe's definition, he added the pertinence to the notion of state: "*Context is a subset of physical and conceptual states having an interest to a particular* *entity"*. Dey specified the notion of an entity: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and application themselves"* (Dey 2001). This definition became the basis for Rey and Coutaz to coin the term interaction context: "*Interaction context = combination of situations. Given a user U engaged in an activity A, then the interaction context at time t is the composition of situations between time t and t in the conduct of A by U"* (Rey and Coutaz 2004).

We adopted the notion of "*interaction context*", but define it in the following manner: An *interaction context*, **IC** = {IC_1, IC_2,…, 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:

$$IC_i = UC_k \otimes EC_l \otimes SC_m \tag{4.1}$$

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 describing the state of the user during his activity. A specific user context *k* is given by:

$$UC_k = \bigotimes_{x=1}^{max_k} ICParam_{kx} \tag{4.2}$$

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

$$EC_l = \bigotimes_{y=1}^{max_l} ICParam_{ly} \tag{4.3}$$

$$SC_m = \bigotimes_{z=1}^{max_m} ICParam_{mz} \tag{4.4}$$

4.4.2 The Virtual Machine and the Incremental Interaction Context

As stated, an instance of **IC** is composed of specific instances of **UC**, **EC**, and **SC**, which themselves are composed of parameters. These parameters are introduced to the system, one at a time. In our work, a virtual machine is designed to add, modify or delete one context parameter, making the **IC** parameters reflective of the system's dynamic needs.

A *virtual machine* (VM) is software that creates a *virtualized* environment on computer platform so that the end user can operate the software. *Virtualization* is the process of presenting a group or subset of computing resources so that they can be accessed collectively in a more beneficially manner than their original configuration. In effect, a VM is an *abstract computer*; it accepts input, has algorithms and steps to solve the problem related to the input, and yields an output. The steps taken by the VM are its "*instructions set"* which is a collection of functions that the machine is capable of undertaking. A *layered VM* is a group of VM's wherein interaction is only between layers that are adjacent to one another; the layering is a design choice to contain the errors within the concerned layer only during a modification of its functionality. Generally, the *top layer* refers to the interface that interacts with the end users while the *bottom layer* interacts with the hardware. Hence, *Layer 0* is a collection of sensors (or machines or gadgets) that generate some raw data representing the value needed by the topmost VM layer. Figure 4.1 shows the functionality of such "*machine*". In general, the transfer of instruction command is top-down (steps 1 to 4). At

Layer 0, the raw data corresponding to the IC are collected for sampling purposes. The sampled data are then collated and interpreted, and the interpretation is forwarded to different layers bottom-up (steps 5 to 8).

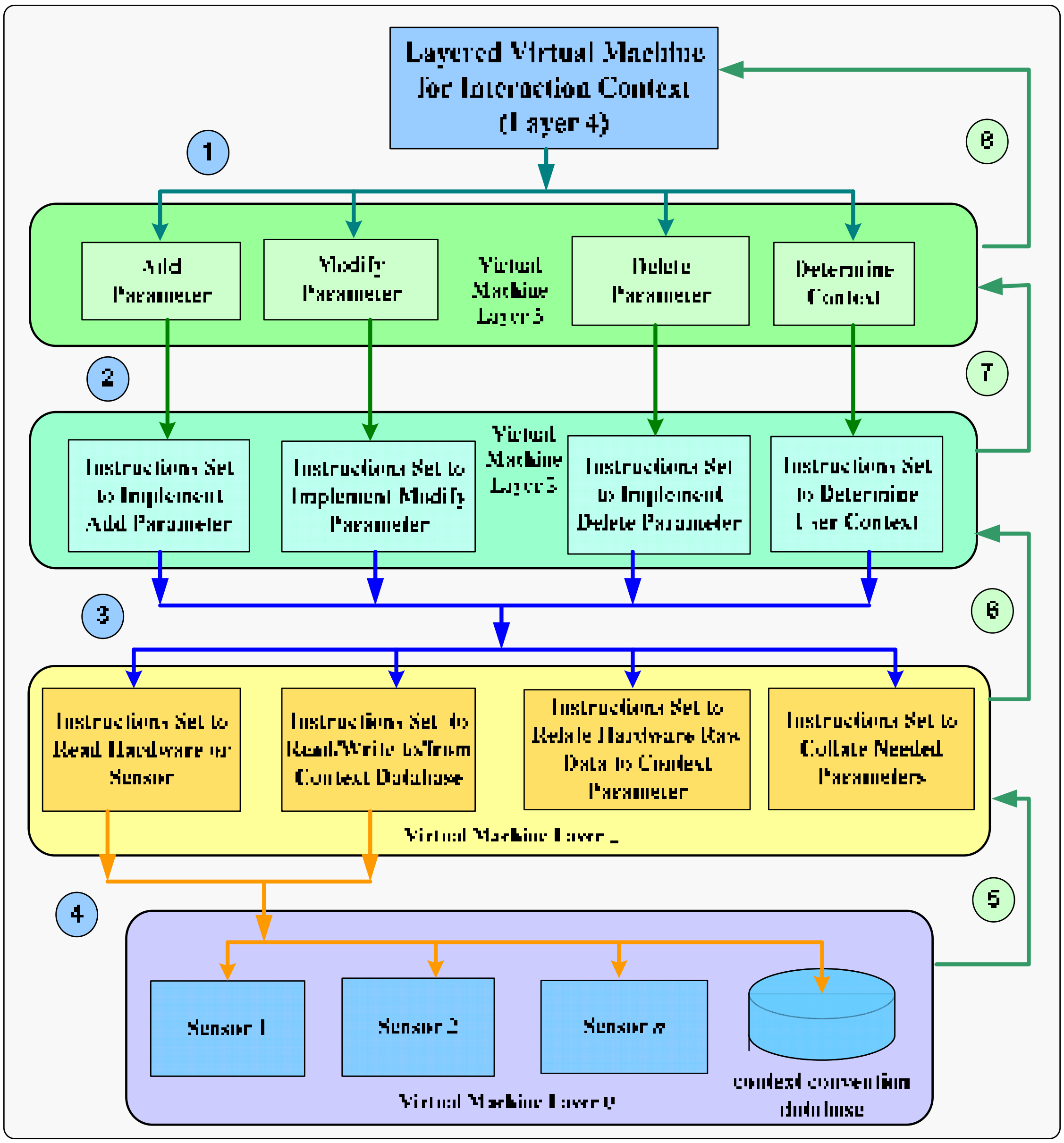


Figure 4.1 The design of a layered virtual machine for incremental user context.

The VM Layer 4 acts as the human-machine interface; its "*instruction set*" are the four functions found in Layer 3 – the "*add parameter*", "*modify parameter*", and "*delete parameter*" are basic commands that manipulate the sensor-based context parameters while "*determine context*" yields the values of currently-defined parameters. VM Layer 2 is a "*library of functions*" that collectively supports Layer 3 instructions while Layer 1 is another "*library of functions*" that acts as a link between Layer 2 and Layer 0.

4.4.3 Adding a Context Parameter

Consider using the VM to add a specimen context parameter: the noise level. See Figure 4.2. Upon invoking the VM user interface (i.e. Layer 4), the user chooses the "Add Parameter" menu. A window opens up, transferring the execution control to Layer 3. To realize adding a new context parameter, at least four functions must exist, namely: (1) *getting context type of the parameter*, (2) *getting parameter name*, (3) *getting number of parameter units*, and (4) *getting number of parameter values and conventions*. Via Layer 3, the user inputs "*Noise level*" as parameter name, itself an EC parameter, "*1*" as parameter unit, and "*3*" as parameter values and conventions. When done, two new windows open up, one window at a time, that brings up the functionalities of Layer 2. For each parameter's unit, the VM receives inputs for the unit name and the sensor (or hardware) that supplies its raw data. As shown, the unit of noise is specified as "*decibel*" and the BAPPU noise measuring device as the sensor supplying the data. When done, another Layer 2 window opens up for data entry of "*Parameter values and conventions*". In the diagram, the user specifies the value (range of decibels) that is equivalent to "*quiet*", "*moderate*" and "*noisy*". When done, a window for Layer 1 opens up to save the newly-added parameter information. This function interacts directly with the hardware (i.e. the context convention database).

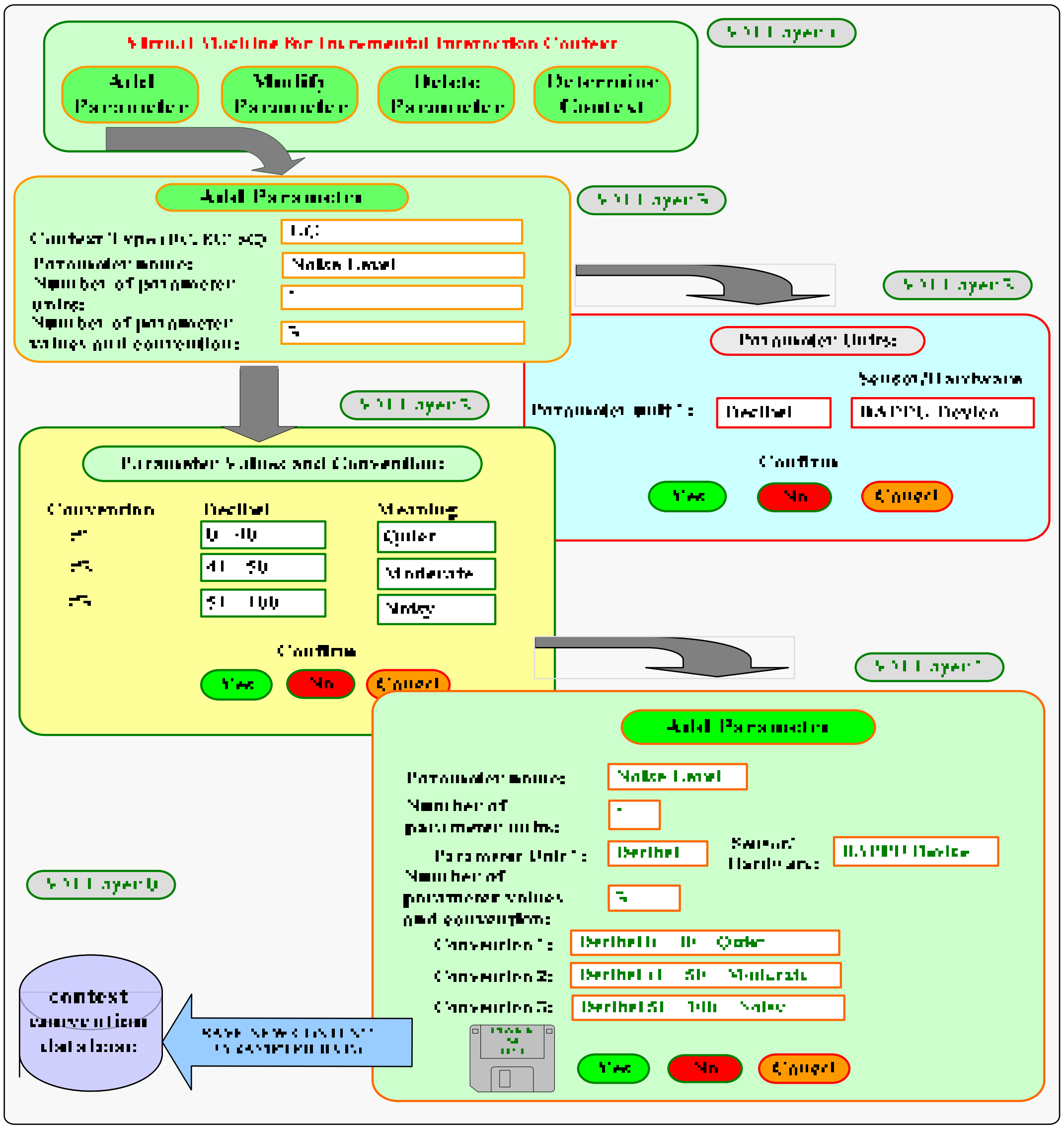


Figure 4.2 The interactions among layers to add new context parameter: "Noise Level".

4.4.4 Modifying and Deleting a Context Parameter

The VM layers interaction in "*Modify parameter*" is almost identical to that of "*Delete Parameter*" function. The only thing extra is that allowing the user to select the context parameter that should be modified. Other than that, everything else is the same. The processes involved in "*Delete Parameter*" menu are shown in Figure 4.3. Upon menu selection, the execution control goes to Layer 3, demanding the user to specify the parameter for deletion (i.e. "*Noise level*" is chosen for deletion). When confirmed, the information on the parameter for deletion is extracted and read from database (transfer of control from Layer 2 to Layer 1 to Layer 0). When information is obtained, the control goes back to Layer 2 where information is presented and a re-confirmation of its deletion is required. When parameter deletion is done, the control goes back to Layer 3 which presents the updated list of context parameters. An "*OK*" button click transfers the control back to Layer 4.

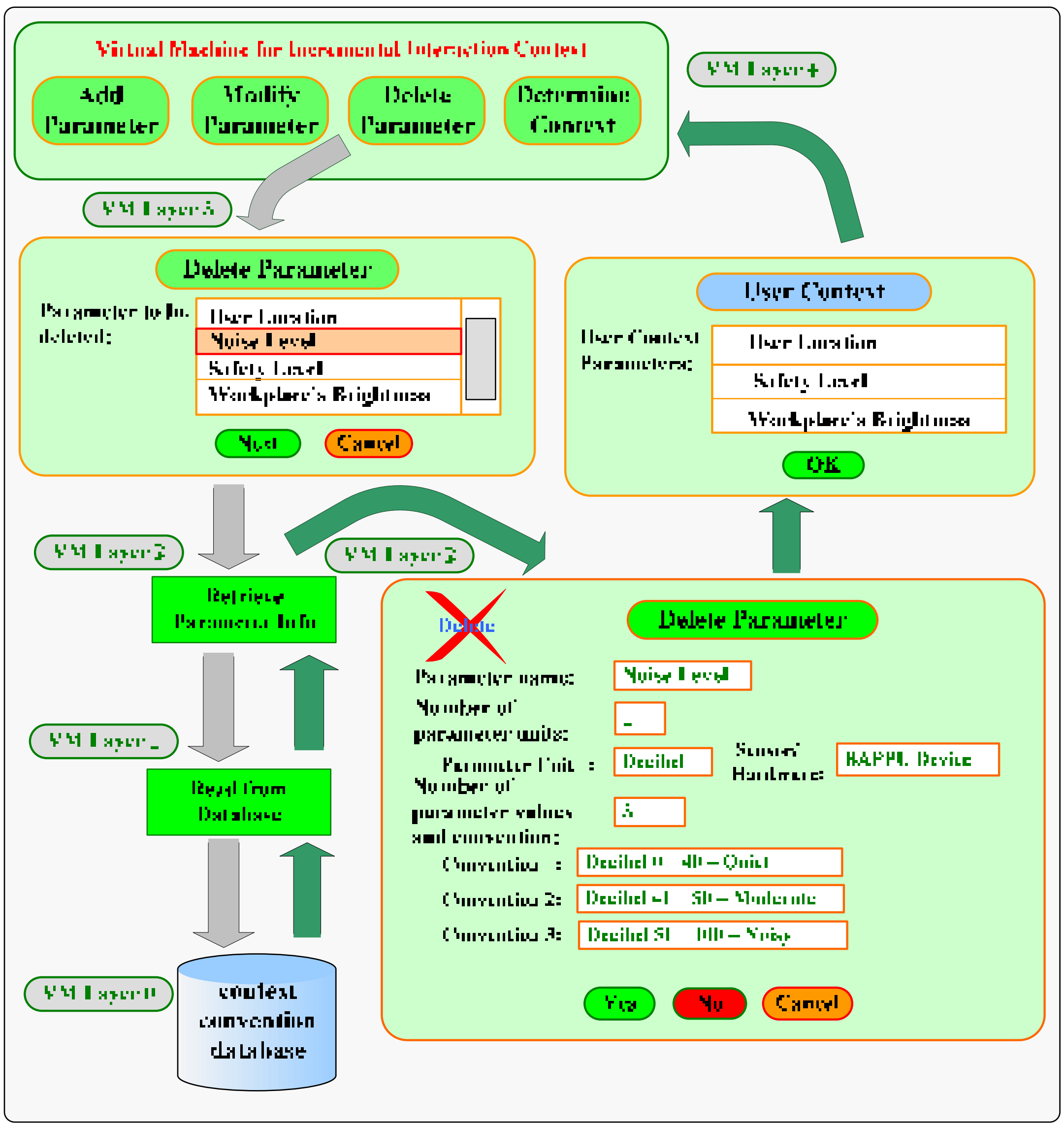


Figure 4.3 The VM layers interaction to realize "deleting a user context parameter".

4.4.5 Capturing the User's Current Context

The interactions of VM layers to "*Determine Context*" are shown in Figure 4.4. This is simulated using *specimen* context parameters, namely (1) *the user location*, (2) *the safety level*, and (3) *the workplace's brightness*. When the user opts for this menu, the VM execution control goes to Layer 3. The function "*get user context*" creates threads equal to the number of parameters. This process produces thread "*get parameter 1*", assigned to detect user location, thread "*get parameter 2*" assigned to get the user's safety level, and the thread "*get parameter 3*" for the user's workplace's brightness (i.e. light intensity). The concepts involved are identical for each thread. Consider the case of "*user location*". The thread passes control to Layer 1 wherein the function takes sample data from a sensor (i.e. global positioning system (GPS) (Herbordt, Horiuchi et al. 2005)) attached to the user computer's USB port. In the VM design, user can specify the number of raw data that need to be sampled and in what frequency (*n* samples per *m* unit of time). These samples are then collated, normalized and interpreted.

For example, a specimen GPS data of 5 samples, taken 1 sample per minute, is shown in Figure 4.5. The data are then normalized (averaged), hence, the user's computer is located at 14°11' latitude and -120°57' longitude. Then, this value is interpreted using the convention values for user location parameter. Tableau 4.1 shows the format of the convention values of the specimen parameters. (Recall that the convention value of a parameter is created during the "*Add Parameter*" process.) Using Tableau 4.1-a, the interpretation identifies if the user (who uses the computer equipped with a GPS) is at home, at work or on the go.

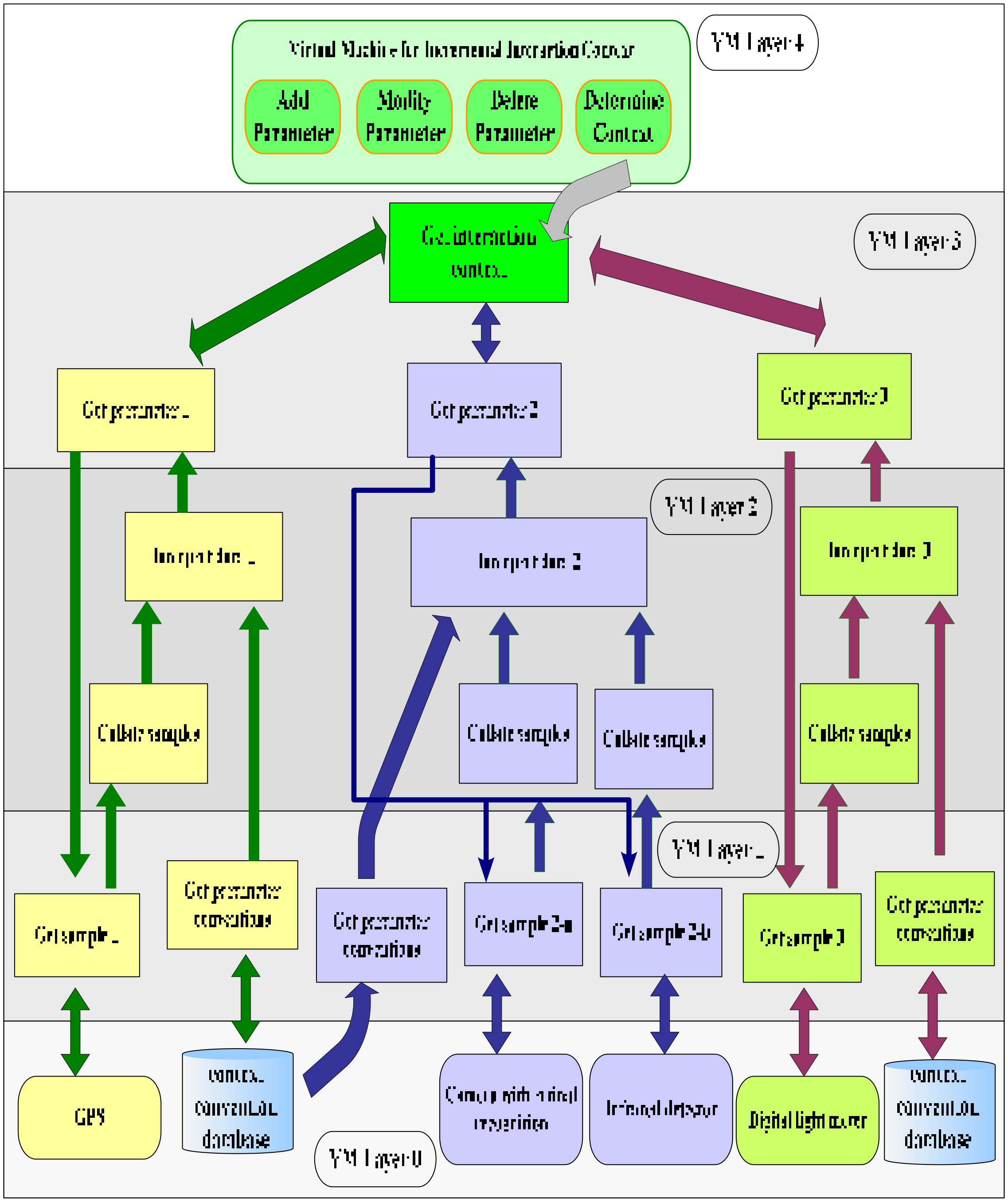


Figure 4.4 VM layers interaction in detecting the current interaction context.

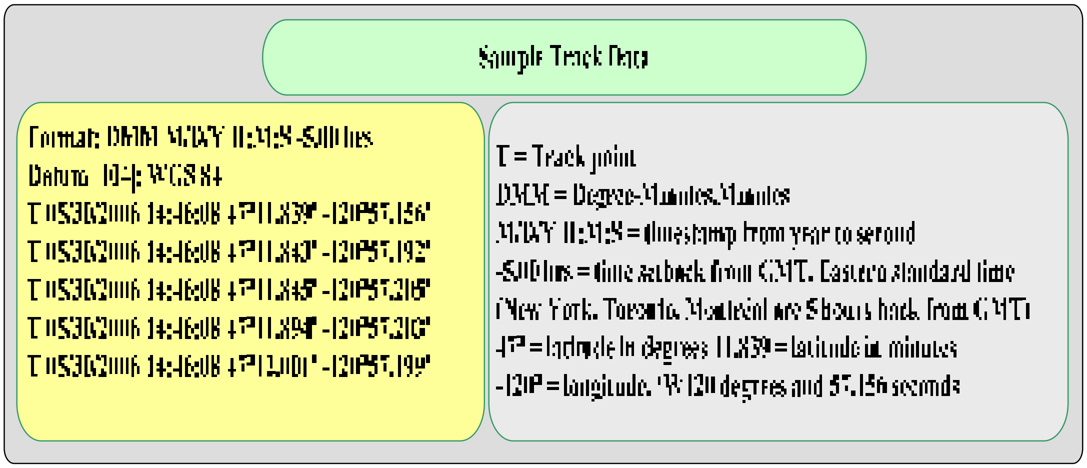


Figure 4.5 Sample GPS data gathered from Garmin GPSIII+.

Specimen parameter 2 (the workplace's safety level) is a function of (i) the person sitting in front of the computer, and (ii) the presence of other people in the user's workplace. A camera with retinal recognition (Bellik 1995) may be used to identify the person sitting in the user's seat. The identification process would yield three values: (1) *User* – if the legitimate user is detected, (2) *Other* – if another person is detected, and (3) *Empty* – if no one is detected. Also, an infrared detector (Archambault 1999) may be used to identify the presence of other person in front or in either side of the user. The identification process would yield two values: (1) *Image* – if at least one person is detected, and (2) *No Image* – if nobody is detected. (Note that the image and pattern recognition is not the subject of this work; hence, the detection process is not elucidated.). The VM takes $n = 5$ samples, normalizes them and compares the result against the convention values in Tableau 4.1-b. The interpretation yields a result indicating if user's workplace is *safe*, *sensitive* or *risky*. This specimen parameter is useful for people working on sensitive data (e.g. bank manager) but can be irritating to a person working with teammates (e.g. students working on a project). Hence, this specimen parameter can be added or deleted on the user's discretion.

Tableau 4.1 Sample conventions of the specimen sensor-based context parameters

(a) Convention format for user's location

Convention No.	Latitude	Longitude	Meaning
1	<value11>	<value12>	At home
2	<value21>	<value22>	At work
3	not <value11> AND not <value21>	NOT <value12> AND NOT <value22>	On the go

(b) Convention format for safety level in user's workplace

Convention No.	Detected in user's seat	Detected in user's workplace	Meaning
2	User	Image	Sensitive
1	User	No Image	Safe
2	Empty	Image	Sensitive
1	Empty	No Image	Safe
3	Other	Image	Risky
3	Other	No Image	Risky

(c) Convention format for light intensity in user's workplace

Convention No.	Foot-Candle	Meaning
1	<value-range1>	Bright
2	<value-range2>	Moderate
3	<value-range3>	Dark

The third specimen parameter (i.e. workplace's brightness) detects the workplace's light intensity. Here, we can assume that a sensor measuring the light's intensity is attached to the computer's USB port. Its measurement unit, the *foot-candle*, is the number of "*lumens*" falling on a square foot of an inch; lumen is a unit of light used to rate the output of a bulb. For example, we may assume the following conventions in a user's workplace: (a) *0 – 9 foot candles* = *dark*, (b) *10 – 20 foot-candles* = *moderate*, and (c) *21 – 100 foot-candles* = *bright*. The processes involved in sampling, collating and interpreting sensor data for parameter 3 is identical with the other 2 parameters mentioned above. Given the specimen parameters, when "*determine context*" is done, the output indicates (1) *if the user is at home, at work or on the* *go*, (2) *if user's workplace is safe, sensitive or risky*, and (3) *if the workplace's light intensity is bright, moderate or dark*.

4.4.6 Context Storage and Dissemination

In general, if a system must obtain an accurate representation of the user's interaction context, then the system must be introduced to the most number of possible context parameters. As a context parameter is added to the system, the VM's context convention database forms a tree-like IC structure, as shown in generic format in Figure 4.6. 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.

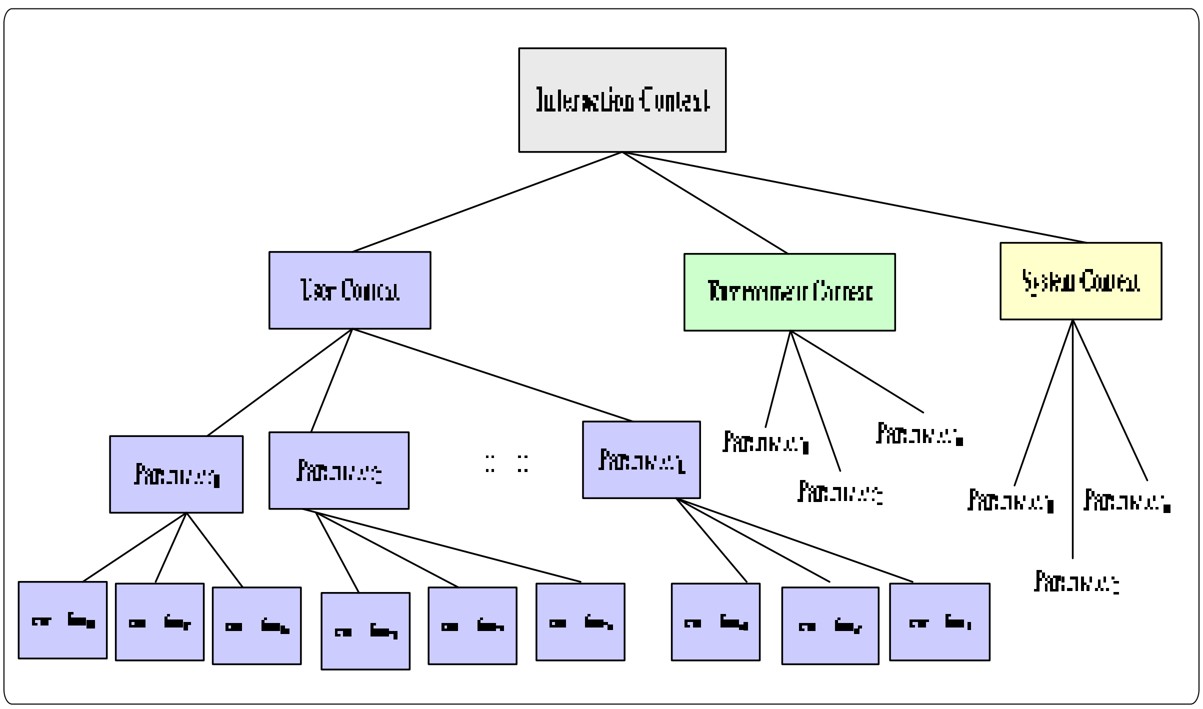


Figure 4.6 The structure of stored IC parameters.

For the **IC** information to be propagated in a pervasive system, the data representation used is XML Schema which is based on XML (Ross and Lightman 2005). Figure 4.7(Left) illustrates the general XML format of a context parameter (i.e. name, units, source of raw data, and conventions) and Figure 4.7(Right) shows the various snapshots of windows involved in adding a parameter in the VM as implemented using Java programming language.

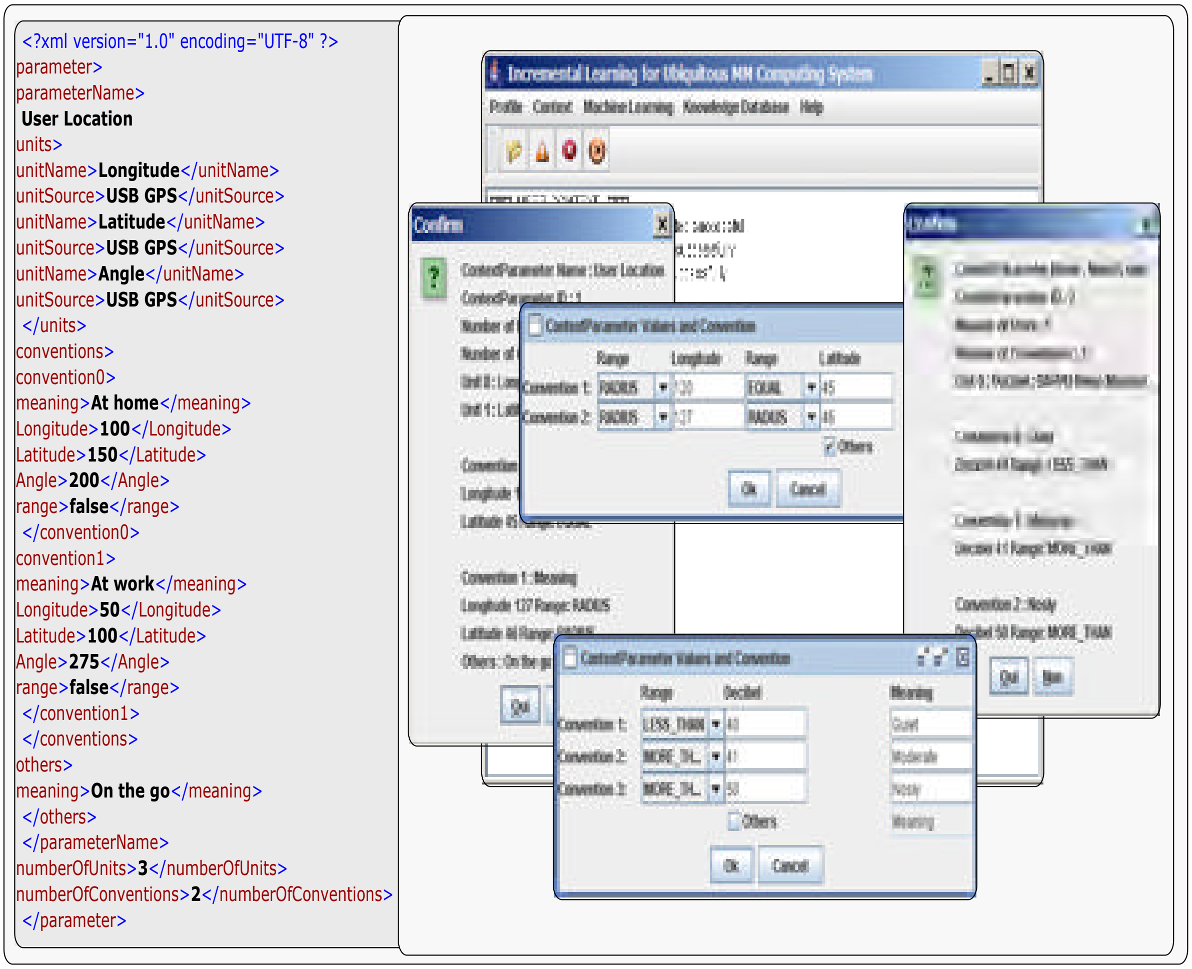


Figure 4.7 (Left) Sample context parameter in XML, (Right) snapshots of windows in add parameter menu.

4.5 Modalities, Media Devices and Context suitability

Here, we formulate the relationships between **IC** and modalities and between modalities and media group. This includes determining the suitability of a modality to a given **IC**.

4.5.1 Classification of Modalities

Here, *modality* refers to the logical interaction structure (i.e. the mode for data input and output between a user and computer). Using natural language processing as basis, we classify modalities into 6 different groups: (1) **Visual Input (VI_{in})**, (2) **Vocal Input (VO_{in})**, (3) **Manual Input (M_{in})**, (4) **Visual Output (VI_{out})**, (5) **Vocal Output (VO_{out})**, and (6) **Manual Output (M_{out}).** To realize multimodality, there should be at least one modality for data input and at least one modality for data output, as denoted by:

$$Modality = (VI_{in} \vee VO_{in} \vee M_{in}) \wedge (VI_{out} \vee VO_{out} \vee M_{out}) \quad (4.5)$$

4.5.2 Classification of Media Devices

In this work, *media* are physical devices that are used to implement a modality. Regardless of size, shape, colour and other attributes, all media – past, present or future – can be classified based on the human body part that uses the device to generate data input and the body part that uses the device to consume the output data. Hence, the classifications are as follows:

(1) **Visual Input Media (VIM)** – these devices obtain user input from human sight,

(2) **Visual Output Media (VOM)** – these devices generate output that is meant to be read,

(3) **Oral Input Media (OIM) –** devices that use user's voice to generate input data,

(4) **Hearing Output Media (HOM)** – devices that output meant to be heard,

(5) **Touch Input Media (TIM)** – these devices generate input via human touch,

(6) **Manual Input Media (MIM)** – these devices generate input using hand strokes, and

(7) **Touch Output Media (TIM)** – the user touches these devices to obtain data output.

4.5.3 Relationship between Modalities and Media Devices

When a modality is found suitable to a given IC, then media that support such modality are chosen. 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 relationship is shown in Figure 4.8. Often, there are many available devices that belong to the same media group. If such is the case then instead of activating them all, devices activation is determined through their priority rankings. To support this scheme, let there be a function $\mathbf{g_2}$ that maps a media group to a media device and its priority rank, and is denoted **$\mathbf{g_2}$: Media Group → (Media Device, Priority)**. Hence sample elements of these functions are:

$\mathbf{g_1}$ = {(VIin, VIM), (VIout, VOM), (VOin, OIM), (VOout, HOM), (Min, TIM), (Min, MIM), (Mout, 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.}.

It must be noted, however, that although media technically refers to a hardware element, we opted to include a few software elements without which VO_{in} and VO_{out} modalities could not possibly be implemented. These are the speech recognition software and speech synthesis software.

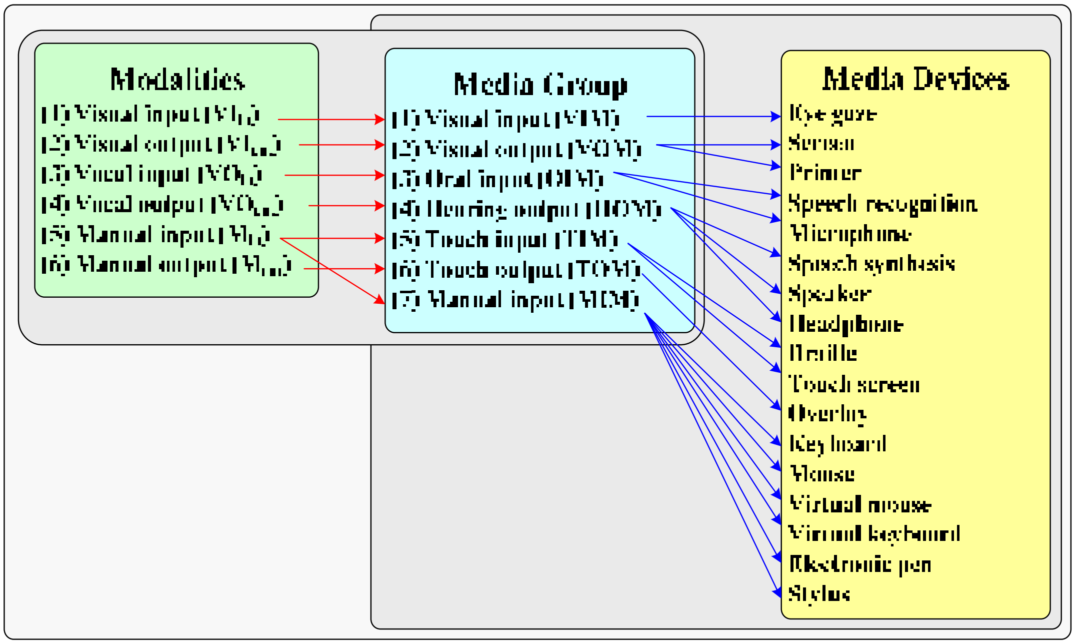


Figure 4.8 The relationship between modalities and media, and media group and media devices.

4.5.4 Measuring the Context Suitability of a Modality

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 in consideration has *n* suitability scores, one for each parameter. The following conventions are adopted:

1. A modality's level of suitability to any context parameter is one of the following: H (high), M (medium), L (low), and I (inappropriate).

2. Mathematically, $H = 100\%$, $M = 75\%$, $L = 50\%$, *and* $I = 0\%$,

3. Given context parameter $i \in$ user context, then a modality's suitability score to the overall context, and its final suitability score are given by:

$$SuitabilityScore_{modality} = \sqrt[n]{\prod_{i=1}^{n} context\_parameter_i} \quad (4.6)$$

$$FinalSuitability_{modality} = \begin{cases} H \text{ if } SuitabilityScore_{modality} = 1.00 \\ M \text{ if } 0.75 \le SuitabilityScore_{modality} < 1.00 \\ L \text{ if } 0.50 \le SuitabilityScore_{modality} < 0.75 \\ I \text{ if } SuitabilityScore_{modality} < 0.50 \end{cases} \quad (4.7)$$

4.5.5 Optimal Modalities and Media Devices' Priority Rankings

Figure 4.9 shows the algorithm for determining the suitability of modalities to a given IC. Also Figure 4.12-Algorithm 4 checks if multimodality is possible by checking that not all of input modalities are scored "inappropriate". The same is true for output modalities. 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. Through function $\mathbf{g_1}$, the media group that supports 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** | **p** $\rightarrow$ **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**, priorities $\mathbf{p_1}$ and $\mathbf{p_2}$ where Priority: £$_1$ (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, $\exists\mathbf{p_1}$**:** Priority, $\forall\mathbf{p_2}$**:** Priority | $d \bullet m \rightarrow (d, p_1) \in g_2 \wedge (p_1 < p_2)$.

```
// Initialization
Assignment index i ← 1 to 6 to represent modalities (VI_in, VO_in, M_in, VI_out, VO_out, and M_out, 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
```

Figure 4.9 Algorithm to determine modality's suitability to IC.

Let there be a *media devices priority table* (MDPT) (see Tableau 4.2) 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 $T_n \in$ **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.

Tableau 4.2 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			

4.5.6 Rules for Priority Ranking of Media Devices

Given that an optimal modality is already selected, then the top-ranked media device/s in the media group that supports the selected modality is/are activated. The rules governing device activation are as follows:

1. If the optimal modality's final suitability = 'H' then the activation of its supporting media group is essential. If no media devices belonging to such media group are found, the implementation of the optimal modality is not possible. The system searches for a replacement to the optimal modality.

2. A replacement modality (see algorithm in Figure 4.9) with 'M' or 'L' suitability score means that the activation of its supporting media group is the last recourse to implement multimodality. The absence of media devices for such media group means that multimodality failed (due to absence of supporting media devices).

For two or more media devices that belong to the same media group, the rules of their priority rankings are as follows:

1. If their functionalities are identical (e.g. a mouse and a virtual mouse), activating both is incorrect because it is plain redundancy. Instead, one should be ranked higher in priority than the other. The most-commonly-used device gets the higher priority.

2. If their functionalities are complementary (e.g. a mouse and a keyboard), activating both is acceptable. However, if one is more commonly used than the other (i.e. they do not always come in pair), then the more-commonly-used one gets the higher priority. If both devices always come together as a pair, then both are ranked equal in priority.

In the early stage of knowledge acquisition, it is the end user that provides this ranking, which depends on the concerned context. For example, in a quiet workplace, a speaker can be the top-ranked hearing output device. In a noisy environment, however, the headphone gets the top priority. This priority is reflected in every MDPT associated with every scenario. Initially, there is one MDPT, similar to Tableau 4.2. A second MDPT can be created from the first one by re-organizing the priority order of different devices and by inserting devices into it, as deemed necessary in the scenario. So does follow for a 3rd, a 4th, and an n^{th} MDPT. A MDPT is not static; it can be modified by the user when needed. The MDPT in Tableau 4.2 is a specimen table and does not contain an exhaustive list of devices. It is merely used to demonstration purposes.

4.6 Context Learning and Adaptation

After establishing the relationships among IC, modalities and media devices, we put these relationships to use by considering a specimen IC to which the pervasive MM computing system will adapt and learn.

4.6.1 Specimen Interaction Context

Our specimen user context is based on the following parameters: (1) *user location* – identifies if the user is *at home*, *at work*, or *on the go*, (2) *noise level* – identifies if the user's workplace is *quiet*, *moderate* or *noisy*, (3) *the safety/risk factor* – determines the one sitting in user's workplace and detects the presence of other people; the result identifies if the workplace is *safe*, *sensitive* or *risky*, (4) *the user's handicap* – determines if user is a *regular user* or is a *handicapped*, (5) *the computing device* – identifies if user is using a *PC*, a *laptop* or a *PDA or cell phone*. As to be expected, for each parameter's distinct value, the degree of modality's suitability varies accordingly.

4.6.2 The Context of User Location, Noise Level, and Workplace's Safety

As Tableau 4.3 shows, sample conventions, in generic format, are made for user's locations. The GPS' readings of latitude and longitude provide a specific meaning (i.e. convention). Also, the degrees of suitability of various modalities for each value of user location are also listed.

In Tableau 4.4, meanings are assigned to a specific range of decibels as observed from the user's workplace. Some sensors, such as those found materials in www.bappu.com, can be attached to the computer's USB port to capture the environment's noise level. The table also shows how suitable a certain modality is based on the level of noise in the workplace.

Tableau 4.3 User location as context parameter: convention and its modalities' suitability scores

(a): User location convention table using GPS values

Convention No.	Latitude	Longitude	Meaning
1	<value11>	<value12>	At home
2	<value21>	<value22>	At work
3	not <value11> AND not <value21>	NOT <value12> AND NOT <value22>	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

Tableau 4.4 Noise level as context parameter: sample convention and modalities' suitability scores

(a): Sample noise level convention

Convention No.	Decibel	Meaning
1	Less than 41	Quiet
2	41 to 50	Moderate
3	Greater than 50	Noisy

(b): Modality selection based on noise level

Type of Modality	Noise level = Quiet	Noise level = Moderate	Noise level = Noisy
Visual Input	H	H	H
Visual Output	H	H	H
Vocal Input	H	M	L
Vocal Output	H	H	M
Manual Input	H	H	H
Manual Output	H	H	H

The context of *safety level* is already briefly discussed in section 4.4.5 – *"Capturing the user's current context*". It is based on two factors: (1) *the person sitting in the user's seat as detected by a camera with retinal recognition*, and (2) *the presence of other people present in the user's workplace as detected by an infrared detector*. The (Bellik 1995) is one method of determining a legitimate user from an intruder. Likewise, (Archambault 1999) provides a wide range of infrared detector products. Figure 4.10 shows the safety level detection process.*The combination of the results obtained from infrared detector and of camera indicates how sensitive the user's workplace is.*

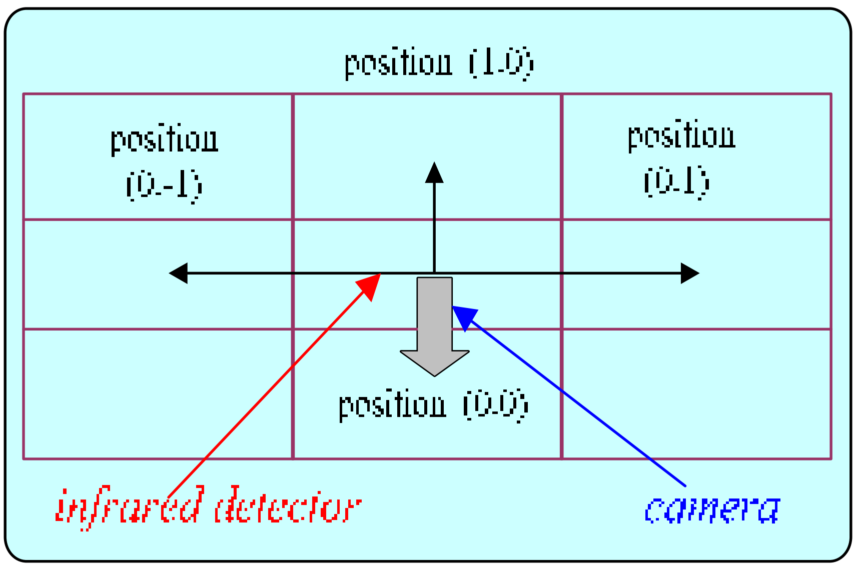


Figure 4.10 The safety/risk factor detection using an infrared detector and a camera.

Tableau 4.5(a) provides the workplace's risk/safety convention. Tableau 4.5(b) shows our perception of modalities' suitability with respect to safety level. Note that all modalities are rated *inappropriate* if safety level becomes bad (i.e. risky), not because they are really inappropriate to the context but as a mean to protect the user from unauthorized people's intrusion. As per our view, in a risky setting, the system automatically saves user's information and then logs him out from the system.

Tableau 4.5 Safety level as context parameter: sample convention and modalities' suitability scores

(a): The safety/risk factor convention table

Convention No.	Detected human's seat	Detected human's workplace	Meaning
2	User	Image	Sensitive
1	User	No Image	Safe
3	Empty	Image	Sensitive
1	Empty	No Image	Safe
3	Other	Image	Risky
3	Other	No Image	Risky

(b): Modality selection based on workplace's safety level

Type of Modality	Safety level = Safe	Safety level = Sensitive	Safety level = Risky
Visual Input	H	M	L
Visual Output	H	M	L
Vocal Input	H	M	L
Vocal Output	H	M	L
Manual Input	H	M	L
Manual Output	H	M	L

4.6.3 The Context of User Handicap and Computing Device

Figure 4.11 shows the generic format of our user profile. For this work, some information (i.e. user QoS and supplier preferences) are not discussed since they are not related to this paper's content. A *user profile* contains, among others, the user's username, password and a list of the user's computing devices and their corresponding schedules. Since the user is mobile, his computing device is identified via this part of user profile. In the *special needs* section, the user is identified as either a *regular user* or a handicapped. If the user is indeed *handicapped*, the disability is specified, indicating if the user is *a mute*, *a deaf*, *a visually impaired*, or *a manually handicapped*. Here, the importance of multimodality is obvious; it provides handicapped users the chance to access informatics through modalities that suit their conditions.

Tableau 4.6 (a) shows the user profile/handicap convention while Tableau 4.6(b) shows the modalities suitability based on such profile. We also consider the user's computing device as a context parameter because the degree of modality's suitability using a PC, a laptop or a

PDA varies. The PDA, for example, has very limited resources (memory, CPU, battery) as compared with a PC or a laptop.

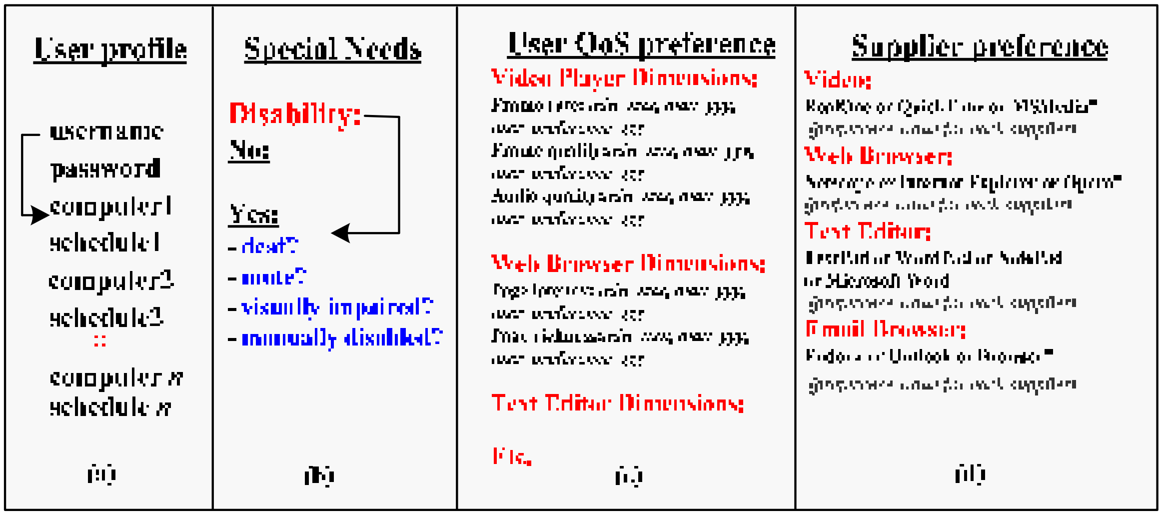


Figure 4.11 A sample user profile.

Tableau 4.6 User handicap as parameter: sample convention and modalities' suitability scores

(?): Profile/disability convention

Convention No.	User Profile
1	Regular User
2	Deaf
3	Mute
4	Visually Impaired
5	Manually Impaired

(?): Modality selection based on user profile/handicap

Type of Modality	User = Regular User	User = Deaf	User = Mute	User = Visually Impaired	User = Manually Impaired
Visual Input	H	H	H	L	H
Visual Output	H	H	H	L	H
Vocal Input	H	M	L	H	H
Vocal Output	H	L	M	H	H
Manual Input	H	H	H	H	L
Manual Output	H	H	H	H	L

Tableau 4.7(a) shows our computing device conventions. Tableau 4.7(b) shows the modalities' suitability based on these computing devices.

Tableau 4.7 Computing device as parameter: sample convention and modalities' suitability scores

(a): Computing device convention

Convention No.	Computing Device
1	Desktop PC
2	Laptop
3	PDA
3	Cellular phone

(b): Modality selection based on user's computing device

Type of Modality	Computing device = Desktop PC	Computing device = Laptop	Computing device = PDA/Cellphone
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	L

4.6.4 Scenarios and Case-Based Reasoning with Supervised Learning

A *scenario* is an event that needs a system response. The stimulus that triggers the event is the *pre-condition scenario*, while the system response to such event is called the *post-condition scenario*. In this work, the pre-condition scenario is a specific interaction context $IC_i \in \mathbf{IC}$. The desired post-condition is the suitable modalities and their supporting media devices.

Given that $IC_i = UC_k \otimes EC_l \otimes SC_m$ then the total number scenarios, denoted as *scenTot*, is the product of the number of convention values of each context parameter, that is,

$$scenTot = \prod_{i=1}^{param\_max} card(IC_i) \quad (4.8)$$

where card(IC_i) = cardinality/total number of convention values for interaction context parameter *i*. scenTot may also be specified as scenTot = card(UC_k) x card(EC_l) x card(SC_m). The scenario number, *scenNum*, assigned to any specific instance of an interaction context is given by:

$$ScenNum = IC_{param\_max} + \sum_{i=1}^{param\_max-1} ((IC_i - 1) \bullet \prod_{j=i+1}^{param\_max-1} card(IC_j)) \quad (4.9)$$

An entry in a scenario table can be done in two ways: through expert (i.e. user) intervention or on the fly as the scenario is encountered. A *scenario table* is simply a tabulation of distinct scenarios, each of which is composed of pre- and post-condition scenarios. An entry in the scenario table is done as follows:

1. the current context parameters and their conventions are listed in the pre-condition scenario, see Figure 4.12-Algorithm 2,
2. the post-condition scenario lists down the corresponding suitability scores of each modality, calculated using Equations 6 and 7, see Figure 4.12-Algorithm 3,
3. the scenario number is calculated using Equation 9, and
4. the pointer to MDPT is initially pointed to the very first MDPT, unless it has already been rectified by the expert. A sample snapshot of such table is shown in Tableau 4.8.

Once a scenario is stored in the scenario table, it becomes an exemplar. An exemplar is a stored knowledge. When the ML component receives a new scenario (i.e. new context), it converts it into a *case*, specifying the problem. The ML component searches for a match between a case and exemplars. When a new scenario is converted into a case, the resulting case is now composed of three elements, namely:

Tableau 4.8 Scenario table contains records of pre-condition and post-condition scenarios

Scenario Number	Pre-Condition					Post-Condition						Media Devices Priority Table
	User Location	Noise Level	Safety Level	User Profile	Computing Device	Visual Input	Vocal Input	Manual Input	Visual Output	Vocal Output	Manual Output	
1	1	1	1	1	1	H	H	H	H	H	H	T_1
2	1	1	1	1	2	H	H	H	H	H	H	T_3
3	1	1	1	1	3	M	H	H	H	H	M	T_2
4	1	1	1	2	1	H	M	H	H	L	H	T_4
::	::	::	::	::	::	::	::	::	::	::	::	::
404	3	3	3	5	2	L	L	L	L	L	L	T_n
405	3	3	3	5	3	L	L	L	L	L	L	T_n

1. **the problem** – the pre-condition scenario in consideration,
2. **the solution** – the final suitability of each modality, and
3. **the evaluation** – the rate of relevance of the solution. Using the similarity algorithm, it compares the problem in the new case against all the available problems in the 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.

Figure 4.12 Algorithms related to knowledge acquisition, entry in scenario table and selection of optimal modality.

Inspired by (Lajmi, Ghedira et al. 2007), 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:

$$Sim(NC, MC) = \frac{1}{3} Sim(UC_{NC}, UC_{MC}) + \frac{1}{3} Sim(EC_{NC}, EC_{MC}) + \frac{1}{3} Sim(SC_{NC}, SC_{MC}) \quad (4.10)$$

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

$$Sim(UC_{NC}, UC_{MC}) = \frac{\sum_{i=1}^{max_{NC}} Sim(UC\_Par_{i_{NC}}, UC_{MC})}{max(UC\_Par_{NC}, UC\_Par_{MC})} \quad (4.11)$$

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 $Sim(UC\_Par_{i_{NC}}, UC_{MC}) = max_{j=1 \, to \, max_{MC}} Sim(UC\_Par_{i_{NC}}, UC\_Par_{j_{MC}})$ where $UC\_Par_{j_{MC}} \in UC_{MC}$ and $Sim(UC\_Par_{i_{NC}}, UC\_Par_{j_{MC}}) \in [0, 1]$ 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 11 must be applied, with the formula adjusted accordingly to denote EC and SC, respectively, yielding:

$$Sim(EC_{NC}, EC_{MC}) = \frac{\sum_{i=1}^{max_{NC}} Sim(EC\_Par_{i_{NC}}, EC_{MC})}{max(EC\_Par_{NC}, EC\_Par_{MC})} \quad (4.12)$$

$$Sim(SC_{NC}, SC_{MC}) = \frac{\sum_{i=1}^{max_{NC}} Sim(SC\_Par_{i_{NC}}, SC_{MC})}{max(SC\_Par_{NC}, SC\_Par_{MC})} \quad (4.13)$$

Equation 4.10 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 himself, however, decides the acceptable threshold score.

If no match, however, is found (i.e. relevance score is lower than accepted threshold) then the ML component takes the closest scenario as the initial solution of the new case. The user may not accept it. In such a case, a new case with *supervised learning* is produced. If the new case's problem contains more context parameters than those of recorded cases, the expert may decide to include the missing parameter(s) into the a priori knowledge (see Figure 4.6). Thereafter, the new case's post-condition scenario is re-evaluated (see Figure 4.12-Algorithm 3). The new case in then added to the scenario table, and its scenario number calculated. This whole learning mechanism is called *case-based reasoning with supervised learning*.

As an example, consider the following IC: user location = at home, noise level = quiet, safety factor = safe, user profile = regular user and computing device = PDA. This **IC** condition (ic_1, ic_2, ic_3, ic_4, ic_5) = (1, 1, 1, 1, 3). It is scenario number 3. The calculated final suitability scores of the modality types are given below and are also stored in scenario table (Tableau 4.8).

Visual Input= $[(H)(H)(H)(H)(L)]^{1/5} = [(1)(1)(1)(1)(0.50)]^{1/5} = 0.87$=Medium suitability
Vocal Input = $[(H)(H)(H)(H)(H)]^{1/5} = [(1)(1)(1)(1)(1)]^{1/5} = 1$ = High suitability
Manual Input = $[(H)(H)(H)(H)(H)]^{1/5} = [(1)(1)(1)(1)(1)]^{1/5} = 1$ = High suitability
Visual Output = $[(H)(H)(H)(H)(H)]^{1/5} = [(1)(1)(1)(1)(1)]^{1/5} = 1$ = High suitability
Vocal Output = $[(H)(H)(H)(H)(H)]^{1/5} = [(1)(1)(1)(1)(1)]^{1/5} = 1$ = High suitability
Manual Output=$[(H)(H)(H)(H)(L)]^{1/5}=[(1)(1)(1)(1)(0.50)]^{1/5}=0.87$=Medium suitability

Given this case, modality is possible. The *optimal input modality* is both Vocal Input and Manual Input. The *optimal output modality* is Visual Output and Vocal Output. All non-optimal modalities are considered *optional*. If this same case reappears again in the future, then using the similarity algorithm (Equation 4.10), there is an exact match (scenario 3) that can be found in the database, hence, recalculation/decision making is evaded.

Also, Let $\mathbf{M_1}$, $\mathbf{M_2}$ … $\mathbf{M_6}$ be the set of modalities VI_{in}, VO_{in}, M_{in}, VI_{out}, VO_{out}, and M_{out} respective suitability scores. At any time, the suitability score of $\mathbf{M_1}$ is m_1 = {H, M, L, I} = {1, 0.75, 0.50, 0}. Such suitability scores also apply to $\mathbf{M_2}$, $\mathbf{M_3}$ … $\mathbf{M_6}$. Hence, the modalities selections, **Y**, as a vectored output is equal to the Cartesian product of the individual modality's suitability score, that is, $\mathbf{Y} = \mathbf{M_1} \times \mathbf{M_2} \times \mathbf{M_3} \times \mathbf{M_4} \times \mathbf{M_5} \times \mathbf{M_6} = (m_1, m_2, m_3, m_4, m_5, m_6)$ where $m_1 \in \mathbf{M_1}$, $m_2 \in \mathbf{M_2}$, … and $m_6 \in \mathbf{M_6}$. In the specimen IC, there are 3*3*3*5*3 = 405 possible context scenario combinations in **X** and 4^6 = 4096 possible modality's suitability combinations in **Y**. Hence, function $\mathbf{f_1}$**: X → Y** that maps user context to appropriate modalities is also expressed $\mathbf{f_1}: (c_1, c_2, c_3, c_4, c_5) \rightarrow (m_1, m_2, m_3, m_4, m_5, m_6)$.

4.6.5 Assigning a Scenario's MDPT

This process is shown in Figure 4.13 using the specimen IC. At the start (step 1), the Context Manager Agent (CMA) gets the current IC. In (step 2), this scenario becomes a case. Using the pre-condition scenario, the case's *scenNum* is calculated and is used as an index to the scenario table. Assuming that a perfect match is found then the post-condition scenario (i.e. the case's solution) is adopted with relevance score = 100%). Since the present case is not new to the system, then steps 3, 4, and 5 are skipped.

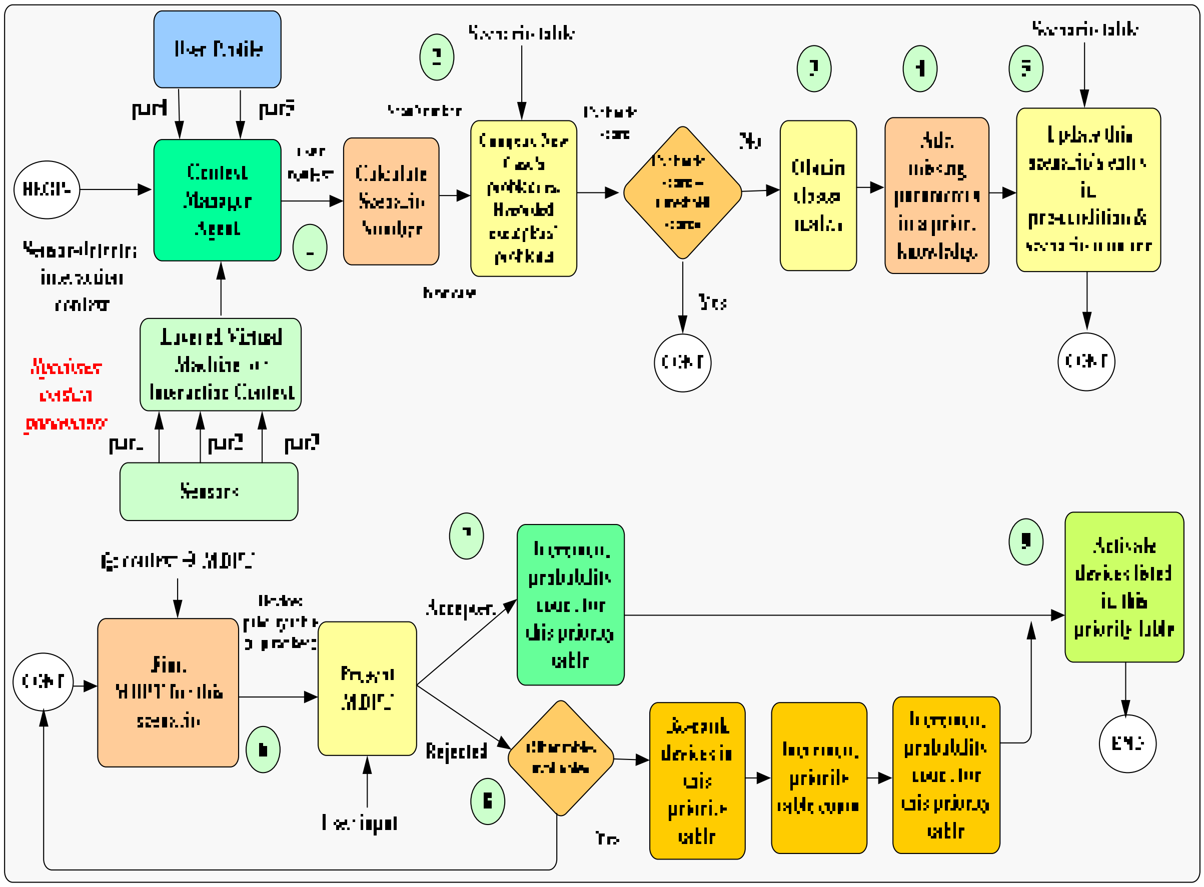


Figure 4.13 ML training for choosing the appropriate devices priority table for a specific context.

If the similarity/relevance score, however, is low (say, 40%), then no match is found. Hence, the closest match is retrieved and presented to the user. Because the proposed solution is wrong (i.e. 40% accurate vs. 60% erroneous), the new case is treated for adaptation maintenance. The large amount of error is brought by the fact that most of the context parameters in the new case cannot be found in the stored cases of scenario table. Hence, an update of a priori knowledge and scenario table is made; the new context parameters are added and the new case is stored in the scenario table. The new case's corresponding post-condition scenario is recalculated. Due to the newly added context parameter(s) in the scenario table, all scenario numbers of previous entries' are recalculated. In the scenario table, a MDPT for the new case has to be established; hence the available MDPT's are presented to the user, one table at time (step 6). If the user accepts the proposed table (step

7), the table's numerical identification is appended onto the scenario table. The media groups corresponding to the selected modalities are noted and their top-ranked media devices are selected for activation (step 9). If the user rejects such MDPT, then each of the other remaining tables will be presented (step 8). Recall that there is just one MDPT in the beginning. Hence, the user needs to modify the contents of the first table to create a second one. When this is done, the identification number of the newly-created MDPT is appended into the scenario table. And step 9 is executed.

Figure 4.14 illustrates the format of a completely filled scenario table for specimen user context. Note, however, that as long as new context parameters are being added, the scenario table will keep on growing. This makes our system adaptive to an ever-changing user context.

4.6.6 Finding Replacement to a Missing or Failed Device

At any time, it is possible that a selected top-ranked media device may be missing or defective. Some techniques for detecting device failures are available in (Hina, Tadj et al. 2006). Hence, a replacement should be found for the system to remain running and operational. The replacement can be found within the same MDPT assigned to the scenario. The algorithm of replacement to a failed device is shown in Figure 4.15. In (step 1), using scenario number (*scenNum*), the system determines its assigned MDPT which identifies the media groups' top-ranked devices. In (step 2), the environment profile is consulted to find out the currently available media devices. In (step 3), the system merely activates the top-ranked media device, if available. Otherwise, in (step 4) the second-ranked device is activated, also if available. If it is also missing or defective, then the third-ranked device is searched. In general, the search goes on until a selected device is found. The worse-case event is when no device in a media group in the MDPT is activated due to cascaded failure or collective absence of needed devices (step 5). In such case, the system abandons the selected optimal modality (because it cannot be implemented) and attempts to replace the optimal modality by available non-optimal modality. This process finds again the available media devices, by priority, to support the non-optimal modality. In the worst case that the non-optimal modalities cannot be supported, this simply means that multimodality is impossible in the given computing environment.

Figure 4.14 A sample snapshot of a completed scenario table, each entry with its assigned MDPT.

Given the failed device **d** of priority $\mathbf{p_1}$, the specification for finding the replacement media device d_{rep} is $\exists$m: **Media Group,** $\forall d_{rep}$: **Media Device,** $\exists p_1$: Priority, $\forall p_2$: Priority $\mid$ $(p_1 = p_1 + 1) \wedge (p_1 < p_2) \wedge m \rightarrow (d_{rep}, p_1) \in g_2 \bullet d_{rep}$.

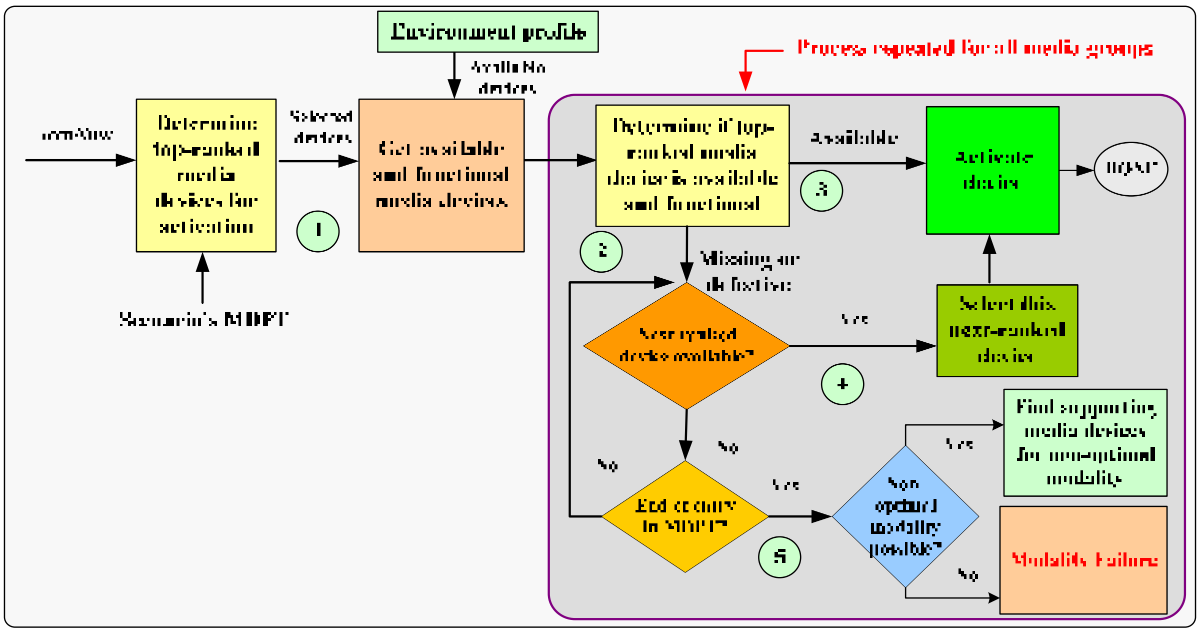


Figure 4.15 The ML process of finding replacement to a failed or missing device.

4.6.7 Media Devices' Priority Re-ranking due to a Newly-Installed Device

A newly-installed device affects the priority rankings of media devices in the media group where the new device belongs. Figure 4.16 illustrates the update process in a MDPT due to the arrival of this newly-installed device. In (step 1), given that the system has already recognized the new device via environment profile, the user provides the media group where it belongs. In (step 2), the MDPT assigned to scenario number 1 is retrieved and becomes the first MDPT to be updated. This priority table is edited (step 3). The new device's name is inserted into the table (step 4). In (step 5), the rankings of other devices in the same media group are updated by the user. When done, the second MDPT is searched. The update process is repeated on other scenarios until the last of MDPT is also updated. The update process is quite long (i.e. equal to the number of all MDPT's).

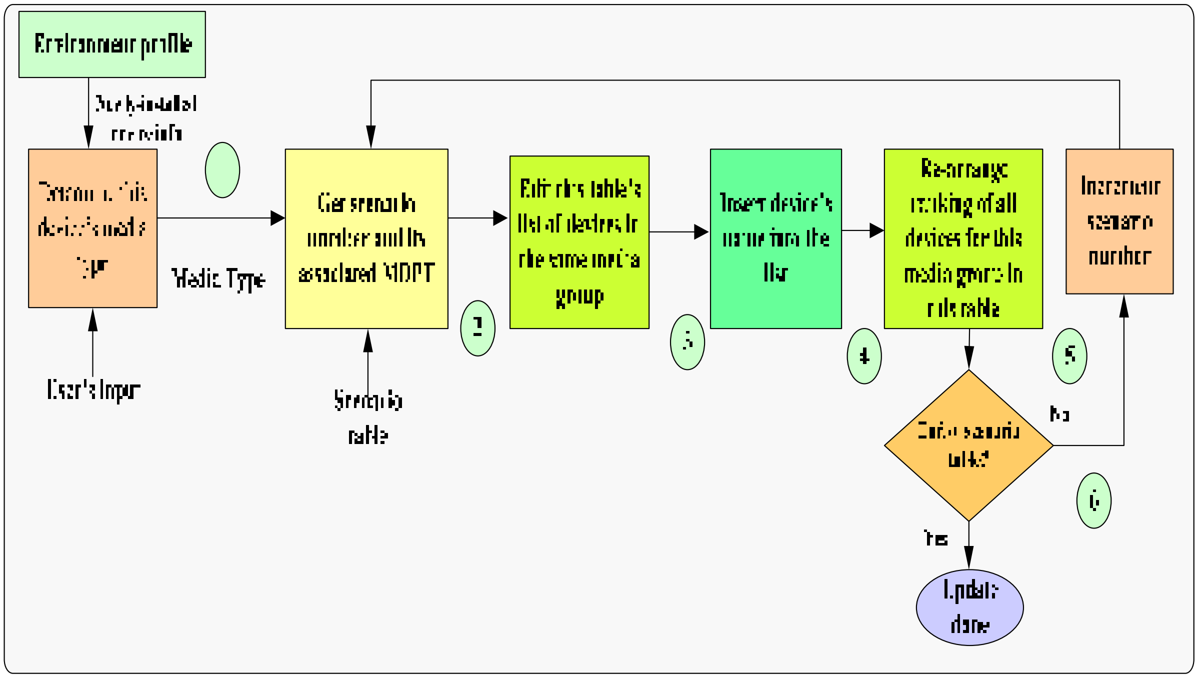


Figure 4.16 The ML process for update of devices priority tables due to a newly-installed device.

4.6.8 Our Multimodal Multimedia Computing System

Our proposed system is conceived for two purposes: (1) to contribute to MM computing and (2) to further advance autonomic computing system. To achieve the first goal, we develop the model that relates modality with user context, and associate media devices to support the implementation of the chosen modality. In the second goal, we advocate the propagation of knowledge, acquired through training, into the user's computing environment so that such knowledge can be used for system adaptation to user needs, and system restrictions. The major components of our MM computing system are shown in Figure 4.17.

The functionality of each component is given below:

1. **The Task Manager Agent (TMA)** – manages user's profile, task and pertinent data and their deployment from a server to the user's computing device, and vice versa.

2. **The Context Manager Agent (CMA)** – detects user context from sensors and user profile, and selects the modality and media apt for the context.

3. **The History and Knowledge-based Agent (HKA)** – responsible for ML training and knowledge acquisition.

4. **The Layered Virtual Machine for Interaction Context (LVMIC)** – detects sensor-based context and allows the incremental definition of context parameters.

5. **The Environmental Manager Agent (EMA)** – detects available and functional media devices in the user's environment.

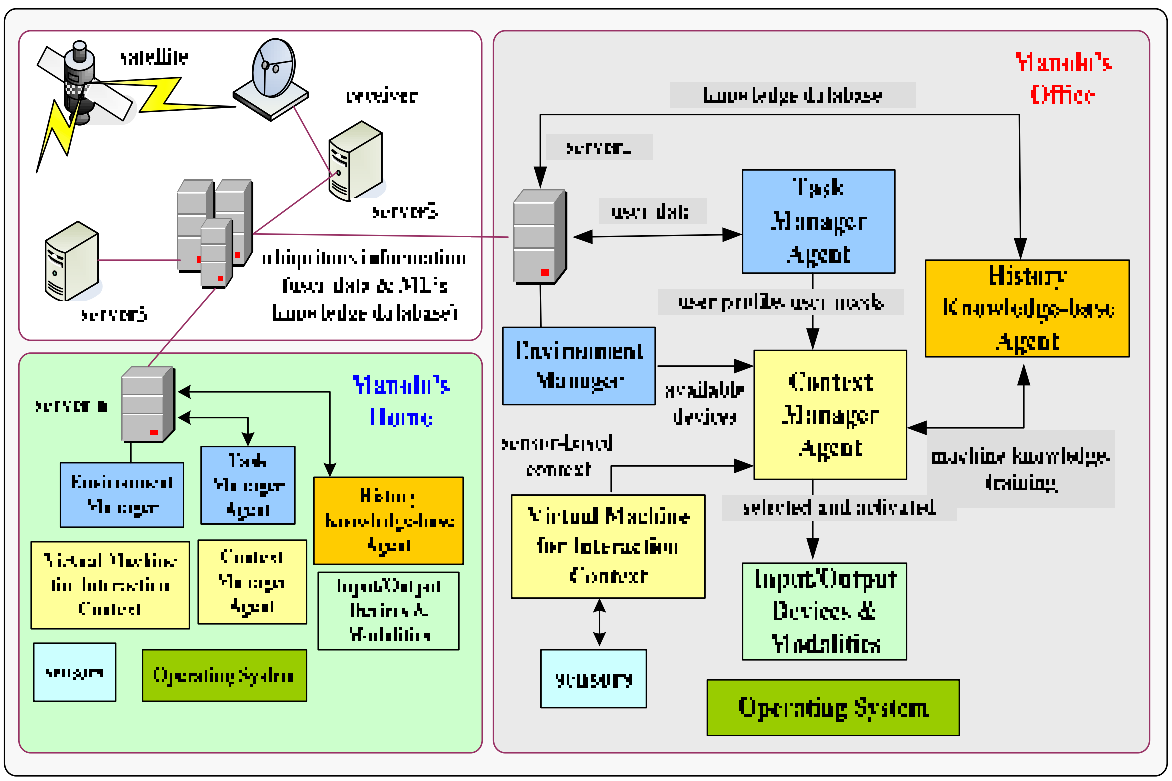


Figure 4.17 The architecture of a context-aware ubiquitous multimodal computing system.

In the diagram, the user (Manolo) can work at home, logs out, and still continue working on the same task at anytime and anywhere. Due to user's mobility, the variation in user's context and available resources is compensated by a corresponding variation in modality and media devices selection.

4.7 Conclusion

In this work, we presented the communication protocols to realize autonomic communication in a pervasive MM computing system. The system detects the user's context and accordingly selects the modalities that suit the context. We define the relationship between context and modality and between modality and media group. Media devices are identified by their membership to a media group. When two or more media devices of the same group are available, their selection is based on their priority ranking. We assert that defining context through parameters should be incremental and based on the dynamic needs of the system. We therefore adopted a layered virtual machine to realize incremental interaction context. It allows user to add, modify, and delete one context parameter at a time.

Using natural language processing as basis, we classify modality as either an input or an output. Then, modalities are further classified based on the body part that uses the modality to input data and the body part that uses the modality to receive output. The same principle is used for media classification, with minor additions. In this work, media are physical devices (and a few software) that support modality. We laid out rules for prioritizing media devices. Device activation and replacement to a failed device depends on this priority ranking.

The system's knowledge acquisition is presented using a specimen interaction context, composed of specimen parameters, namely: user location, noise level, the safety factor, the user profile and the user's computing device. The ML's progressive knowledge acquisition is also applied on context parameters and interaction contexts. When a device failed, a replacement is searched from a list of devices in the same media group within the MDPT. When a new device is introduced onto the system for the first time, all the MDPT's are updated, and the priority rankings of media are updated in each possible scenario.

4.8 References

"HPSim." from http://www.winpesim.de/.

. "Noise Measuring Device." from http://www.bappu.com/.

. "Noise Measuring Device, ELK Company, from http://www.bappu.com/.

Antoniol, G. and Penta, M. D. (2004). *A Distributed Architecture for Dynamic Analyses on User-profile Data*. 8th European Conference on Software Maintenance and Reengineering.

Archambault, D. (1999). *BrailleSurf : An HTML Browser for Visually Handicapped People*. Conference CSUN. Los Angeles, USA.

Awdé, A., Hina, M. D., et al. (2006). *A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users*. GPC 2006, 1st International Conference on Grid and Pervasive Computing, Tunghai University, Taichung, Taiwan, Springer - Verlag, LNCS.

Bellik, Y. (1995). *Interfaces multimodales : concepts, modéles et architectures*. LIMSI. Paris, Université d'Orsay.

Bougant, F., Delmond, F., et al. (2003). "*The User Profile for the Virtual Home Environment*." IEEE Communications Magazine 41(1): pp. 93-98.

Brown, P. J., Bovey, J. D., et al. (1997). *Context-Aware Applications: From the Laboratory to the Marketplace*. IEEE Personal Communications. 4(1): pp. 58 - 64.

Coutaz, J., Crowley, J. L., et al. (2005). "*Context is key*." Communications of the ACM 48(3): pp. 49-53.

Dey, A. K. (2001). "*Understanding and Using Context* " Springer Personal and Ubiquitous Computing 5(1): pp. 4 - 7.

Dey, A. K. and Abowd, G. D. (1999). *Towards a Better Understanding of Context and Context-Awareness*. 1st Intl. Conference on Handheld and Ubiquitous Computing, Karlsruhe, Germany, Springer-Verlag, LNCS 1707.

Dey, A. K., Salber, D., et al. (1999). *An Architecture to Support Context-Aware Applications*. GVU Technical Report GIT-GVU-99-23.

Garlan, D., Siewiorek, D., et al. (2002). "*Project Aura: Towards Distraction-Free Pervasive Computing*." Special Issue, Integrated Pervasive Computing Environments 21(2): pp. 22 - 31.

Herbordt, W., Horiuchi, T., et al. (2005). *Noise-Robust Hands-Free Speech Recognition on PDA's Using Microphone Array Technology*. Autumn Meeting of the Acoustical Society of Japan, Sendai, Japan.

Hina, M. D., Tadj, C., et al. (2006). *Design of an Incremental Learning Component of a Ubiquitous Multimodal Multimedia Computing System*. WiMob 2006, 2nd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Montreal, Canada, IEEE.

Lajmi, S., Ghedira, C., et al. (2007). "*Une méthode d'apprentissage pour la composition de services web*." L'Objet 8(2): pp. 1 - 4.

Oviatt, S. L. (2000). "*Designing the User Interface for Multimodal Speech and Gesture Applications: State-of-the-art Systems and Research Directions*." Human Comp. Interaction, 15(4): pp. 263-322.

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

Pascoe, J. (1998). *Generic Contextual Capabilities to Wearable Computers*. 2nd International Symposium on Wearable Computers.

Rey, G. and Coutaz, J. (2002). *Le contexteur: une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte* Proceedings of the 14th French-speaking Conference on Human-Computer Interaction (Conférence Francophone sur l'Interaction Homme-Machine) IHM '02 Poitiers, France, ACM Press.

Rey, G. and Coutaz, J. (2004). *The Contextor Infrastructure for Context-Aware Computing*. 18th European Conference on Object-Oriented Programming (ECOOP 04), Workshop on Component-Oriented Approach to Context-Aware Systems, Oslo, Norway.

Ringland, S. P. A. and Scahill, F. J. (2003). "Multimodality - The future of the wireless user interface." BT Technology Journal 21(3): pp. 181-191.

Ross, D. A. and Lightman, A. (2005). *Talking Braille: A Wireless Ubiquitous Computing Network for Orientation and Wayfinding* Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility Baltimore, MD, USA ACM Press.

Rubin, P., Vatikiotis-Bateson, E., et al. (1998). "*Audio-Visual Speech Processing*." Speech Communications 26(1-2): 1998.

Ryan, N., Pascoe, J., et al. (1997). *Enhanced Reality Fieldwork: the Context-Aware Archeological Assistant*. Exxon Computer Applications in Archeology.

Schilit, B. and Theimer, M. (1994). "*Disseminating Active Map Information to Mobile Host."* IEEE Network_8(5): pp. 22 - 32.

Schroeter, J., Ostermann, J., et al. (2000). *Multimodal Speech Synthesis*, New York, NY.

Shumate, K. (1988). *Layered virtual machine/object-oriented design*. 5th Symposium on Ada, ACM Press, New York, NY, USA.

Ward, A., Jones, A., et al. (1997). "*A New Location Technique for the Active Office*." IEEE Personal Communications: pp. 42 - 47.

CONCLUSION

This thesis is conceived as a joint research work of two academic institutions, specifically two research laboratories – the LATIS laboratory of Université du Québec, École de technologie supérieure under the supervision of Dr. Chakib Tadj on one hand, and the PRISM laboratory of Université de Versailles-Saint-Quentin-en-Yvelines under the joint supervision of Dr. Nicole Lévy and Dr. Amar Ramdane-Cherif on the other. It is therefore apparent that the subject of the thesis is a marriage between dynamic configuration of software architecture (which is within the domain of PRISM laboratory) and of multimodal signal processing and computing (which is the domain of LATIS laboratory).

The resulting work – **the paradigm of interaction context-sensitive pervasive multimodal multimedia computing system** – is a design of an intelligent system that is composed of robust components. It is a design that supports the notions of pervasive computing, multimedia and multimodal computing, dynamic configuration of software architecture and machine learning.

We have achieved what we have aimed for in our general objective and accordingly have contributed in the advancement of pervasive multimodality.

We support the vision of *transparency* and *calm technology* in Marc Weiser's vision of *ubiquitous computing*. In our work, the computer, as it takes its dynamic configuration to adapt itself to the current instance of interaction context, becomes an information tool that does not demand the focus and attention of the user. The adaptation of the system to provide the end user with the necessary and suitable modality of human-machine communication, supported by its associated media devices yields a result in which the end user concentrates on his computing task and not bothers himself with the intricacies of context awareness and the system's adaptation to it. Our concept of automation of pervasive task goes even further to the extent that the system itself associates the user files with applications, each application with its suppliers, and each supplier with its corresponding QoS dimensions configurations. In effect, the realization of this concept renders the computer to become servant to the user needs rather than the user spending time and effort to serve the needs of the computer.

In the conceptualization of context-aware pervasive multimodal system, efforts have been expended to consider a much broader context that is also reflective of the needs of the end user. This is accomplished through a concept called incremental definition of interaction context, realized through layered virtual machine for incremental interaction context.

In connection with our idea that context should be inclusive – meaning that it will fit all definitions of context given by previous researchers – we broaden the notion of context to become *context of interaction*. Interaction context refers to the *collective context* of the user (i.e. *user context*), of his working environment (i.e. *environment context*) and of his computing system (i.e. *system context*). Each of these interaction context elements – user context, environment context and system context – is composed of *various parameters* that describe the state of the user, of his workplace and his computing resources as he undertakes an activity in accomplishing his computing task, and each of these parameters may evolve over time. To realize the incremental definition of incremental context, we developed a tool called *layered virtual machine for incremental interaction context* which can be used to add, modify and delete a context parameter on one hand and determine the sensor-based context (i.e. context that is based on parameters whose values are obtained from raw data supplied by sensors) on the other.

To benefit from the *richness of interaction context* with regards to communication in human-machine interaction, we invoke the adoption of *multimodality* which allows for a much wider range of modes and forms of communication, selected and adapted to suit the given user's context of interaction, by which the end user can transmit data with computer and computer responding or yielding results to the user's queries. In multimodal communication, multimodality is beneficial to the end user because with multimodality, the weaknesses of one mode of interaction, with regards to its suitability to a given situation, is compensated by replacing it with another mode of communication that is more suitable to the situation. For example, when the environment becomes disturbingly noisy, using vocal input may not be ideal for data input; instead, the system would advocate the use of manual or visual data input. Multimodality also promotes inclusive informatics as those with permanent or temporary disability are given the opportunity to use and benefit from information technology advancement. Since mobile computing is within our midst and wireless communication is available to promote access to information and services, *pervasive* and *adaptive multimodality* is more than ever apt to enrich communication in human-computer interaction and in providing the most suitable modes for data input and output in relation to the evolving context of interaction. This research work is a contribution to this domain.

In our investigation with the current state of the art, we come to realize that a great deal of efforts were exerted and expended in defining what context is all about, how to acquire it, how to disseminate it within the system and use it to suit the needs of a system in a specific domain of application (e.g. healthcare, education, etc.). Also, our analysis shows us that a great deal of research efforts on ubiquitous computing were devoted on some application domains (e.g. identifying the user whereabouts, identifying services and tools, etc.) but there was no effort made with regards to making *multimodality pervasive and accessible to various user situations*. To this end, our research provides for the much needed solutions and answers. Our work – **the paradigm of an interaction context-sensitive pervasive multimodal multimedia computing system** is an architectural design that exhibits *adaptability* to a much larger and inclusive context called interaction context. It is intelligent and pervasive, functional even when the end user is stationary, mobile or on the go. It has mechanisms serving two distinct purposes. First, given an instance of interaction context, one which evolves over time, our system *determines the optimal modalities* that suit such interaction context. By optimal, we mean the selection is based on the trade-offs on appropriate multimodality after considering the given interaction context, available media devices that support the modalities and user preferences. We designed a mechanism (i.e. a paradigm) that does this task and simulated its functionality with success. This mechanism employs *machine learning* and uses *case-based reasoning with supervised learning*. An input to this decision-making component is an instance of interaction context and its output is the most optimal modality and its associated media devices that are for activation. This mechanism is tasked to continuously monitor the user's context of interaction and on behalf of the user continuously adapts accordingly. This *adaptation* is through *dynamic reconfiguration* of the *pervasive multimodal system's architecture*. Second, given an instance of interaction context and the user's task and preferences, we designed a mechanism that allows the *automatic selection of user's applications, the preferred suppliers* to these applications and the *preferred quality of service (QoS) dimensions'* configurations of these suppliers. This mechanism does its task in consultation with computing resources, sensing the available suppliers and possible configuration restrictions within the given computing set-up.

We also formulated scenarios on the provision of *user interface* once we have already identified that optimal modalities are available to support the given instance of user's context of interaction. We presented these possible configurations of *unimodal* and *bimodal interfaces* in consideration with the given interaction context as well as *user preferences*.

Our work differs from previous works in the domain in the sense that while others capture, disseminate and consume context to suit its preferred domain of application, our system captures the context of interaction and *reconfigures* its *architecture dynamically* in *generic* fashion with the aim that the user may continue working on his task anytime, anywhere he wishes regardless of the application domain he wishes to undertake. In effect, the system that we come up with, being generally generic in design, can be adapted or integrated with ease or with very little amount of modification to various computing systems of various domains of applications. This is our main *contribution*.

We provided simulations using formal specifications (using Z language and stochastic Petri Net) and mathematical formulations to support our ideas and concept in relation to the design of the paradigm. An actual program in Java was developed to support the layered virtual machine for incremental interaction context.

FUTURE DIRECTIONS

We are, at present, still in the infancy stage of pervasive computing environments and much more developments in this domain are yet to come. For one, future pervasive environments are likely to immerse users in a consistent world of probes, sensors and context detections. Multimodal interfaces combined with social computing interactions and high-performance networking can foster a new generation of pervasive environments. Future computer terminals will all be equipped with sensors, such as video cameras or audio microphones, capable of collecting information from the environment. Sooner, our own voice, hands, eyesight and the whole body will even be equipped with sensors that we will become the ultimate mobile multimodal input devices. In this new paradigm, a much richer interaction context will be made available to applications. The applications themselves will be more proactive than ever, as other elements will be taken into consideration, such as where is the user when a certain application-related event takes place, where the user is heading to or even whether the user is alone or accompanied by others. In this regard, our idea of *incremental interaction context* is important towards the realization of this new futuristic paradigm.

The future, more than ever, seems to point towards further advancement of pervasive multimodality. Our paradigm for pervasive multimodality and the selection of optimal modalities based on a given instance of interaction context will still hold true, even with this futuristic advancement. In the future, however, selecting and activating a single optimal input modality and a single optimal output modality will not be sufficient anymore due to the complexity of context and what the application software needs to do. Instead, two or more input and output modalities may be selected and activated and the use of one modality complements the others. Hence, there is a need for a comprehensive research in the domain of multimodal fusion (and fission) if we are to advance further in pervasive multimodality. In the *multimodal fusion*, the user may opt to use one modality at one time, and then another modality another time, the difference between these times may even be in milliseconds. There may even be a need for complementation of information involving two or more modalities, such as clicking a mouse and using speech together to denote data input. There are ongoing researches in this domain but the actual implementation and use of the proposed paradigms are rarely existent in regular computing set-ups involving ordinary, regular users. In this regard, further research in this domain is highly sought.

The current implementation of adaptation of various applications with regards to a given instance of interaction context is very slow. *Regular software applications, such as word processor, video/audio player and spreadsheets do not even evolve in the last 5 years*. To run these applications, it is always been the case that the user will seek what file to open and where the file is to be saved, and once a document is opened, the user seeks to position the cursor to a point where it was before. Some features of these applications, such as word processor's spelling checker, were developed 5 years ago. New features – such as automatic positioning of the cursor to its previous position or proposing words as the user types – are missing. This is in complete contrast to web browsers where new features are developed and integrated as newer versions are proposed to the users. Google Chrome, for instance, proposes commonly used words, sites or queries as a user types in the browser. It adapts accordingly whenever some features are not functioning as fast as they are intended to be. In this regard, there is a need to inject inputs into the advancement in the adaptation of the software applications with regards to the given context of the user and the computing system. The monopoly of some giant software corporations such as Microsoft, maker of MSWord is bad for everyone as we users gets stuck on using their software packages that barely evolve and adapt to the need of the users yet are constantly being repackaged as new software for everyone to buy in a much higher price. New mindset and new ideas are needed, more than ever. We need software applications that adapts to interaction context and can accommodate user inputs via various modalities such as speech, gesture and eye gaze. They barely exist these days. Various and lucrative opportunities exist for creative minds with entrepreneurial skills.

There are other related research works in the domain of pervasive multimodal multimedia computing where advancement is being sought. Among them would be: (a) *Pervasive Healthcare* – providing medical services to people even when physicians and medical practitioners themselves are physically distant from the patients; access to medical information whenever and wherever we need them, etc., (b) *Multimodal Interfaces* – data input and data output through multiple modalities that are suitable to the given interaction context, (c) *Pervasive Security* – making our homes and property secure from intruders, implemented through various context and risk detection using probes, sensors and actuators, (d) *Computer-Assisted Convalescent Hospital* – detects a patient's context and provide computer-assisted services, such as turning lights on/off, reminding patients to take medication, brush teeth, etc., (e) *Pervasive Banking for the Disabled* – adaptable banking machines and services based on the given disability of the client.

ANNEXE I

DETAILED CONTRIBUTION

Articles published as book chapters:

1. Manolo Dulva Hina, Chakib Tadj, Amar Ramdane-Cherif, and Nicole Levy, *"A multi-agent based multimodal system adaptive to the user's interaction context"*, a chapter in the book "*Multi-Agent Systems*", ISBN 978-953-7619-X-X, Publisher: INTECH (Accepted June 2010, book to be published in 2011).

2. Manolo Dulva Hina, Chakib Tadj, Amar Ramdane-Cherif, and Nicole Levy, "*Autonomic Communication in Pervasive Multimodal Multimedia Computing System*", a chapter in the book "*Autonomic Communication*" by Vasilakos, A., Parashar, M., Karnouskos, S., and Pedrycz, W., Springer, March 2009, pp. 251-283.

Articles published in refereed journals:

1. Manolo Dulva Hina, Amar Ramdane-Cherif, Chakib Tadj and Nicole Levy, "*Infrastructure of a Context Adaptive and Pervasive Multimodal Multimedia Computing System*", Journal of Information, Intelligence and Knowledge, Vol. 1, Issue 3, Nova Science Publishers, 2009, pp. 281 - 308.

2. Ali Awde, Manolo Dulva Hina, Chakib Tadj, Yacine Bellik and Amar Ramdane-Cherif, "*An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually-Impaired Users*", Journal of Multimedia, Vol. 4, No. 4, 2009, Academy Publisher, pp. 182-195.

3. Ali Awde, Manolo Dulva Hina, Chakib Tadj, Yacine Bellik and Amar Ramdane-Cherif, "*Middleware for Ubiquitous Access to Mathematical Expressions for Visually-Impaired*

Users", Ubiquitous Computing and Communication Journal, Vol. 3, Issue 5, November 2008, pp. 1 - 14.

4. Manolo Dulva Hina, Chakib Tadj, Amar Ramdane-Cherif, and Nicole Levy, "*Towards a Context-Aware Adaptive and Pervasive Multimodality*", Research in Computing Science Journal, Vol.29 Special Issue on "*Advances in Computer Science and Engineering*", ISSN 1870-4069, Instituto Politecnico National, Mexico, November 2007

Articles published in international conferences:

1. Amar Ramdane-Cherif, Manolo Dulva Hina, Chakib Tadj, and Nicole Levy, "*Analysis of a New Ubiquitous Multimodal Multimedia Computing System*", ISM 2007, IEEE International Symposium on Multimedia, Taichung, Taiwan, 10 – 12 December 2007.

2. Manolo Dulva Hina, Chakib Tadj, Amar Ramdane-Cherif, and Nicole Levy, "*Towards a Context-Aware Adaptive and Pervasive Multimodality"*, Poster, 16th International Conference on Computing, Mexico City, Mexico, 4 – 9 November 2007.

3. Amar Ramdane-Cherif, Manolo Dulva Hina, Chakib Tadj, and Nicole Levy, "*Quality Attribute-Driven Software Architecture of a Pervasive Multimodal Computing System*", CAINE 2007, 20th International Conference on Computer Applications in Industry and Engineering, San Francisco, CA, USA, 7 – 9 November 2007.

4. Ali Awde, Manolo Dulva Hina, Yacine Bellik, Amar Ramdane-Cherif and Chakib Tadj,, "*Task Migration in a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users"*, GPC 2007, Proceedings, 2nd Intl. Conference on Grid and Pervasive Computing, Paris, France, 2 – 4 May 2007.

5. Manolo Dulva Hina, Chakib Tadj and Amar Ramdane-Cherif, "*Self-Management Considerations in Designing a Pervasive Multimodal Multimedia Computing System*",

Proceedings – ISCIT 2006, IEEE International Symposium on Communications and Information Technology, Bangkok, Thailand, 18 – 20 October 2006.

6. Chakib Tadj, Manolo Dulva Hina, Ghislain Ngantchaha and Amar Ramdane-Cherif, "*The LATIS Pervasive Patient Subsystem: Towards a Pervasive Healthcare System*", Proceedings – ISCIT 2006, IEEE International Symposium on Communications and Information Technology, Bangkok, Thailand, 18 – 20 October 2006.

7. Manolo Dulva Hina, Chakib Tadj and Amar Ramdane-Cherif, "*Machine Learning-Assisted Device Selection in a Context-Sensitive Ubiquitous Multimodal Multimedia Computing System*", Proceedings - ISIE 2006, IEEE International Symposium on Industrial Electronics, Vol. 4. Montreal, Canada, 9-13 July 2006, pp. 3014 – 3019.

8. Ali Awde, Manolo Dulva Hina, Chakib Tadj, Amar Ramdane-Cherif and Yacine Bellik, "*Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users*", Proceedings - ISIE 2006, IEEE Intl. Symposium on Industrial Electronics, Vol. 4, Montreal, Canada, 9-13 July 2006, pp. 2834 – 2839.

9. Manolo Dulva Hina, Chakib Tadj and Amar Ramdane-Cherif, "*Design of an Incremental Machine Learning Component of a Ubiquitous Multimodal Multimedia Computing System*", Proceedings – WiMob 2006, IEEE 2nd International Conference on Wireless and Mobile Computing, Networking and Communications, Montreal, Quebec, Canada, 19 – 21 June 2006, pp. 434 – 441.

10. Manolo Dulva Hina, Chakib Tadj and Amar Ramdane-Cherif, "*Attribute-driven Design of an Incremental Learning Component of a Ubiquitous Multimodal Multimedia Computing System*", Proceedings - IEEE CECE 2006 Canadian Conference on Electrical and Computer Engineering, Ottawa, Canada, 7 – 10 May 2006, pp. 323 – 327.

11. Ali Awde, Manolo Dulva Hina, Chakib Tadj, Amar Ramdane-Cherif and Yacine Bellik, "*A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users*", Proceedings - GPC 2006, 1st International Conference on Grid and Pervasive Computing, Tunghai University, Taichung, Taiwan, 3 – 5 May 2006, pp. 620-633.

12. Manolo Dulva Hina, Chakib Tadj and Amar Ramdane-Cherif, "*From Requirements Analysis to Architecture Evaluation of a Ubiquitous Multimodal Multimedia Computing System*", Proceedings - CUSEC 2006, Canadian University Software Engineering Conference, Montreal, Quebec, Canada, 19 – 21 January 2006.

13. Manolo Dulva Hina, Amar Ramdane-Cherif, Chakib Tadj, "*A Ubiquitous Context-sensitive Multimodal Multimedia Computing and Its Machine-Learning Assisted Reconfiguration at the Architectural Level*", MIPR '05, IEEE International Workshop on Multimedia Information Processing and Retrieval, Irvine, California, USA, December 2005, pp. 585 – 591.

14. Manolo Dulva Hina, Amar Ramdane-Cherif, Chakib Tadj, "*A Context-Sensitive Incremental Learning Paradigm of a Ubiquitous Multimodal Multimedia Computing System*", Proceedings - WiMob 2005, IEEE 1st International Conference on Wireless and Mobile Computing, Networking and Communications, Vol. 4, Montreal, Quebec, Canada, 22 – 24 August 2005, pp. 104 - 112.

BIBLIOGRAPHY

. "Centre for Pervasive Healthcare." from http://www.pervasivehealthcare.dk/.

. "HPSim." from http://www.winpesim.de/.

. "Noise Measuring Device." from http://www.bappu.com/.

. Abowd, G. D. and Mynatt, E. D. (2000). *"Charting Past, Present and Future Research in Ubiquitous Computing."* ACM Transactions on Computer-Human Interaction 7(1): pp. 29 - 58.

Alpaydin, E. (2004). *"Introduction to Machine Learning"*. Cambridge, Massachusetts, MIT Press.

Antoniol, G. and Penta, M. D. (2004). *"A Distributed Architecture for Dynamic Analyses on User-profile Data"*. 8th European Conference on Software Maintenance and Reengineering.

Archambault, D. (1999). *BrailleSurf : An HTML Browser for Visually Handicapped People*. Book *"Conference CSUN"*. Los Angeles, USA**.**

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

Awde, A., Hina, M. D., et al. (2007). *"Task Migration in a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users"*. GPC 2007 - 2nd International Conference on Grid and Pervasive Computing, Paris, France, Springer Verlag.

Awdé, A., Hina, M. D., et al. (2006). *"A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users"*. GPC 2006, 1st International Conference on Grid and Pervasive Computing, Tunghai University, Taichung, Taiwan, Springer - Verlag, LNCS.

Bachmann, F. and Bass, L. (2001). *"Introduction to the Attribute Driven Design Method"*. Proceedings - IEEE Computer Society Intl. Conference on Software Engineering, Toronto, Ontario, Canada.

Bachmann, F., Bass, L., et al. (2005). *"Designing Software Architectures to Achieve Quality Attribute Requirements."* IEE Proceedings: Software 152(4): pp. 153-165.

Banavar, G., Beck, J., et al. (2000). *"Challenges: An Application Model for Pervasive Computing"*. 6th Annual Intl. Conf. on Mobile Computing and Networking (MobiCom 2000), Massachusetts, USA.

Bellifemine, F. L., Caire, G., et al. (2007). *"Developing multi-agent systems with JADE"*. Hoboken, NJ, John Wiley.

Bellik, Y. (1995). *"Interfaces multimodales : concepts, modéles et architectures."*. LIMSI. Paris, Université d'Orsay.

Benarif, S. (2008). *"Plate-forme multi-agent pour la réconfiguration dynamiques des architectures logicielles"*. Informatqiue. Versailles, France, Université de Versailles-Saint-Quentin-en-Yvelines. Thèse Ph.D.

Bennett, F., Richardson, T., et al. (1994). *"Teleporting - Making Applications Mobile"*. IEEE Workshop on Mobile Computing Systems and Applications. Santa Cruz, California**:** pp. 82 - 84.

Bougant, F., Delmond, F., et al. (2003). *"The User Profile for the Virtual Home Environment."* IEEE Communications Magazine 41(1): pp. 93-98.

Bouhuys, A. (1995). *" Induction of depressed and elated mood by music influences - the Perception of Facial Emotional Expressions in Healthy Subjects."* Journal of Affective Disorders 33(1): pp. 278 - 282.

Brown, P. J., Bovey, J. D., et al. (1997). *"Context-Aware Applications: From the Laboratory to the Marketplace"*. IEEE Personal Communications. 4(1)**:** pp. 58 - 64.

Brown, P. J., Bovey, J. D., et al. (1997). *"Context-Aware Applications: From the Laboratory to the Marketplace."* IEE Personal Communications 4(1): pp. 58 - 64.

Buxton, W. (1983). *" Lexical and Pragmatic Considerations of Input Structures."* Computer Graphics Journal of the association of computing and machinery 17(1): pp. 31 - 37.

Chen, G. and Kotz, D. (2000). *"A Survey of Context-Aware Mobile Computing Research"*. Technical Report, Department of Computer Science, Darmouth College.

Chen, H., Finin, T., et al. (2004). *"An Ontology for Context Aware Pervasive Computing Environments."* Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review 18(3): pp. 197 - 207.

Chibani, A. (2006). *"Intergiciel multi agents orienté web sémantique pour le développement d'applications ubiquitaires sensible au contexte"*. Informatique. Paris, France, Université Paris XII - Val de Marne. Thèse Ph.D.

Clements, P., Garlan, D., et al. (2003). *"Documenting software architectures: Views and beyond"*, Portland, OR, United States, IEEE Computer Society.

Clements, P., Kazman, R., et al. (2002). *Evaluating Software Architecture: Methods and Case Studies*, Addison-Wesley Professional.

CMU_CS. (2010). *"Speech at CMU."* from www.speech.cs.cmu.edu.

Coutaz, J., Crowley, J. L., et al. (2005). *"Context is key."* Communications of the ACM 48(3): pp. 49-53.

CS_Rutgers. (2010). *"The Vision, Interaction, Language, Logic and Graphics Environment."* from http://www.cs.rutgers.edu/~village/.

da Costa, C. A., Yamin, A. C., et al. (2008). *"Toward a General Infrastructure for Ubiquitous Computing."* IEEE Pervasive Computing, Special Issue on Integrated Pervasive Computing Environments 7(1): pp. 64 -73.

DeVaul, R. W. and Pentland, A. S. (2000). *"The Ektara Architecture: The Right Framework for Context-Aware Wearable and Ubiquitous Computing Applications"*. MIT Technical Report.

Dey, A. K. (2000). *"Providing Architectural Support for Building Context-Aware Applications"*. Computer Science. Atlanta, Georgia, USA, Georgia Institute of Technology. Ph.D. Thesis.

Dey, A. K. (2001). *"Understanding and Using Context "* Springer Personal and Ubiquitous Computing 5(1): pp. 4 - 7.

Dey, A. K. and Abowd, G. D. (1999). *"Towards a Better Understanding of Context and Context-Awareness"*. 1st Intl. Conference on Handheld and Ubiquitous Computing, Karlsruhe, Germany, Springer-Verlag, LNCS 1707.

Dey, A. K., Salber, D., et al. (2001). *"A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications."* Human- Computer Interaction Journal 16(2-4): pp. 97-166.

Dey, A. K., Salber, D., et al. (1999). *"The Conference Assistant: Combining Context-Awareness with Wearable Computing"*. Digest of Papers, The 3rd International Symposium on Wearable Computers.

Dey, A. K., Salber, D., et al. (1999). *"An Architecture to Support Context-Aware Applications"*. GVU Technical Report GIT-GVU-99-23.

Djenidi, H. (2007). *"Architectures logicielles dynamiques dédiées aux applications multimodales"*. Génie électrique. Montréal, Université du Québec, École de technologie supérieure & Université de Versailles-Saint-Quentin-en-Yvelines. Thèse Ph.D.

Djenidi, H., Benarif, S., et al. (2004). *Generic multimedia multimodal agents paradigms and their dynamic reconfiguration at the architectural level*. Book *"Eurasip Journal on Applied Signal Processing"*, Hindawi Publishing Corp., NY, USA, 2004**:** pp. 1688-1707.

Djenidi, H., Lévy, N., et al. (2004). *"Generic Multi-Agents Architecture for Multimedia Multimodal Software Environment."* International Journal of Object Technology 3(8,): pp. 147-168.

Djenidi, H., Ramdane-Cherif, A., et al. (2002). *"Dynamic Multi-agent Architecture for Multimedia Multimodal Dialogs"*. IEEE Workshop on Knowledge Media Networking.

Djenidi, H., Ramdane-Cherif, A., et al. (2003). *"Generic Multimedia Multimodal Agents Pradigms and their Dynamic reconfiguration at the Architecture Level. Architecture for Software Environment."* Eurasip : European Journal of Apllied Signal Processing-Special Issue on Multimedia Applications.

Dong, S. H., Xiao, B., et al. (2000). *"Multimodal user interface for internet."* Jisuanji Xuebao/Chinese Journal of Computers 23(12): pp. 1270-1275.

Efstratiou, C., Cheverst, K., et al. (2001). *"An Architecture for the Effective Support of Adaptive Context-Aware Applications"*. 2nd Int. Conf. in Mobile Data Management (MDM'01), Hong Kong.

Elrod, S., Bruce, R., et al. (1992). *"Liveboard: a large interactive display supporting group meetings, presentations and remote collaboration"*. ACM Conference on Human Factors in Computing Systems, Monterey, CA, USA.

Esler, M., Hightower, J., et al. (1999). *"Next Century Challenges: Data-Centric Networking for Invisible Computing"*. 5th Annual Intl. Conference on Mobile Computing Networking (MobiCom'99), Seattle, WA, USA.

Eustice, K., Lehman, T. J., et al. (1999). *"A Universal Information Appliance."* IBM Systems Journal 38(4): pp. 575-601.

Gajos, K. (2010). *"SUPPLE: Automatically Generating Personalized User Interfaces."* from http://iis.seas.harvard.edu/projects/supple/.

Gajos, K. Z. (2008). *"Automatically Generating Personalized User Interfaces"*. Seattle, WA, USA, University of Washington. Ph.D. Thesis.

Gajos, K. Z., Wobbrock, J. O., et al. (2007). *"Automatically generating user interfaces adapted to users' motor and vision capabilities"*. UIST '07, ACM Symposium on User Interface Software and Technology. NY, USA, ACM Press, pp. 231-240.

Garlan, D., Siewiorek, D., et al. (2002). *"Project Aura: Towards Distraction-Free Pervasive Computing."* Special Issue, Integrated Pervasive Computing Environments 21(2): pp. 22 - 31.

Garlan, D., Siewiorek, D., et al. (2002). *"Project Aura: Towards Distraction-Free Pervasive Computing."* IEEE Pervasive Computing, Special Issue on Integrated Pervasive Computing Environments, 21(2): pp. 22 - 31.

Giraud-Carrier, C. (2000). *"A Note on the Utility of Incremental Learning"*. AI Communications, IOS Press, Amsterdam, Netherlands.

Grimm, R., Anderson, T., et al. (2000). *"A System Architecture for Pervasive Computing"*. ACM SIGOPS European Workshop: Beyond the PC: New Challenges for Operating System, Kolding, Denmark, ACM.

Gwizdka, J. (2000). *"What's in the Context?"*. Workshop on Context-Awareness, CHI 2000, Conference on Human factors in computing systems, The Hague, Netherlands.

Held, A. (2002). *"Modeling of Context Information for Pervasive Computing Applications"*. 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI2002), Orlando, Florida.

Herbordt, W., Horiuchi, T., et al. (2005). *"Noise-Robust Hands-Free Speech Recognition on PDA's Using Microphone Array Technology"*. Autumn Meeting of the Acoustical Society of Japan, Sendai, Japan.

Hina, M. D., Tadj, C., et al. (2006). *"Design of an Incremental Learning Component of a Ubiquitous Multimodal Multimedia Computing System"*. WiMob 2006, 2nd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Montreal, Canada.

Hina, M. D., Tadj, C., et al. (2006). *"From Requirements Analysis to Architecture Evaluation of a Ubiquitous Multimodal Multimedia Computing System"*. CUSEC 2006, 3rd Canadian University Software Engineering Conference, Montreal, QC, Canada.

Hina, M. D., Tadj, C., et al. (2006). *"Machine Learning-Assisted Device Selection in a Context-Sensitive Ubiquitous Multimodal Multimedia Computing System"*. IEEE ISIE 2006, International Symposium on Industrial Electronics, Montreal, Quebec, Canada.

Hina, M. D., Tadj, C., et al. (2006). *"Attribute-Driven Design of an Incremental Learning Component of a Ubiquitous Multimodal Multimedia Computing System"*. IEEE CCECE '06, Canadian Conference on Electrical and Computer Engineering, Ottawa, Canada.

Horn, P. (2001). *"Autonomic Computing: IBM's Perspective on the State of Information Technology"*, IBM Research: pp. 1 - 22.

Hunter, D., Ayers, D., et al. (2007). *"Beginning XML"*, Wrox Printing Inc.

Indulska, J., Loke, S. W., et al. (2001). *"An Open Architecture for Pervasive Systems"*. 3rd Intl. Work. Conf. on Distributed Applications and Interoperable Systems (DAIS 2001), Kraków, Poland.

Indulska, J., Robinson, R., et al. (2003). *"Experiences in Using CC/PP in Context-Aware Systems"*. 4th International Conference on Mobile Data Management, Melbourne, Australia.

Jensen, K., Kristensen, L. M., et al. (2007). *"Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems."* International Journal on Software Tools for Technology Transfer, Springer Verlag 9(1): pp. 213 - 254.

Kallenberg, O. (2002). *"Foundations of Modern Probability"*, Springer.

Kantarjiev, C. K., Demers, A., et al. (1993). *"Experiences with X in a wireless environment"*. Mobile & Location-Independent Computing Symposium on Mobile & Location-Independent Computing, Cambridge, MA, USA.

Kephart, J. O. and Chess, D. M. (2001). *"The Vision of Autonomic Computing"*, IBM Thomas J. Watson Research Centre: pp. 41 - 50.

Kindberg, T. and Barton, J. (2001). *"A Web-Based Nomadic Computing System."* Elsevier Computer Networks, 35(4): pp. 443-456.

Kolodner, J. (1993). *"Case-Based Reasoning"*. San Mateo, CA, Morgan Kaufman.

Lajmi, S., Ghedira, C., et al. (2007). *"Une méthode d'apprentissage pour la composition de services web."* L'Objet 8(2): pp. 1 - 4.

Liang, Y. D. (2010). *"Introduction to Java Programming, 8th ed."*, Prentice Hall.

Lightfoot, D. (2001). *"Formal Specification Using Z, 2nd ed."*, McMillan Press.

Miraoui, M. (2009). *"Architecture logicielle pour l'informatique diffuse: Modélisation du contexte et adaptation dymaique des services"*. Génie électrique. Montréal, Université du Québec, École de technologie supérieure. Thèse Ph.D.

Mitchell, T. (1997). *"Machine Learning"*, McGraw-Hill.

Oviatt, S. (2002). *"Multimodal Interfaces: Handbook of Human-Computer Interaction"*. New Jersey, USA, Lawrence Erbaum.

Oviatt, S. L. (2000). *"Designing the User Interface for Multimodal Speech and Gesture Applications: State-of-the-art Systems and Research Directions."* Human Computer Interaction 15(4): pp. 263-322.

Oviatt, S. L. (2000). *"Designing the User Interface for Multimodal Speech and Gesture Applications: State-of-the-art Systems and Research Directions."* Human Comp. Interaction, 15(4): pp. 263-322.

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

Pahlavan, K. and Krishnamurthy, P. (2002). *Principles of Wireless Networks*, Prentice Hall, ISBN 0-13-093003-2

Pascoe, J. (1998). *"Adding Generic Contextual Capabilities to Wearable Computers"*. 2nd IEEE Intl. Symposium on Wearable Computers, Pittsburg, NY, IEEE Computer Society.

Pascoe, J. (1998). *"Generic Contextual Capabilities to Wearable Computers"*. 2nd International Symposium on Wearable Computers.

Pettit, R. G. I. and Gomaa, H. (2004). *"Modeling behavioral patterns of concurrent software architectures using Petri nets"*. WICSA 2004, Proceedings, Fourth Working IEEE/IFIP Conference on Software Architecture.

Poladian, V., Sousa, J. P., et al. (2006). *"Task-based Adaptation for Ubiquitous Computing."* IEEE Transactions on Systems, Man and Cybernetics 36(6): pp. 328 - 340.

Prekop, P. and Burnett, M. (2003). *"Activities, context and ubiquitous computing."* Computer Communications 26: pp. 1168-1176.

Razzaque, M. A., Dobson, S., et al. (2005). *"Categorization and Modelling of Quality in Context Information"*. IJCAI 2005 Workshop on AI and Autonomic Communications, Edinburg, Scotland.

Rey, G. and Coutaz, J. (2002). *"Le contexteur: une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte "*. Proceedings of the 14th French-speaking Conference on Human-Computer Interaction (Conférence Francophone sur l'Interaction Homme-Machine) IHM '02 Poitiers, France, ACM Press.

Rey, G. and Coutaz, J. (2004). *"The Contextor Infrastructure for Context-Aware Computing"*. 18th European Conference on Object-Oriented Programming (ECOOP 04), Workshop on Component-Oriented Approach to Context-Aware Systems, Oslo, Norway.

Ringland, S. P. A. and Scahill, F. J. (2003). *"Multimodality - The future of the wireless user interface."* BT Technology Journal 21(3): pp. 181-191.

Ross, D. A. and Lightman, A. (2005). *"Talking Braille: A Wireless Ubiquitous Computing Network for Orientation and Wayfinding "*. Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility Baltimore, MD, USA ACM Press.

Rubin, P., Vatikiotis-Bateson, E., et al. (1998). *"Audio-Visual Speech Processing."* Speech Communications 26(1-2): pp. 1998.

Ryan, N., Pascoe, J., et al. (1997). *Enhanced Reality Fieldwork: the Context-aware Archaeological Assisstant*. Book *"Computer Applications in Archaeology"*. V. Gaffney, Leusen, M. v. and Exxon, S.

Ryan, N., Pascoe, J., et al. (1997). *"Enhanced Reality Fieldwork: the Context-Aware Archeological Assistant"*. Exxon Computer Applications in Archeology.

Salehie, M. and Tahvildari, L. (2005). *"Autonomic Computing: Emerging Trends and Open Problems"*. 2005 Workshop on Design and Evolution of Autonomic Application Software, St. Louis, Missouri, USA, ACM Press.

Satyanarayanan, M. (1990). *"Scalable, secure, and highly available distributed file access."* Computer 23(5): pp. 9-18.

Satyanarayanan, M. (1996). *"Mobile information access."* IEEE Personal Communications 3(1): pp. 26-33.

Satyanarayanan, M. (2001). *"Pervasive Computing: Vision and Challenges."* IEEE Personal Communications 8(4): pp. 10-17.

Schilit, B., Adams, N., et al. (1994). *"Context-aware computing applications"*. First Workshop on Mobile Computing Systems and Applications , WMCSA 1994, Santa Cruz, California, USA.

Schilit, B. and Theimer, M. (1994). *"Disseminating Active Map Information to Mobile Host."* IEEE Network 8(5): pp. 22 - 32.

Schilit, B. N., Adams, N., et al. (1993). *"The PARCTAB mobile computing system"*. Fourth Workshop on Workstation Operating Systems (WWOS-IV), Napa, CA, USA.

Schilit, B. N. and Theimer, M. M. (1994). *"Disseminating Active Map Information to Mobile Hosts."* IEEE Network 8(5): pp. 22 - 32.

Schroeter, J., Ostermann, J., et al. (2000). *"Multimodal Speech Synthesis"*, New York, NY.

Shumate, K. (1988). *"Layered virtual machine/object-oriented design"*. 5th Symposium on Ada, ACM Press, New York, NY, USA.

Strang, T. and Linnhoff-Popien, C. (2003). *"Service Interoperability on Context Level in Ubiquitous Computing Environments"*. Intl. Conf. on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet (SSGRR 2003), L'Aquila, Italy.

Truman, T. E., Pering, T., et al. (1998). *"The InfoPad multimedia terminal: a portable device for wireless information access."* IEEE Transactions on Computers 47(10): pp. 1073-1087.

Tse, D. and Viswanath, P. (2005). *"Fundamentals of Wireless Communication"*, Cambridge University Press.

Vallejos, J. and Desmet, B. (2007). *"Pervasive Communication: The Need for Distributed Context Adaptations"*. ECOOP 2007 Workshop on Object Technology for Ambient Intelligence and Pervasive Systems. Berlin, Germany.

Varshney, U. (2003). *"Pervasive Healthcare."* Computer 36(12): pp. 138 - 140.

Vasilakos, A. V. and Pedrycz, W. (2006). *"Ambient Intelligence, Wireless Networking, Ubiquitous Computing"*. USA, ArtecHouse.

Want, R., Hopper, A., et al. (1992). *"Active badge location system."* ACM Transactions on Information Systems 10(1): pp. 91-102.

Want, R., Schilit, B. N., et al. (1995). *"Overview of the ParcTab ubiquitous computing experiment."* IEEE Personal Communications 2(6): pp. 28-43.

Ward, A., Jones, A., et al. (1997). *"A New Location Technique for the Active Office."* IEEE Personal Communications: pp. 42 - 47.

Weiser, M. (1991). *"The computer for the twenty-first century."* Scientific American 265(3): pp. 94 - 104.

Weiser, M. (1993). *"Some computer science issues in ubiquitous computing."* Communications of the ACM 36(7): pp. 74-84.

Weiser, M. and Brown, J. S. (1996). *"Designing Calm Technology."* Powergrid Journal 1(1): pp. 94 - 110.

Witten, I. H. and Frank, E. (2005). *"Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed."*. San Francisco, Morgan Kaufmann.

Wooldridge, M. (2001). *"An Introduction to Multi-agent Systems"*. Chichester, UK, Wiley.

Zhang, Q., Imamiya, A., et al. (2004). *"A gaze and speech multimodal interface"*, Hachioji, Japan, Institute of Electrical and Electronics Engineers Inc., Piscataway, USA.