

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 ÉTAT DE L'ART SUR LA PRÉSENTATION DES EXPRESSIONS MATHÉMATIQUES À DES UTILISATEURS NON-VOYANTS	10
1.1 Définitions.....	10
1.1.1 L'interaction homme-machine.....	10
1.1.2 Le mode	11
1.1.3 La modalité	11
1.1.4 Le média.....	11
1.1.5 La forme de la présentation.....	11
1.1.6 Le contexte de l'interaction	12
1.2 Les médias et les méthodes d'accès aux informations utilisés par les non-voyants....	12
1.2.1 Le braille	13
1.2.2 Les médias supportant le mode TPK	15
1.2.3 La parole	21
1.2.4 Les sons non parlés, les bips sonores, les earcons	23
1.3 Les besoins des utilisateurs	23
1.4 Les approches pour l'accès aux mathématiques	27
1.4.1 Les formes visuelles.....	28
1.4.2 Les approches Braille/tactiles	31
1.4.3 Les approches sonores	41
1.5 Les approches pour l'édition des expressions.....	45
1.5.1 Lambda	47
1.5.2 Le projet Infty	48
1.5.3 WinTriangle	49
1.6 Comparaison et analyse des approches.....	49
1.7 Résumé : les approches pour l'accès aux informations par les non-voyants	51
1.8 La multimodalité.....	53
1.9 Utilisation des modalités.....	55
1.9.1 TYCOON.....	55
1.9.2 Propriétés CASE	56
1.9.3 Propriétés CARE.....	57
1.9.4 Fusion/Fission.....	58
1.9.5 Les types de multimodalités selon Bellik	58
1.10 Taxonomie des modalités	60
1.11 La présentation multimodale de l'information	61
1.12 Architectures des systèmes intelligents	65
1.12.1 Seeheim.....	65
1.12.2 MVC	66
1.12.3 ARCH	67
1.12.4 PAC.....	69

1.12.5	PAC-Amodeus	70
1.12.6	L'architecture O. A. A.	71
1.12.7	Le modèle à couches	73
1.12.8	Choix de l'architecture.....	74
1.13	Interfaces multimodales pour les non-voyants	75
1.14	Résumé : les interactions multimodales.....	77
1.15	Conclusion du chapitre 1	78
CHAPITRE 2 INFRASTRUCTURE OF AN ADAPTIVE MULTI-AGENT SYSTEM FOR PRESENTATION OF MATHEMATICAL EXPRESSION TO VISUALLY-IMPAIRED USERS		82
2.1	Introduction.....	82
2.2	Review of the State of the Art.....	83
2.3	Technical Challenges	85
2.4	The Components of an Adaptive Multimodal Computing System.....	86
2.4.1	Our Adaptive Multimodal Computing System for Mathematical Expression Presentation to Visually Impaired Users.....	86
2.4.2	Modality and Media.....	88
2.4.3	The Context Gathering Layer	89
2.4.4	The Control and Monitoring Layer.....	91
2.4.5	The Data Analysis Layer	92
2.4.6	The Presentation Layer	98
2.4.7	The Data Access Layer	100
2.5	Fault Tolerant System.....	101
2.5.1	General Principle of a Fault-tolerant System.....	101
2.5.2	Our Multi-agent Fault-Tolerant System.....	102
2.6	Example of Simulation with Jade	104
2.7	Conclusion	108
2.8	References.....	109
CHAPITRE 3 AN ADAPTIVE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM FOR PRESENTATION OF MATHEMATICAL EXPRESSIONS TO VISUALLY-IMPAIRED USERS		113
3.1	Introduction.....	113
3.2	Technical Challenges	116
3.3	Detailed Design and Principles for the Resolution of System Requirements.....	119
3.3.1	Using Machine Learning to Find Optimal Suitable Modalities to an Interaction Context.....	119
3.3.2	Algorithms for Finding Optimal Modalities for Interaction Context	123
3.3.3	Relationship between Modality and Media Devices	123
3.3.4	Mathematical Expression and Presentation Formats	126
3.3.5	Analysis of a Mathematical Expression.....	127
3.3.6	Selection of Presentation Format	128
3.4	Detailed Design and Principles for the Resolution of System Requirements.....	133
3.4.1	Architectural Framework	133

3.4.2	Interaction Context Specification	134
3.5	Formal Specification, Simulation and Examples	139
3.5.1	Formal Specification and Petri Net	140
3.5.2	Sample Case 1	141
3.5.3	Simulation 1	143
3.5.4	Sample Case 2	144
3.5.5	Simulation 2	145
3.6	Conclusion	146
3.7	Appendix A - figures on optimization algorithms	147
3.8	Acknowledgment	149
3.9	References	149
CHAPITRE 4 COMPLEXITY OF MATHEMATICAL EXPRESSIONS IN ADAPTIVE MULTIMODAL MULTIMEDIA SYSTEM ENSURING ACCESS TO MATHEMATICS FOR VISUALLY IMPAIRED USERS...152		
4.1	Introduction	152
4.2	Related Work	154
4.3	Purpose of the Work	156
4.4	Modality, Media and Presentation Format	158
4.4.1	Modality and Media	158
4.4.2	Presentation Format	159
4.5	Adaptive Multimodal Computing Design	160
4.5.1	Architectural Framework	160
4.5.2	Contextual Information	162
4.6	Mathematics	165
4.6.1	Mathematical Taxonomy	165
4.6.2	Principal Branches of Mathematics:	165
4.6.3	Mathematical Expression	166
4.7	The Experiment	169
4.7.1	Hypothesis	169
4.7.2	Evaluation Protocol	170
4.7.3	Experiment Running	171
4.8	Results of Experiments and Discussions	172
4.8.1	Arithmetic	173
4.8.2	Algebra	174
4.8.3	Analysis	175
4.8.4	Geometry / Trigonometry	177
4.8.5	Logic	178
4.8.6	Statistics & Probability	180
4.8.7	Combination	181
4.9	Results and Application	183
4.9.1	Results Analysis	183
4.9.2	Integration of Expression Complexity in Our System	187
4.10	Conclusion	189
4.11	Appendix	189

4.12	References.....	192
CHAPITRE 5 MIDDLEWARE FOR UBIQUITOUS ACCESS TO MATHEMATICAL EXPRESSIONS FOR VISUALLY-IMPAIRED USERS.....		
		198
5.1	Introduction.....	198
5.2	Review of the State of the Art.....	199
5.3	Challenges, Proposed Solutions and Contribution.....	202
5.4	Infrastructure and System Architecture	204
	5.4.1 System Architecture.....	204
	5.4.2 The Ubiquitous Mathematical Document.....	206
	5.4.3 Anytime, Anywhere Mathematical Computing.....	207
5.5	Designing an Interaction Context Adaptive System.....	209
	5.5.1 Theoretical Machine Learning.....	209
	5.5.2 Basic Principles of Interaction Context	211
	5.5.3 Finding Optimal Modalities for Interaction Context	215
	5.5.4 Realizing User Task Using Optimal Modalities and Supporting Media Devices.....	215
	5.5.5 Machine Learning Training for Selection of Application Supplier	216
5.6	Configuration and Optimization of Application Supplier	218
	5.6.1 Alternative Configuration Spaces.....	218
	5.6.2 Optimizing Configuration of User’s Task	219
5.7	Interaction Context, Modality and Media Devices	222
5.8	Design Specification and Scenario Simulations	225
	5.8.1 Selection of Modalities	226
	5.8.2 Selection of Media Devices	228
	5.8.3 Selection of Application Supplier.....	228
	5.8.4 Optimizing User’s Task Configuration.....	229
	5.8.5 Specification for Detecting Suitability of Modality.....	230
	5.8.6 Simulation Results	231
5.9	Conclusion	233
5.10	References.....	234
CONCLUSION		
		235
ANNEXE I CONTRIBUTION DÉTAILLÉE		
		242
BIBLIOGRAPHIE.....		
		244

LISTE DES TABLEAUX

	Page
Tableau 1.1	Les médias utilisés en entrée et en sortie dans les outils développés pour accéder aux mathématiques.52
Tableau 1.2	Les outils développés dans la littérature pour présenter les mathématiques aux non-voyants.53
Tableau 2.1	A sample Media Devices Priority Table (MDPT).104
Tableau 3.1	A sample Media Devices Priority Table (MDPT).125
Tableau 3.2	Presentation Format Priority Table.131
Tableau 3.3	Functionalities of Various Agents in Our Adaptive Multimodal Multimedia Computing System.135
Tableau 3.4	The Convention Table of User's Additional Handicap and Its Effect on User's Modality Disposition.136
Tableau 3.5	The Convention Table of Affected Modalities By Combined Noise Level and Environment Restriction.137
Tableau 3.6	The Convention Table of User's Computing Device and Its Effect on Modality Selection.139
Tableau 4.1	Functionalities of various agents in our adaptive multimodal multimedia computing system.163
Tableau 4.2	The complexities and its suiting values.171
Tableau 4.3	Experiment's results of arithmetic branch.173
Tableau 4.4	Experiment's results of algebra branch.175
Tableau 4.5	Experiment's results of analysis branch.176
Tableau 4.6	Experiment's results of trigonometry branch.177
Tableau 4.7	Experiment's results of logic branch.179

Tableau 4.8	Experiment's results of statistics & probability branch.....	180
Tableau 4.9	Experiment's results of a collection of 12 expressions.....	182
Tableau 4.10	4 similar expressions and its average complexity.....	183
Tableau 4.11	Comparison table of estimated complexity using our formula and average complexity according to experiments.....	186
Tableau 4.12	The five categories of complexity.....	187
Tableau 5.1	The tabulation for affected modalities by combined noise level and environment restriction.....	223
Tableau 5.2	The convention table of user's computing device and its effect on modality selection.....	225

LISTE DES FIGURES

		Page
Figure 1.1	Structure d'un caractère braille standard et informatique.	14
Figure 1.2	Le mot « braille » et son équivalent en braille.	14
Figure 1.3	Un document écrit en braille.	15
Figure 1.4	Le terminal braille VARIO 80.	16
Figure 1.5	La matrice qui permet d'afficher un graphique en tactile.	17
Figure 1.6	TIGER printer : imprimante de « fonte » tactile.	17
Figure 1.7	Un exemple d'un « Overlay Keyboard » développé par APH.	19
Figure 1.8	Le gant à retour d'effort Cyberglove II.	21
Figure 1.9	La structure d'un modèle dans MathType.	28
Figure 1.10	L'interface de MathType.	28
Figure 1.11	Une fraction écrite en MathML.	30
Figure 1.12	Une fraction écrite en LaTeX.	31
Figure 1.13	La transcription d'une fraction dans 4 différentes notations.	34
Figure 1.14	Une expression écrite sous trois différentes formes.	36
Figure 1.15	Un arbre qui représente une fraction.	39
Figure 1.16	Une formule présentée par EasyMath lors de l'évaluation du système.	40
Figure 1.17	Les différents types de multimodalité selon Bellik. Tirée de Bellik (1995, P.31).	59
Figure 1.18	Diagramme UML présentant la taxonomie des modalités en sortie. Seuls les deux modes utilisables par les non-voyants sont pris en compte.	61

Figure 1.19	Le concept d'élection selon Rousseau. Tirée de Rousseau (2006, P. 43)	63
Figure 1.20	Exemple de profil (arbre de pondération). Tirée de Jacquet (2006, P. 120)	64
Figure 1.21	Le modèle de Seeheim.	65
Figure 1.22	Le modèle MVC.	66
Figure 1.23	Les composants du modèle ARCH.	68
Figure 1.24	Une architecture PAC.	69
Figure 1.25	Le modèle PAC-Amodeus.	70
Figure 1.26	Un exemple de l'architecture O.A.A, le modèle communauté d'agents formant le «office assistant». Tirée de Balvet (2002, P. 94).....	72
Figure 1.27	Un exemple de l'architecture multicouche en conjonction avec le modèle multi-agent.	73
Figure 2.1	The architectural abstraction of a generic MM computing system for visually-impaired users.	87
Figure 2.2	Architectural layer view of the multimodal computing system.....	88
Figure 2.3	Attributes of device context negotiator for Braille terminal, speech recognition and synthesis.	91
Figure 2.4	The mathematical expression repository, in generic format.	92
Figure 2.5	A fraction in bi-dimensional form and its corresponding equivalent in LaTeX, MathML and Braille.	93
Figure 2.6	Two sample expressions in bi-dimensional form and its corresponding equivalent in Braille and DotsPlus format.	95
Figure 2.7	The Analysis and Conversion Agent.	95
Figure 2.8	A sample analysis of a specimen fraction.....	96
Figure 2.9	The Machine Learning Agent and its interaction with other system components.	97

Figure 2.10 The Translator Agent and its components.98

Figure 2.11 The translation of a specimen fraction into 4 formats.99

Figure 2.12 Parameters settings for presentation formats and some media devices.99

Figure 2.13 Searching agent and editor agent cooperates with Device manager agent.101

Figure 2.14 All possible states of agent and primitives in our system.102

Figure 2.15 A simple Jade simulation of SMA and DMA, also a sample MDPT.106

Figure 2.16 Sniffer agent monitors and checks ACL messages exchanged among agents SMA and DMA.107

Figure 3.1 A fraction converted into its linear representation, and its corresponding MathML and Braille representations.126

Figure 3.2 Structure of the analysis of a MathML expression.128

Figure 3.3 A sample analysis of a specimen fraction.128

Figure 3.4 The presentation format selection process based on the given interaction context and MathML expression.130

Figure 3.5 Algorithm for finding the suitable presentation format for a mathematical expression.132

Figure 3.6 Architectural layer view of our multimodal computing system for visually-impaired users.133

Figure 3.7 A sample user profile.135

Figure 3.8 The DMA, responsible for detecting user’s computing device and available media devices.138

Figure 3.9 A sample snapshot of a scenario repository (SR).142

Figure 3.10 Snapshot of the simulated interaction context case in Section 3.3.144

Figure 3.11 Selection of the optimal presentation format based on available media devices and presentation formats.145

Figure 3.12	Algorithm that calculates for the suitability score of each element M_j of the modality power set, given an interaction context IC_i	147
Figure 3.13	Algorithm for finding the optimal modality.	148
Figure 3.14	Algorithm for updating optimal modality based on the availability of media devices to support the chosen modalities.	148
Figure 4.1	A sample of media devices priority table and different modality's forms.	159
Figure 4.2	Two sample expressions in bi-dimensional form and its corresponding equivalent in Braille and DotsPlus format.	160
Figure 4.3	Architectural layer view of our multimodal computing system for visually-impaired users.	161
Figure 4.4	A sample of user profile.	163
Figure 4.5	A syntax tree of the sample expression.	169
Figure 4.6	Curves representing reading and writing times taken by participants during experiment of arithmetic branch.	174
Figure 4.7	Curve representing reading and writing times taken by participants during experiment of algebra branch.	175
Figure 4.8	Curves representing reading and writing times taken by participants during experiment of analysis branch.	176
Figure 4.9	Curves representing reading and writing times taken by participants during experiment of trigonometry branch.	178
Figure 4.10	Curves representing reading and writing times taken by participants during experiment of logic branch.	179
Figure 4.11	Curves representing reading and writing times taken by participants during experiment of probability & statistics branch.	181
Figure 4.12	Curves representing reading and writing times taken by participants during experiment of the 12 expressions.	182
Figure 4.13	Our linear programming problem.	185

Figure 4.14	Structure of the analysis of a MathML expression.	188
Figure 4.15	A sample analysis of a specimen fraction.	188
Figure 5.1	The architectural abstraction of a generic MM computing system for visually-impaired users.	205
Figure 5.2	A fraction in bi-dimensional form and its corresponding equivalent in LaTeX, MathML and Braille.	207
Figure 5.3	An anytime, anywhere adaptive computing with mathematical document for visually-impaired users.	208
Figure 5.4	A generic ML system: an instance of interaction context serves input to a ML component yielding a corresponding post-condition result.	210
Figure 5.5	The relation between pre-condition scenarios and the post-condition scenarios using supervised learning.	211
Figure 5.6	Media selections to support a modality.	215
Figure 5.7	The training process for ML knowledge acquisition: (Up) the mapping of data type to a supplier and (Down) building a user's preferred QoS dimensions for a supplier.	217
Figure 5.8	Algorithm for optimized supplier and its QoS configuration.	221
Figure 5.9	A sample user profile.	222
Figure 5.10	The system context as function of user's computing device and available media devices.	224
Figure 5.11	A sample of a scenario repository (SR).	227
Figure 5.12	A snapshot of the simulated selection of optimal modality based on interaction context.	232
Figure 5.13	Various graphs showing variations of user's satisfaction with respect to its preferred supplier and QoS dimension and their replacements.	233

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

1D	Une dimension
2D	Deux Dimensions
3D	Trois Dimensions
A.A.A.	Adaptive Agent Architecture
ACA	Analysis and Conversion Agent
ACC	Agent Communication Channel
ACL	Agent Communication Language
AMS	Agent Management System
ASCII	American Standard Code for Information Interchange
ASTER	Audio System for Technical Readings
BraMaNet	Braille Mathématique sur le Net
BRT	Braille Terminal
D.P.I.	Dots Per Inch
DB	Decibels
DCN	Device Context Negotiator
DE	Deutsch
DF	Directory Facilitator
DMA	Device Manager Agent
DMD	Dot Matrix Display
EC	Environment's Context
EN	English
EnvA	Environment Agent
FR	French
GUI	Graphic User Interface
H	Hauteur
HST	Headset
IC	Interaction Contexte
IML	Incremental Machine Learning

IP	Internet Protocol address
JADE	Java Agent Development Framework
KB	Keyboard
KD	Knowledge Database
L	Largueur
LaBraDoor	LaTeX-to-Braille-Door
LAMBDA	Linear Access to Mathematic for Braille Device and Audio-synthesis
MAC	Media Access Control Address
MAS	Multi-Agent System
MATHS	Mathematical Access for Technology and Science
MAVIS	Mathematics Accessible to Visually Impaired Students
MDPT	Media Devices Priority Table
MER	Mathematical Expression Repository
MIC	Microphone
MICOLE	Multimodal Collaboration Environment for inclusion of Visually Impaired Children
ML	Machine Learning
MLA	Machine Learning Agent
MVC	Modèle-Vue-Contrôleur
O.A.A.	Open Agent Architecture
OKB	Overlay Keyborad
PDA	Personnal Digital Assistant
PFPT	Presentation Format Priority Table
QoS	Quality-of-Service
RMA	Remote Management Agent
SC	System Context
SEA	Searching and Editing Agent
SMA	Système Multi-Agent
SPK	Speaker
SR	Scenario Repository

TA	Translation Agent
TIGER	The Tactlle Graphics EmbosseR
T _{in}	Tactile Input
T _{out}	Tactile Output
TPK	Tactilo-Proprio-Kinesthésique
TPR	Tactile Printer
TTS	Text To Speech
UA	User Agent
UC	User Context
UMA	Universal Mathematics Accessibility
UML	Unified Modeling Language
UP	User Profile
USB	Universal Serial Bus
VICKIE	Visually Impaired Children Kit for Inclusive Education
V _{in}	Vocal Input
V _{out}	Vocal Output
WEKA	Waikato Environment for Knowledge Analysis
WER	Word Error Rate
WYSIWYG	What You See Is What You Get

INTRODUCTION

Contexte de nos travaux de recherche

Ce travail s'inscrit dans le cadre de la problématique d'accès à l'information par les non-voyants. Il présente nos travaux de recherche et d'étude des interfaces multimodales en vue de proposer une architecture d'un système adaptatif pour la présentation d'expressions mathématiques à des utilisateurs non-voyants.

Avant l'apparition du braille (1829), les non-voyants n'utilisaient qu'un seul moyen de communication, la parole, faisant ainsi appel au sens de l'ouïe. Il leur était donc impossible d'accéder à des documents écrits sans l'aide d'une tierce personne. Ainsi, avec l'invention du braille, l'accès à la lecture est devenu possible en exploitant le sens du toucher. Cependant, la transcription en braille reste volumineuse et n'est pas toujours simple à manipuler. Ce n'est qu'un siècle plus tard, que l'enregistrement vocal est venu faciliter l'accès au savoir, sans toutefois remplacer le braille.

Quelques décennies plus tard, l'arrivée de l'ordinateur a permis une véritable révolution. Sa diffusion aux non-voyants a ouvert une voie prometteuse vers un espace de connaissances beaucoup plus vaste. En effet, le développement de nouvelles technologies de l'information apporte une véritable opportunité pour les personnes non-voyantes. Celles-ci ont alors accès à l'information, et donc au savoir dans les mêmes conditions (quantité et qualité) que les voyants. Grâce à l'ordinateur, les non-voyants pouvaient enfin communiquer par écrit et/ou par la voix avec les voyants en exploitant les mêmes dispositifs tels que le clavier et le haut-parleur. L'ordinateur s'impose alors parce qu'il présente un certain nombre d'avantages que les autres outils de communication n'offrent pas. Cependant, il ne réussit pas pour autant à supplanter les méthodes pré existantes qui sont la transcription en braille et l'enregistrement vocal.

Aujourd'hui, l'utilisation de l'ordinateur et des équipements informatiques constitue des outils de travail quotidiens pour la population voyante. L'accès aux données numériques a de plus en plus un impact sur la réussite de l'individu au sein de la société (performances scolaires et professionnelles). Ainsi, ne pas avoir accès à cette technologie risque d'isoler la personne et de limiter ses ressources et ses ambitions.

Dans ce contexte, l'accès aux mathématiques est primordial pour les étudiants qui désirent avancer dans les domaines scientifiques et dans les disciplines connexes telle que l'ingénierie. En effet, les mathématiques constituent une base fondamentale pour les sciences. Pour beaucoup d'étudiants voyants, apprendre les mathématiques est une tâche complexe et peut être une barrière les séparant des disciplines techniques. Pour les étudiants non-voyants, la tâche est encore plus complexe, en raison des difficultés supplémentaires auxquelles ils doivent faire face pour accéder aux mathématiques.

Dans le cadre de notre recherche, nous nous intéressons à l'accès aux données complexes telles que les formules mathématiques. La nature et la complexité de ces expressions rendent cet accès particulièrement difficile aux personnes non-voyantes. Il est donc important de mettre sur pied un projet qui consiste à étudier de nouvelles techniques d'interactions visant à faciliter cet accès, offrant ainsi aux personnes présentant un handicap visuel de nouvelles possibilités dans le domaine scientifique en leur permettant d'acquérir les bases nécessaires pour les sciences exactes.

Dans cette optique, des travaux ont permis de montrer l'intérêt de l'utilisation de l'interaction multimodale dans les applications non visuelles (par exemple, Meditor (Bellik, 1995) est un éditeur textuel multimodal pour les aveugles, BrailleSurf (Archambault, 1999) est un navigateur web pour les non-voyants).

La multimodalité combine différents modes de communication entre l'homme et la machine. Ainsi, la multimodalité permet d'adapter un système aux différentes situations d'interaction et au profil de l'utilisateur. Dans un système multimodal pour non-voyants, il est possible

d'avoir des médias (au sens de dispositifs physiques) qui remplacent ceux qui mettent à contribution la vision. Par exemple, au lieu d'utiliser un écran pour visualiser les informations, on peut utiliser un terminal braille combiné à un lecteur d'écran (traduction du terme anglo-saxon screen reader). Par ailleurs, la combinaison de deux modes de communication contribue à la réduction de la charge informationnelle supportée par chaque mode (Catinis, 1998).

Problématique

L'accès aux mathématiques, pour un utilisateur non-voyant, demeure une tâche complexe pour plusieurs raisons: premièrement, les expressions mathématiques exploitent souvent des structures bidimensionnelles (indice, exposant, matrices, bornes d'intégrales, etc.). Or, la conversion d'une structure multi-dimensionnelle en une structure linéaire (non visuelle) est un problème qui peut s'avérer ardu. À titre d'exemple, le braille linéaire utilise des informations supplémentaires pour dénoter certaines composantes de l'expression (ex. exposants). Deuxièmement, les 64 caractères (du braille standard¹) ne sont pas suffisants pour représenter tous les symboles fréquemment utilisés par les voyants pour écrire une expression mathématique. De plus, le grand nombre de symboles exige une mémoire et un effort cognitif important de la part de l'utilisateur non-voyant. Troisièmement, la manipulation de données pour l'utilisateur non-voyant n'est pas aussi simple qu'elle l'est pour un voyant. Par exemple, la navigation dans un document braille est lourde et comporte deux risques majeurs pour le lecteur : la désorientation et la surcharge cognitive (Raman, 1994).

De nombreuses applications ont déjà été développées pour offrir un accès aux mathématiques pour les non-voyants. Différentes formes ont été utilisées. Nous pouvons citer, à titre d'exemple, le braille linéaire qui traduit les expressions mathématiques écrites en LaTeX² ou MathML³ en braille. Les outils qui offrent cette forme de présentation sont nombreux :

¹ Braille : <http://6dotbraille.com> . Consulté le 4/4/2009.

² L. Lamport, LaTeX, 1994: The Macro Package, <http://web.mit.edu/texsrc/source/info/latex2e.pdf>. Consulté le 4/4/2009.

³ MathML: <http://www.w3.org/Math>. Consulté le 4/4/2009.

VICKIE (Visually Impaired Children Kit for Inclusive Education) (Moço et Archambault, 2003b) et BraMaNet (Braille Mathématique sur le Net) (Schwebel et Goiffon, 2005) sont deux exemples. Il existe aussi un autre système nommé DotsPLus (Preddy et al., 1997) qui permet d'embosser les symboles alphanumériques en braille tout en gardant la structure spatiale de l'expression mathématique telle qu'elle est écrite « en noire⁴ ». Plusieurs autres outils exploitent les sons pour présenter les expressions mathématiques à l'utilisateur non-voyant. Nous pouvons citer, à titre d'exemple, AudioMath⁵ (Ferreira et Freitas, 2005) et ASTER⁶ (Audio System for Technical Readings) (Raman, 1998) qui permettent de lire les expressions mathématiques codées en Latex ou MathML en utilisant plusieurs tonalités ainsi que des notes musicales.

Le problème de ces formes est qu'elles exigent des informations supplémentaires qui peuvent alourdir la présentation de l'expression. En outre, ces outils ne répondent pas aux besoins des utilisateurs (Garlini et Fogarolo, 2003; Podevin, 2002). Les résultats des études précédentes indiquent que les utilisateurs de ces outils ne sont ni indépendants, ni capables de réaliser leurs tâches sans l'aide d'un voyant. En effet, ces outils ont leurs propres limitations et ne peuvent pas s'ajuster selon le contexte de l'utilisateur.

À l'ère des systèmes intelligents, un des défis de la recherche, concernant l'accès aux mathématiques des non-voyants, est l'élaboration d'un système adaptatif qui exploite les avantages des techniques déjà mises en application, et répond aussi aux besoins de l'utilisateur.

⁴ Le terme « en noire » sera employé dans ce document pour désigner l'écriture manuscrite que les voyants utilisent.

⁵ AudioMath : http://lpf-esi.fe.up.pt/~audiomath/search_en.html. Consulté le 4/4/2009.

⁶ ASTER : <http://www.cs.cornell.edu/Info/People/raman/aster/demo.html>. Consulté le 4/4/2009.

Objectifs et démarche

L'objectif principal de ce travail est de proposer une approche pour présenter les expressions mathématiques et de la valider en modélisant les techniques utilisées. Cette approche doit permettre aux handicapés visuels un accès plus facile aux expressions mathématiques tout en garantissant un meilleur niveau d'autonomie.

Afin de traiter les problèmes et les limitations cités précédemment, nous proposons un système multi-agent (SMA) adaptatif qui interagit avec l'utilisateur. Le système a la capacité de sélectionner les modalités et les médias adéquats en se basant sur le contexte d'interaction (i.e. utilisateur, environnement et système). La forme de la présentation d'une expression mathématique est sélectionnée, parmi différentes formes possibles, en considérant les médias disponibles et les préférences de l'utilisateur ainsi que la complexité de l'expression.

Nous proposons une nouvelle approche qui répond aux carences des solutions traitées dans l'état de l'art de ce document:

1. Le système permet de présenter des expressions mathématiques en utilisant une ou plusieurs formes simultanément. En exploitant la multimodalité, il permet de combiner différentes formes afin : a) de présenter les informations pertinentes à l'utilisateur, b) de diminuer le plus possible la charge cognitive, et c) de rendre ainsi plus simple, aux personnes non-voyantes, l'accès aux formules mathématiques.
2. Le système proposé est un SMA, réparti en couches multiples. Le découpage du système en couches permet de modifier une couche sans induire un impact sur les autres couches. Chacune des couches requiert des services de la couche précédente, et en offre à la couche suivante. Cette architecture permet de créer une structure fondamentale et relativement simple pour ce SMA.

3. Le système est adaptatif. Il tient compte du contexte de l'interaction et il s'adapte aux changements affectant ce contexte. L'agent d'apprentissage intégré dans l'architecture veille à trouver la configuration optimale (les modalités, les médias et les formes de présentations appropriés au contexte de l'interaction). Cela fait en sorte que l'utilisateur a l'impression que le système a été développé spécialement pour lui.
4. Nous proposons une méthodologie originale pour calculer la complexité de l'expression mathématique. L'objectif visé est d'arriver à déterminer la forme de la présentation la plus appropriée, non seulement au contexte de l'interaction, mais également à la nature même de l'expression.
5. Le système proposé est tolérant aux fautes grâce à la reconfiguration dynamique de son architecture multimodale et ses agents intelligents. En effet, le système détecte les pannes et évalue la situation. Ensuite, il adapte ses comportements au nouveau contexte de l'interaction. Cela permet à l'utilisateur de continuer son activité sans intervention explicite de l'utilisateur.
6. L'architecture que nous proposons est évolutive. Nous pouvons donc ultérieurement ajouter de nouvelles formes et des médias pour enrichir les ressources du système et ensuite présenter l'expression mathématique à l'aide de la forme la plus convenable à l'utilisateur et selon son contexte.
7. Le système proposé est ubiquitaire. L'utilisateur aura donc la possibilité de travailler sur sa tâche, n'importe où et n'importe quand. En effet, notre architecture prend en considération le contexte du système qui implique la machine de l'utilisateur et les médias disponibles.
8. Nous avons examiné comment le système offre continuellement du service à l'utilisateur en tenant compte d'un contexte d'interaction inconstant qui peut affecter la stabilité du

système. Nous avons validé notre approche en utilisant des spécifications formelles (écrites dans le langage Z) et des simulations (via des réseaux de Pétri et JADE).

9. Enfin, notre architecture générique peut être adaptée pour répondre à d'autres types de déficiences.

Cette architecture a pour finalité un comportement intelligent parce qu'elle réalise deux propriétés fondamentales : l'accès universel et la mobilité (Djenidi, 2007). L'utilisateur bénéficie alors d'une certaine flexibilité dans l'utilisation du système.

Organisation de la thèse

Dans le premier chapitre, nous présenterons une revue de la littérature pour illustrer la contribution des travaux précédents à l'élaboration de notre thèse, ainsi que l'originalité de nos principales contributions. Les quatre chapitres suivants s'articuleront autour de quatre articles dont les deux premiers sont acceptés et les deux derniers ont fait l'objet de publications parues.

L'article du deuxième chapitre est accepté pour publication dans la revue International Society for Advanced Science and Technology (ISAST) Transactions on Intelligent Systems :

A. Awdé, C. Tadj, Y. Bellik, "Infrastructure of an Adaptive Multi-agent System for Presentation of Mathematical Expression to Visually Impaired Users", International Society for Advanced Science and Technology (ISAST) Transactions on Intelligent Systems, To be published in 2009.

Dans cet article, nous proposons un système multi-agent permettant de présenter des expressions mathématiques à des utilisateurs non-voyants. Nous présentons une architecture multicouche d'un système adaptatif. Ce système est tolérant aux fautes grâce à la

reconfiguration dynamique de son architecture et ses agents. Nous illustrons les différents agents de chaque couche et leurs principaux rôles.

L'article du troisième chapitre est accepté pour publication dans la revue *Journal of Multimedia (JMM)*:

A. Awdé, M. D. Hina, C. Tadj, Y. Bellik, A. R. Cherif, "An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually-Impaired Users", *Journal of Multimedia (JMM)*, Vol. 4, No. 4, 2009.

Cet article a pour thème l'adaptation des formes de présentation d'une expression mathématique au sein d'un système multimodal sensible au contexte d'interaction (i.e. utilisateur, environnement et système). En se basant sur le contexte d'interaction, un algorithme d'apprentissage détermine les modalités possibles et les médias supportant celles-ci. Les formes de présentation d'une expression mathématique sont sélectionnées en prenant en considération les médias disponibles, les préférences de l'utilisateur et la nature de l'expression mathématique.

L'article du quatrième chapitre a été publié dans la revue *International Journal of Computer and Information Science and Engineering (IJCISE)*:

A. Awdé, Y. Bellik, C. Tadj, "Complexity of Mathematical Expressions in Adaptive Multimodal Multimedia System Ensuring Access to Mathematics for Visually Impaired Users", *International Journal of Computer and Information Science and Engineering (IJCISE)*, Vol. 2, No. 2, 2008.

Dans cet article, nous développons une méthode originale de calcul de la complexité d'une expression mathématique en se basant sur les paramètres de l'expression (les opérandes, opérateurs et la profondeur de son arbre syntaxique) et aussi en tenant compte du profil du lecteur.

L'article du cinquième chapitre a été publié dans la revue *Ubiquitous Computing and Communication Journal (UBICC)*:

A. Awdé, M. D. Hina, C. Tadj, Y. Bellik, A. R. Cherif, "Middleware for Ubiquitous Access to Mathematical Expressions for Visually-Impaired Users", *Ubiquitous Computing and Communication Journal (UBICC)*, Vol. 3, No. 5, 2008.

Dans ce dernier article nous présentons l'architecture d'un intergiciel qui permet à un utilisateur non-voyant d'accéder aux mathématiques n'importe où, n'importe quand. L'objectif n'étant pas seulement l'accès aux mathématiques par les non-voyants, mais également de disposer d'informations ubiquitaires afin d'assurer une accessibilité dynamique sans une intervention humaine. Quand un utilisateur se déplace d'un environnement à un autre, le contexte et les ressources changent. Suite à ces changements, un système multimodal, sensible au contexte, doit s'adapter à la nouvelle situation en activant ou désactivant les médias ou les modalités qui sont appropriés à la nouvelle situation. Ainsi, des problèmes mettant en jeu des reconfigurations de l'architecture plus complexes que ceux du chapitre précédent sont traités ici.

Enfin, le sixième chapitre sera consacré à la conclusion de notre travail de recherche. Il dresse le bilan et donne des perspectives autour de l'accès aux expressions mathématiques et l'adaptativité dans les systèmes multimodaux.

CHAPITRE 1

ÉTAT DE L'ART SUR LA PRÉSENTATION DES EXPRESSIONS MATHÉMATIQUES À DES UTILISATEURS NON-VOYANTS

Dans ce chapitre, nous présentons les méthodes d'accès, à l'information numérique, par des utilisateurs non-voyants et en particulier l'accès aux expressions mathématiques. Tout d'abord, nous définissons certaines terminologies et concepts utilisés dans ce travail. Nous présentons les médias et les modalités exploitables par les non-voyants. Ensuite, un ensemble, non exhaustif de systèmes proposés dans la littérature pour présenter les expressions mathématiques aux non-voyants, est décrit.

1.1 Définitions

Dans le cadre de notre travail, nous utilisons des notions et des termes dont les définitions sont multiples dans la littérature. Nous pouvons citer les travaux suivants: (Frohlich, 1991), (Bernsen, 1993), (Nigay et Coutaz., 1995) et (Bellik, 1995) qui ont défini l'interaction et ses composants selon le point de vue de l'utilisateur ou de la machine. Pour notre part, nous privilégions les notions de Bellik (i.e. telles qu'elles sont perçues par l'utilisateur).

1.1.1 L'interaction homme-machine

L'interaction homme-machine étudie la manière dont l'utilisateur interagit avec la machine (Bellik, 1995). Les composants d'interaction sont généralement perçus comme les moyens de communication (logiques ou physiques) entre l'utilisateur et la machine (Rousseau et al., 2006). Dans la littérature, trois types de composants d'interaction sont identifiés: le mode, la modalité et le média.

1.1.2 Le mode

Dans (Bellik, 1995), la notion de « mode » réfère à l'organe humain utilisé pour la production et la perception d'information. Chez l'être humain, les modes de communication d'entrée (de la machine vers l'homme) correspondent aux cinq sens de perception (le visuel, l'auditif, l'olfactif, le gustatif et le tactilo-proprio-kinesthésique (TPK)). Alors que les modes de sortie correspondent (de l'homme vers la machine) aux moyens d'expression suivants : modes gestuel et oral. Seuls les modes Auditifs et TPK sont employés dans les systèmes destinés aux utilisateurs non-voyants. Le mode visuel peut être employé par un assistant voyant mais il est non exploitable par les non-voyants. Cependant, il existe deux modes, gustatif et olfactif, qui ne sont pas utilisés dans une interaction multimodale.

1.1.3 La modalité

La modalité fait référence à la structure des informations telle qu'elle est perçue par l'utilisateur (texte, sonnerie, parole, braille, etc.). Un mode peut comporter une ou plusieurs modalités. Par exemple, le mode auditif comporte plusieurs modalités comme la parole, la sonnerie, le bruit, l'earcon, etc.

1.1.4 Le média

Le média est défini comme un dispositif physique ou une entité logique apte à acquérir ou à livrer l'information (par exemple: l'écran, le terminal braille, la souris, le clavier, etc.). Le média permet de supporter une ou plusieurs modalités. À titre d'exemple, le terminal braille permet l'expression de la modalité braille. Les hauts parleurs supportent la sonnerie, la parole, le bruit, l'earcon, etc.

1.1.5 La forme de la présentation

La détermination de la modalité et des médias qui la supportent ne suffit pas pour, définir de façon précise, la forme de la présentation de l'information (Jacquet, 2006). Par exemple, en

choisissant la parole comme modalité, il faut choisir le genre (masculin ou féminin), l'âge (jeune, vieux), le volume (bas, moyen, haut) afin de bien présenter l'information. Ces paramètres sont appelés les attributs de la modalité. Dans (Rousseau, 2006) les composants de l'interaction (modalité, média) déterminent la forme de la présentation, tandis que leurs attributs fixent l'instance de la présentation.

1.1.6 Le contexte de l'interaction

Schilit (Schilit, Adams et Want, 1994) a défini le contexte par le lieu d'utilisation, l'identité de l'utilisateur et les objets dans son environnement. Certains chercheurs (Dey, 2001; Schmidt et al., 1999; Souchon, Limbourg et Vanderdonckt, 2002) ont pris cette définition et ont tenté de définir formellement le contexte. D'autres, comme Chen & Kotz (Chen et Kotz, 2000) ont ajouté au contexte la situation temporelle, par exemple la date, l'heure, etc.

En ce qui nous concerne, nous adaptons la définition de Dey qui considère le contexte comme toute information utilisée pour caractériser la situation d'une entité. Dans sa définition, l'entité est perçue comme étant une personne, un lieu ou un objet pertinent pour l'interaction entre l'utilisateur et l'application.

Formellement, nous définissons le contexte d'interaction comme un triplet qui se compose du: contexte d'utilisateur (UC), contexte d'environnement (EC) et du contexte du système (SC) (Awde et al., 2009).

1.2 Les médias et les méthodes d'accès aux informations utilisés par les non-voyants

Nous distinguons deux sous-ensembles de médias : les médias qui exploitent le mode TPK et ceux qui permettent d'exploiter le mode auditif.

1.2.1 Le braille

Tout d'abord, le braille n'est pas un média mais une modalité. Avant l'existence du braille (1829), les non-voyants n'utilisaient qu'un seul moyen de communication, la parole, faisant ainsi appel au sens de l'ouïe. Il leur était donc impossible d'accéder aux documents écrits sans l'aide d'une tierce personne. Ainsi, avec l'avancé technique, que constitue le braille, l'accès à la lecture est devenu possible en exploitant le sens du toucher (i.e. le mode TPK). Cependant, la transcription du braille reste volumineuse d'une part, et difficile à manipuler d'autre part.

Malgré ses inconvénients, le braille permet aux non-voyants d'échanger de l'information textuelle avec les personnes qui maîtrisent bien la lecture de cette écriture (malheureusement trop peu de gens maîtrisent le braille). Par contre, avec l'ordinateur, il n'est pas nécessaire d'apprendre le braille en vue de transcrire un document, étant donné l'existence de plusieurs outils logiciel accomplissant déjà cette tâche. Nous pouvons citer NAT⁷ qui est un logiciel libre de transcription automatique capable de convertir un fichier texte (plusieurs formats sont pris en compte : .txt, .doc, etc.) en braille.

Enfin, en braille standard, un caractère est représenté par une matrice de six points ou picots sur deux colonnes. Selon la position des picots, levée ou baissée, il est possible de disposer de 64 caractères différents. Cela ne suffit pas pour coder tous les caractères et symboles utilisés par les voyants. Cependant, le braille informatique utilise huit points pour représenter un caractère (voir Figure 1.1).

La Figure 1.2 illustre le mot « braille » écrit en braille français. Un point noir est l'équivalent d'un picot levé, un petit point représente un picot baissé.

⁷ <http://natbraille.free.fr/>. Consulté le 4/4/2009

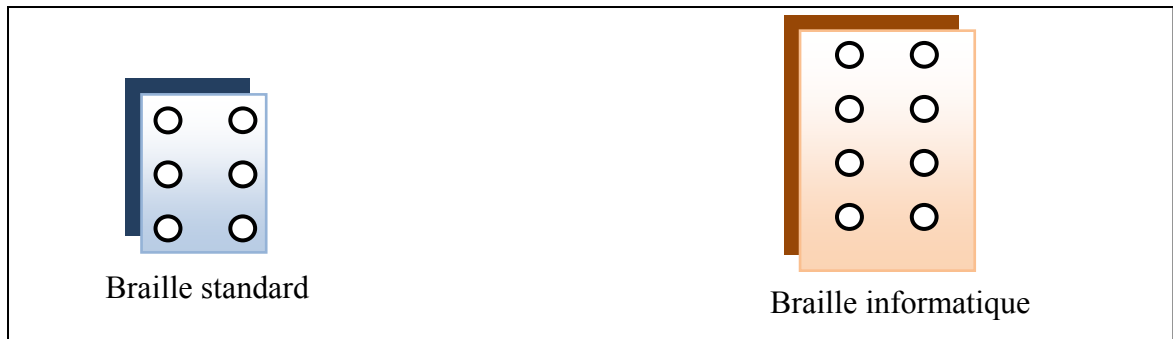


Figure 1.1 Structure d'un caractère braille standard et informatique.

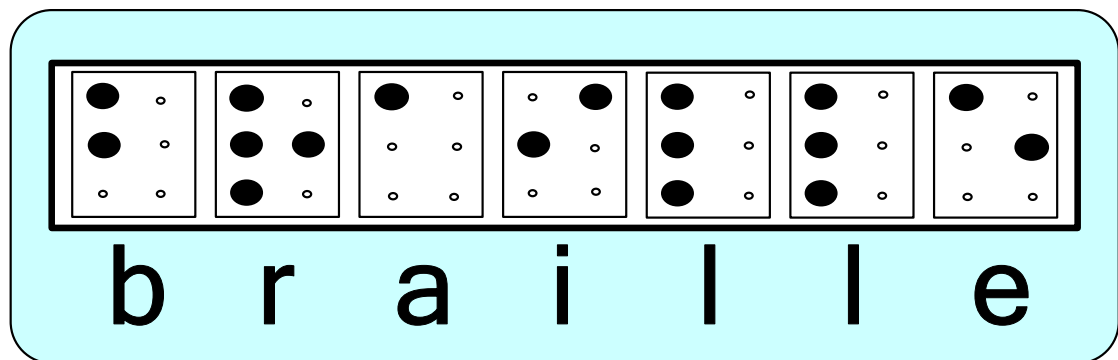


Figure 1.2 Le mot « braille » et son équivalent en braille.

Il existe plusieurs types de matériels pour présenter les données en braille. Le papier est le média le plus utilisé et demeure le plus connu depuis l'invention du braille (voir l'exemple de la Figure 1.3). Avec la technologie de l'information, plusieurs concepts ont été intégrés, telle que la dynamique du média utilisé. Dans cette perspective, nous présentons quelques médias utilisés pour l'affichage du braille.

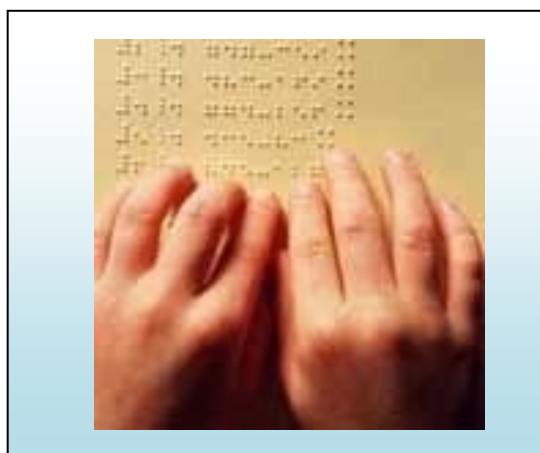


Figure 1.3 Un document écrit en braille.

1.2.2 Les médias supportant le mode TPK

Terminal Braille

Un terminal braille possède en général des cellules brailles composées de huit points (voir l'exemple de la Figure 1.4). Ainsi, il peut présenter les équivalents de 256 caractères ASCII. Il permet également l'affichage d'une ligne de 20, 40 ou 80 caractères. Les points d'un caractère braille sont représentés par des picots. Plusieurs terminaux, comme ClioBraille, disposent d'un clavier braille de huit touches et ils intègrent également des boutons de désignation, des touches de fonctions et un clavier numérique. Un tel dispositif bénéficie d'une meilleure ergonomie. L'affichage braille est la méthode classique d'accès aux informations (sur papier ou numériques) pour les aveugles. Cependant une bonne maîtrise de cette méthode exige souvent plusieurs années de pratique. En outre, la lecture du braille est assez lente, ce qui implique un temps supplémentaire pour mieux lire et comprendre un texte.



Figure 1.4 Le terminal braille VARIO 80.

Dot Matrix Display

Il s'agit d'une matrice de picots (voir Figure 1.5). Cette plage est composée de quelques milles picots (il existe plusieurs modèles et de différentes dimensions). Par exemple : la matrice « DMD 120060) de Metec⁸ dispose de 7200 picots soit de 120 (L)* 60 (H) picots. Ce matériel est bien intéressant parce qu'il propose une surface vaste, de haute résolution, pour afficher les données. En effet, les terminaux braille classiques peuvent afficher une seule ligne de 80 caractères en même temps. Cependant, avec 7200 picots, on pourrait éditer douze lignes, de quarante caractères!

Ce périphérique « rafraîchissable » de haute résolution a été utilisé pour afficher des formes tactiles telles que le braille, les courbes, et les cartes géographiques (Jürgensen et Power, 2005; Roberts et Ertl, 2004; Vani et al., 2005). Par exemple, la Figure 1.5 tirée de (Roberts et Ertl, 2004) affiche des données graphiques.

Le développement d'afficheur braille de très bonne « résolution » serait la meilleure manière de présenter à la fois du texte et des informations d'autre nature. Mais malheureusement, ces outils sont encore loin d'être à un prix abordable.

⁸ Metec est une compagnie allemande, qui fabrique ce genre de matériel. <http://www.metec-ag.de/english.html>. Consulté le 4/4/2009.

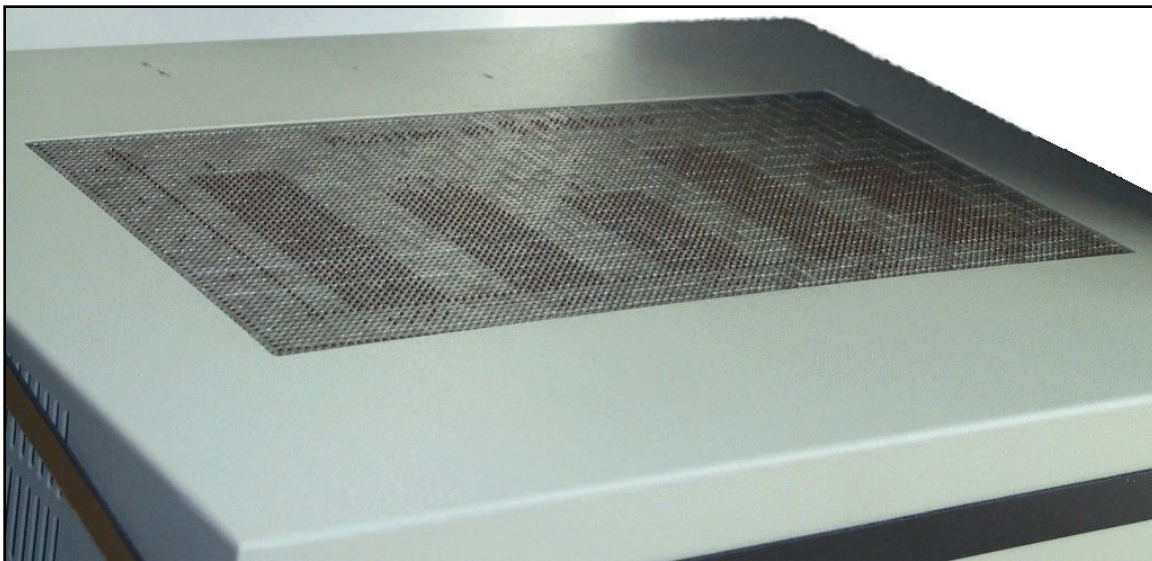


Figure 1.5 La matrice qui permet d'afficher un graphique en tactile.

La mise en relief (Tiger Printer)

Il existe d'autres périphériques qui peuvent présenter les informations sous forme tactile en les transcrivant sur papier ou sur des supports similaires. Parmi ces outils, l'un des plus prometteurs est l'imprimante du TIGER (The TactIle Graphics EmbosseR, voir Figure 1.6) (Walsh et Gardner, 2001).



Figure 1.6 TIGER printer : imprimante de « fonte » tactile.

Ces imprimantes ont une haute résolution (20 D.P.I.). Elles sont capables d'imprimer en relief quasiment tous les formats de données électroniques tels que le texte, les graphiques, etc. provenant des applications informatiques standard. Elles offrent une facilité d'utilisation et des fonctionnalités inégalées parce qu'elles peuvent imprimer les dossiers qui contiennent le texte et les graphiques directement des applications, desquelles on imprimerait normalement un document.

Ce genre d'imprimante a été utilisé dans plusieurs projets (Gardner, 1993; 1998; Preddy et al., 1997) qui combinent des symboles d'impression en relief et les caractères en braille. Enfin, ce type d'accès tactile offre un accès un peu plus rapide à l'information et à sa présentation spatiale. Prenons comme exemple concret une expression mathématique. Grâce au toucher, le lecteur peut facilement distinguer quelques symboles tels que la barre de fraction, une ligne verticale, etc. Cependant, l'inconvénient majeur de ce type de périphériques est qu'il nécessite une sortie sur papier du document en question. Il s'agit donc d'un média statique.

Il existe également des imprimantes brailles qui permettent d'embosser uniquement les écritures en braille, mais qui restent inutile pour imprimer les formes en relief comme l'imprimante TIGER.

La tablette tactile « Overlay Keyboard »

C'est une tablette tactile (voir Figure 1.7) utilisée pour commander l'ordinateur sans l'obligation de passer via le clavier habituel. Ce média fonctionne comme suit : le matériel est connecté à un ordinateur via le port série (ou USB). On dispose sur la tablette une feuille de papier (overlay), plastique ou une feuille embossée en braille dans la zone tactile. Quand on appuie sur le support posé sur la tablette, une certaine action est déclenchée sur la machine. Cette action peut être destinée à afficher du texte, montrer une image, jouer de la musique, ou lancer toute autre application.



Figure 1.7 Un exemple d'un « Overlay Keyboard » développé par APH⁹.

Le rôle de cet outil consiste principalement à faciliter l'interaction entre la machine et les utilisateurs ayant des déficiences motrices ou visuelles. Ce média a joué un rôle important dans plusieurs projets (Podevin, 2002; Sheehan, 2005).

Le matériel se présente comme une matrice (par exemple une matrice de 16 * 16 cases). Sur la feuille disposée au-dessus, appelée « overlay », nous pouvons embosser du braille ou des formes en relief. Chaque objet (un caractère braille, une forme en relief) représenté coïncide en dessous avec une ou plusieurs cases des 256 déjà existantes. Quand l'utilisateur clique sur l'objet, si effectivement cet objet est rattaché à une action, alors un message est envoyé à l'ordinateur et ce dernier effectue l'action correspondante. On peut imaginer un scénario où chaque case est un mot. Dans ce cas, quand l'utilisateur appuie dessus, il entend le mot correspondant. De ce fait, l'utilisateur non-voyant peut avoir un autre moyen pour comprendre les symboles qu'il vient de toucher, exerçant ainsi sa coordination audio-motrice. Nous pouvons ici parler de redondance des modalités (tactile et auditive).

L'avantage d'un tel dispositif est son prix (il n'est pas cher comparativement au DMD). Il est léger et portable. La seule manipulation que l'utilisateur doit faire, c'est l'embossage des feuilles disposées dessus. Par contre, l'inconvénient majeur est que le document est statique.

⁹American Printing House for the Blind (APH) est une compagnie spécialisée dans les matériels destinés aux non-voyants: <http://sun1.aph.org/index.htm>. Consulté le 4/4/2009.

En effet, une fois le document imprimé, il est non modifiable. Il est juste possible d'interagir afin d'avoir un retour. L'édition d'un overlay reste cependant une tâche difficile pour le type d'utilisateurs visés.

Le clavier

Bien sûr, si l'utilisateur maîtrise bien les touches du clavier (le clavier standard, comportant environ 100 touches), il peut saisir rapidement du texte et tout type de caractères. De plus, on peut dans l'interface utilisateur créer des raccourcis clavier qui simplifient l'accès à certaines fonctions (ils doivent être peu nombreux, faciles à retenir et configurables). Le clavier comme tel n'est pas relié directement à une méthode d'affichage, mais le texte saisi peut-être transcrit ultérieurement en braille à l'aide d'outils spécifiques ou bien être synthétisé en utilisant un système de synthèse vocale.

Le retour d'effort

Les interfaces à retour d'effort permettent de matérialiser les objets présents dans le monde virtuel. Elles appliquent, sur la partie du corps en contact avec l'objet virtuel, la force réciproque de la force qu'exercerait l'utilisateur sur l'objet réel. Le retour d'effort et l'effet de contact peuvent être normalement produits par trois sortes d'énergies : l'énergie électrique (système vibratoire), l'énergie mécanique (les poids d'un tel dispositif commencent à contraindre l'utilisateur) et l'énergie pneumatique (tels que des poches d'air incluses dans un gant).

La Figure 1.8 montre un exemple de type d'interfaces. Il s'agit d'un gant à retour d'effort (Cyberglove) fabriqué par VRLOGIC¹⁰. Ce média ne peut pas être trop utile pour les formules mathématiques, mais il peut être utilisé pour faire comprendre aux non-voyants les formes géométriques. Aussi, il permet aux non-voyants de se repérer dans l'espace, et d'étudier les graphes et les courbes. Avec les interfaces à retour d'effort, l'utilisateur a

¹⁰ VRLOGIC est une compagnie allemande : <http://www.vrlogic.com/>. Consulté le 4/4/2009.

l'impression de rester dans les limites de son espace de travail. Des exploitations de ce genre d'interfaces ont déjà été réalisées auparavant (Nemec, Sporka et Slavik, 2004; Sjöström, 2001).



Figure 1.8 Le gant à retour d'effort Cyberglove II.

Néanmoins, ces médias, permettant de sentir les formes et travailler en multi-dimensions, n'ont pas été exploités pour améliorer l'accès aux expressions mathématiques.

1.2.3 La parole

La parole en sortie et la synthèse vocale

La lecture des informations est la méthode la plus anciennement utilisée par les personnes non-voyantes pour accéder aux informations. De ce fait, pour connaître le contenu d'un document, on demande à une personne voyante de lire à haute voix les informations écrites afin de les enregistrer sur différents supports (cassette audio, CD, etc.) pour que l'utilisateur non-voyant puisse les réécouter plus tard quand il le désire.

Cependant, les révolutions technologiques, et en particulier la synthèse vocale ont accru l'utilisation de la parole. En effet, la technique de synthèse de la parole est la plus utilisée par les non-voyants pour accéder aux ordinateurs. Ceci s'explique par deux raisons :

- Seule une minorité des non-voyants lit effectivement le braille (Bellik, 1995) (moins de 10%).

- Le coût d'un système de synthèse vocale est beaucoup plus faible que celui d'un terminal braille.

Plutôt que d'envoyer le texte correspondant à la ligne courante, sur l'afficheur braille, le programme d'accès l'enverra cette fois-ci à un système de synthèse vocale et le fera sortir via les haut-parleurs ou les écouteurs. Cependant, durant la lecture, l'utilisateur doit être très attentif s'il veut mémoriser le texte lu.

Cette technologie a été utilisée pour créer des interfaces vocales destinées aux utilisateurs non-voyants (Xydas et al., 2005), tels que les lecteurs d'écran (en anglais screen reader). Ce type d'application est très connu parmi les logiciels destinés aux déficients visuels. Il transmet ce qui est affiché sur l'écran à une synthèse vocale. Il existe déjà plusieurs lecteurs d'écran fonctionnels, comme screader¹¹ et Speakup¹².

La parole en entrée et la reconnaissance vocale

La parole est la modalité de communication la plus naturelle pour l'homme. Il suffit de relier le microphone à un système de reconnaissance vocale pour que la machine reconnaisse des éléments lexicaux. Mais malheureusement les logiciels de reconnaissance vocale ne sont pas toujours robustes. De plus, le système implique un apprentissage de la voix du locuteur. Le nombre d'erreurs peut être important pour la saisie d'une formule ou même d'un simple texte.

Par contre, elle est largement utilisée pour désigner des commandes vocales. En effet, le remplacement des raccourcis claviers, par des mots, facilite la mémorisation des commandes, car les mots, contrairement aux raccourcis claviers, possèdent un contenu sémantique. La reconnaissance vocale permet à l'utilisateur d'accéder rapidement à des commandes et ensuite aux informations: elle peut par exemple lui offrir une navigation plus facile au sein

¹¹ Screader est disponible sur : <http://web.inter.nl.net/users/lemmensj/homepage/uk/screader.html>. Consulté le 4/4/2009.

¹² Speakup est disponible sur : <http://www.linux-speakup.org/speakup.html>. Consulté le 4/4/2009.

d'un document. Parler n'occupe ni les mains, ni « les yeux, pour un utilisateur voyant »! Par conséquent, il est possible grâce à la parole de communiquer et d'agir en même temps. Par exemple, l'utilisateur peut dicter un texte qu'il est en train de lire du bout de ses doigts (braille papier) sans que ses mains aient besoin de se déplacer d'un média à un autre (i.e. d'un papier à un clavier).

1.2.4 Les sons non parlés, les bips sonores, les earcons

Les earcons sont des sons obtenus par l'utilisation de fréquences musicales organisées selon une grammaire plus ou moins complexe faisant intervenir des changements de hauteur et de rythmes (Blattner, Sumikawa et Greenberg, 1989; Brewster, 1998; Mynatt, 1994). On utilise normalement ces earcons pour fournir de l'information à l'utilisateur concernant une action, un objet manipulé, des données graphiques, ou bien des expressions mathématiques. Les earcons ont été employés dans plusieurs projets qui s'adressent aux utilisateurs non-voyants (McGookin et Brewster, 2006; Stevens, Edwards et Harling, 1997). Des études précédentes ont aussi essayées d'exploiter la combinaison de la synthèse vocale, des earcons et des sons 3D (Djennane, 2003; Gorny, 2000; Kramer, 1994; Xydas, Spiliotopoulos et Kouroupetroglou, 2003).

Cette méthode est quelque peu difficile à apprendre. L'utilisateur doit avoir une oreille très attentive et musicale. Dans le cas des formules mathématiques, il doit apprendre à reconnaître qu'un tel son représente un tel opérateur, que telle note signifie que c'est un indice ou un exposant, etc. Les earcons sont en général utilisés que pour donner un aperçu globale de la formule.

1.3 Les besoins des utilisateurs

La réalisation d'un logiciel ou d'un système informatique doit généralement être centrée sur l'utilisateur. Elle doit répondre à ses besoins, avoir une bonne connaissance de ses attentes et prendre en considération ses critiques. Cependant, l'évaluation des besoins des utilisateurs n'est pas une phase facile. En effet, celle-ci dépend directement du questionnaire, et de la

méthode suivie pour récolter les informations et les interpréter. Dans notre cas, plusieurs variables sont prises en compte pour établir des questions homogènes, classifiées, couvrant la totalité du sujet et tous les cas possibles. Les réponses à ces questions doivent être récoltées dans un cadre statistique qui facilite l'interprétation des résultats. Ainsi, les questionnaires doivent être bien identifiés et bien clairs afin d'éviter les questions non pertinentes. Mis à part, les questions posées durant l'évaluation des besoins des utilisateurs, nous tenons compte de l'utilité de l'observation de l'utilisateur dans son milieu de travail ou bien à l'école (la façon dont il réagit devant un système informatique, un document braille, une équation mathématique, etc.).

Dans cette section, nous présentons et analysons les résultats obtenus dans des études précédentes (Archambault et Burger, 2002; Casson, 2005; Garlini et Fogarolo, 2003; Podevin, 2002). Dans ces études, les observations des étudiants et personnes non-voyantes se concentraient sur les habitudes, les difficultés et la manière dont ils agissaient pour accéder aux données.

Dans Lambda¹³ (*Linear Access to Mathematic for Braille Device and Audio-synthesis*) l'évaluation a été menée dans six pays : Italie, Royaume Unis, France, Espagne, Grèce et Russie. Les questionnaires ont été adressés aux étudiants aveugles (niveau universitaire dans la plupart des cas) et leurs enseignants (Garlini et Fogarolo, 2003). Au sein du projet MICOLE¹⁴ (*Multimodal Collaboration Environment for inclusion of Visually Impaired Children*), les évaluations des besoins sont réalisées par le biais d'entretiens et d'observations des élèves ayant des déficiences visuelles et de leurs professeurs de mathématiques (Casson, 2005). Cependant dans VICKIE¹⁵ (*Visually Impaired Children Kit for Inclusive Education*), les systèmes existants ont été tout d'abord analysés, ensuite trois types d'utilisateurs ont été observés : les étudiants non-voyants qui étaient dans leurs classes utilisaient leurs matériels spécialisés, les professeurs qui préparaient les documents adaptés aux non-voyants ainsi que les utilisateurs professionnels qui préparent les matériels éducatifs pour les déficients visuels.

¹³ Le projet Lambda : <http://www.lambdaproject.org/>. Consulté le 4/4/2009.

¹⁴ Le projet MICOLE : <http://micole.cs.uta.fi/>. Consulté le 4/4/2009.

¹⁵ Le projet VICKIE : <http://inova.snv.jussieu.fr/vickie>. Consulté le 4/4/2009.

L'objectif principal de ces études était de collecter des informations précises sur les utilisateurs, leurs environnements, les expériences des enseignants ainsi que les besoins des étudiants selon eux-mêmes et selon les évaluateurs qui les observaient en train de manipuler les expressions mathématiques. Les observations et les tests ont été réalisés sur la base de deux hypothèses principales:

- chez les étudiants déficients visuels, étudier les mathématiques est plus difficile que chez les voyants, mais cela n'est pas impossible;
- les avantages des nouvelles technologies peuvent amener l'étudiant non-voyant à une autonomie presque complète même si les résultats actuels sont trop modestes.

Les participants confirmaient que le calcul mental n'est pas un problème pour les opérations simples ni pour les opérations qui ne demandent pas de précisions. Dans tous les cas, les non-voyants ne considèrent pas le calcul comme un problème majeur. D'ailleurs, les observateurs ont noté que la calculatrice parlante était l'outil le plus utilisé pour aider les aveugles à effectuer leurs calculs. Cependant, le braille était le plus populaire et le plus adapté pour les aveugles, mais il n'était pas adopté partout : la plupart des Italiens et des Français questionnés utilisaient principalement l'ordinateur (synthèse vocale et clavier traditionnel). Pour l'écriture, plus que la moitié des étudiants utilisent des outils électroniques pour rédiger des documents mathématiques.

L'analyse des données récoltées a relevé un problème sérieux : la plupart des répondants ne connaissait pas le braille à 8 picots (braille informatique). La majorité des étudiants avait l'opportunité d'utiliser l'ordinateur pour faire des travaux scolaires. Mais un nombre minimal d'étudiants utilisait ce moyen pour accéder aux mathématiques à cause des limitations des méthodes développées. En effet, juste 7% des utilisateurs questionnés étaient satisfaits du logiciel utilisé pour accéder aux mathématiques.

Les utilisateurs ont spécifié des caractéristiques majeures pour ce genre de logiciel à considérer lors de la conception d'un futur projet. Les besoins des utilisateurs et les caractéristiques du système peuvent être résumés par :

- la simplicité (le système utilise seulement des commandes adaptées aux utilisateurs non-voyants et faciles à mémoriser);
- le système n'exige pas une longue phase d'apprentissage (un système simple n'exige pas une phase d'apprentissage compliquée);
- le logiciel doit permettre la lecture et l'écriture facilement et rapidement (les utilisateurs désirent un logiciel permettant de lire et éditer les expressions mathématiques et surtout un logiciel permettant la réduction du temps pour comprendre ou manipuler une expression mathématique);
- la compatibilité (le système doit accepter les formats des fichiers manipulés par les voyants tels que MathML, LaTeX, etc.);
- l'utilisation du braille et de la parole (la plupart des participants aux études sont des étudiants qui fréquentaient des écoles spécialisées pour aveugles et qui connaissaient le braille);
- l'utilisation des messages « feedback » (pour corriger et diriger l'utilisateur durant la réalisation de sa tâche);
- la navigation au sein d'un document mathématique et de l'expression elle-même (l'utilisateur a besoin de manipuler l'expression et de pouvoir accéder aux termes voulus facilement);
- la conversion automatique des expressions mathématiques entre les différentes formes de présentations (les étudiants et les professeurs ont manifesté un intérêt pour les outils qui leur permettent de s'échanger les données sans difficulté);
- le système doit accepter les nouvelles technologies (médiâs, techniques, etc.) pour développer des mécanismes intelligents adaptés aux non-voyants ainsi que réduire la complexité de l'interaction de l'utilisateur avec la machine;
- la nécessité des médiâs spéciaux pour la lecture et l'écriture (c.-à-d. des imprimantes TIGER, des médiâs d'affichage rafraîchissable, etc.);

- le système doit réduire le temps de la manipulation des expressions mathématiques et simplifier la tâche de l'utilisateur;
- et finalement, le logiciel doit s'adapter aux besoins de l'utilisateur (il se configure selon les préférences et le contexte de l'utilisateur) pour éviter à l'utilisateur de s'adapter à un outil non conçu pour lui (actuellement, c'est lui qui s'adapte aux outils).

L'analyse de ces études nous a permis de mettre en évidence les attentes et les besoins des utilisateurs visuellement déficients et, de ce fait, les fonctionnalités qui leur seraient utiles pour un futur logiciel permettant un accès aux mathématiques.

Après avoir exploré les besoins des utilisateurs, nous présentons les approches utilisées pour présenter les expressions mathématiques aux non-voyants. Nous décrivons, également, les avantages et les inconvénients de chaque approche et le contexte de leur utilisation.

1.4 Les approches pour l'accès aux mathématiques

Les recherches tentent de diminuer, autant que possible, la charge et les informations supplémentaires des formules mathématiques, en rendant leur accès aussi simple pour les voyants. Les formes visuelles que la plupart des voyants et « non-voyants » utilisent pour éditer les expressions mathématiques sont MathML et LaTeX. MathType^{TM16} de Microsoft est une autre forme de présentation visuelle qui s'intègre dans les outils de traitement de texte. Dans ce cadre, plusieurs projets ont été réalisés pour présenter les expressions mathématiques aux non-voyants sous différentes formes. Nous les classifions en deux catégories : les approches braille/tactiles et les approches sonores. Nous abordons ces deux catégories après avoir présenté les formes visuelles.

¹⁶ MathType : <http://www.dessci.com/en/products/mathtype/>. Consulté le 4/4/2009.

1.4.1 Les formes visuelles

MathType

MathType est un éditeur d'expressions mathématiques qui fonctionne sous Microsoft Windows ou Apple MacOS. Il s'intègre dans les applications (exemple MS-Word) et il permet d'insérer des expressions à l'aide de techniques simples dans des documents de traitement de texte, des pages web, etc. Pour créer une expression mathématique, il ne propose pas moins de 175 modèles contenant des symboles (par exemple des intégrales, des matrices, etc.), et diverses zones vides. L'utilisateur crée une expression en insérant le modèle désiré et en remplissant ses zones vides. La Figure 1.9 montre la structure du modèle qui permet d'écrire une fraction.

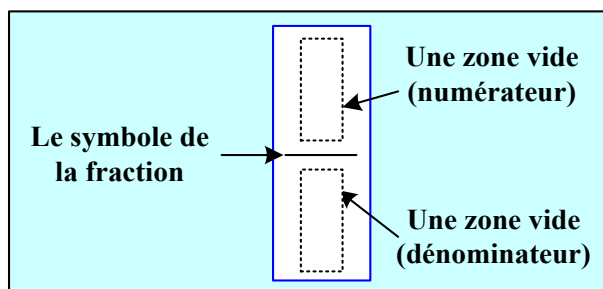


Figure 1.9 La structure d'un modèle dans MathType.

L'interface de MathType est intuitive et visuelle (voir Figure 1.10).

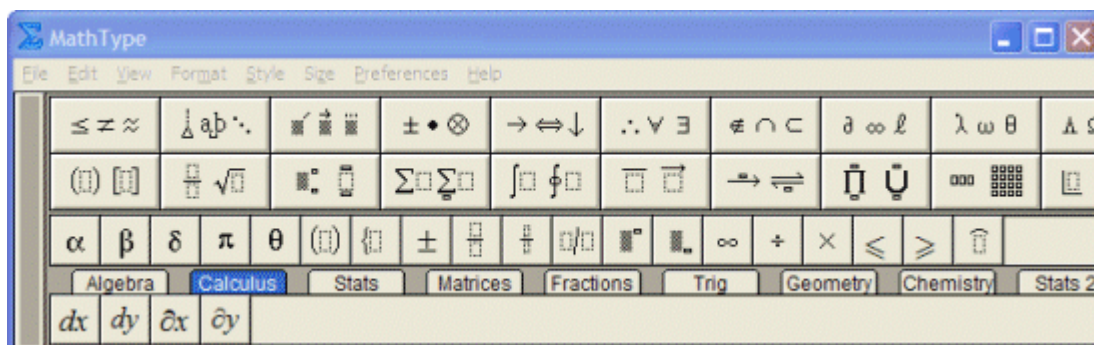


Figure 1.10 L'interface de MathType.

MathML

MathML¹⁷ est un langage basé en XML permettant d'afficher et d'assurer le traitement des expressions et des symboles mathématiques sur l'Internet (Caprotti et Carlisle, 1999). Il est devenu une recommandation du W3C¹⁸. Il permet d'inclure des expressions mathématiques de manière simple. Il présente les symboles et les notations mathématiques complexes sous une forme lisible qui facilite le travail des navigateurs Internet. Enfin, MathML résout les problèmes posés par l'utilisation d'images comme, par exemple, le fait de ne plus pouvoir éditer une expression une fois convertie sous forme d'image.

MathML traite non seulement la présentation mais aussi le sens des composantes des expressions mathématiques. Il propose un format adapté à la présentation et un autre au contenu des expressions mathématiques. Nous distinguons deux types de balises : les balises de la présentation (presentation markup) et les balises du contenu (content markup).

La Figure 1.11 montre les lignes du code MathML (présentation et contenu) qui permettent d'établir la fraction suivante :

$$\frac{x-1}{x+1} \quad (1.1)$$

Même si MathML n'est pas destiné aux non-voyants, il est utilisé dans plusieurs applications pour préparer des documents mathématiques. Ces applications seront présentées ultérieurement.

¹⁷ MathML : <http://www.w3.org/Math/>. Consulté le 4/4/2009.

¹⁸ World Wide Web Consortium : <http://www.w3.org/>. Consulté le 4/4/2009.

<u>Présentation</u>	<u>Contenu</u>
<pre> <math> <mrow> <mfrac> <mrow> <mi>x</mi> <mo>-</mo> <mn>1</mn> </mrow> <mrow> <mi>x</mi> <mo>+</mo> <mn>1</mn> </mrow> </mfrac> </mrow> </math> </pre>	<pre> <apply> <divide/> <apply> <minus/> <ci>x</ci> <cn>1</cn> </apply> <apply> <plus/> <ci>x</ci> <cn>1</cn> </apply> </apply> </pre>

Figure 1.11 Une fraction écrite en MathML.

LaTeX

LaTeX¹⁹ est un langage permettant de rédiger des documents (des articles, des thèses ou des livres, etc.). Lamport (Lamport, 1994) a écrit LaTeX en se basant sur le formateur de texte (TeX²⁰) de Knuth (Knuth, 1986). En effet, Lamport a développé une collection de macro-commandes destinées à simplifier l'utilisation de TeX.

Avec LaTeX, on peut insérer dans le texte des figures, des tableaux, des expressions mathématiques, etc. L'utilisateur n'édite pas directement le document final en mode WYSIWYG (comme avec Microsoft Word par exemple), mais il part d'un fichier source (avec l'extension tex) qui sera compilé pour arriver à un fichier cible (en général, de type dvi, ps ou pdf).

La Figure 1.12 met en lumière un exemple simple document LaTeX qui permet d'afficher la fraction (1.1) de la section précédente.

¹⁹ LaTeX : <http://www.latex-project.org/>. Consulté le 4/4/2009.

²⁰ TeX : <http://www-cs-faculty.stanford.edu/~knuth/>. Consulté le 4/4/2009.

```

\documentclass[12pt]{article}
\usepackage{amsmath}
\begin{document}
  \begin{align}
    \frac{x-1}{x+1}
  \end{align}
\end{document}

```

Figure 1.12 Une fraction écrite en LaTeX.

LaTeX possède un mode mathématique très avancé permettant l'édition d'expressions mathématiques de façon simple et précise. Il gère efficacement la mise en page de ces expressions. Pour ces raisons, il est souvent utilisé par des individus qui rédigent des documents dont le contenu est parfois complexe (des symboles et des équations). Il est important pour les non-voyants de pouvoir lire ce type de documents. Plusieurs méthodes sont adoptées actuellement pour assurer l'accès aux documents mathématiques pour les non-voyants. Nous explorons ces méthodes dans les sections qui suivent.

1.4.2 Les approches Braille/tactiles

Il existe plusieurs méthodes pour présenter les expressions mathématiques aux personnes atteintes d'une déficience visuelle qui exploitent le relief, le braille ou un de ses dérivés. Nous présentons, dans les sections suivantes, les principales approches tactiles. Mais tout d'abord, nous abordons les notations utilisées pour rédiger les mathématiques en braille.

Les notations mathématiques

Bien que les mathématiques soient une langue universelle, le code mathématique braille n'est, bien au contraire, pas du tout standardisé. En effet, la notation mathématique braille diffère d'un pays à un autre et d'une langue à une autre. Par exemple, en France, on utilise le braille communément Français (*Commission Evolution du Braille Francais*, 2001). Aux États-Unis, au Canada, en Nouvelle Zélande, en Grèce et en inde, on emploie le braille mathématique Nemeth qui tient son nom de son inventeur Dr. Abraham Nemeth (Nemeth, 1972). Par ailleurs, ItalBra (*Biblioteca Italiana per Ciechi "Regina Margherita" O.N.L.U.S.* ,

1998) est la notation qui est en vigueur en Italie, alors que Marburg (Epheser, Pograniczna et Britz, 1992) est la notation utilisée au sein de la communauté allemande (i.e. l'Allemagne et l'Autriche). Il faut aussi noter que la notation britannique (*Braille Authority of the United Kingdom*, 1987) est répandue en Grande-Bretagne. Il y a aussi la notation (Stuttgart) (Schweikhardt, 1987) basée sur le braille informatique (8 picots) (Schweikhardt, 1998). D'autres pays ont aussi leurs propres notations et certains d'entre eux en ont même plusieurs. Tout ceci constitue un problème en soi.

La forme braille linéaire

Elle permet d'écrire les formules de manière entièrement linéaire, en utilisant des caractères supplémentaires afin de désambiguïser les expressions (début et fin du bloc). Par exemple, pour présenter linéairement une équation, il faut délimiter le numérateur, dénominateur, les exposants, la racine carrée, etc. L'équation (1.1) s'écrit linéairement comme suit :

$$(x-1)/(x+1) \quad (1.2)$$

Cette méthode est la plus populaire et elle est enseignée dans presque tous les centres éducatifs spécialisés dans l'enseignement pour les non-voyants. Il faut mentionner que les aveugles maîtrisant le braille ne sont pas nombreux. Ainsi, la lecture du braille est lente (par rapport aux méthodes utilisées par les voyants) et cela signifie qu'il faut plus de temps pour lire et comprendre une formule mathématique écrite en Braille. L'accès via la lecture peut être facilement effectué grâce aux afficheurs braille ou bien à l'aide d'impressions sur papiers.

Cependant, cette méthode exige d'incorporer trop de parenthèses, ainsi la plupart des notations braille utilisent une grande variété de symboles pour délimiter les différents blocs. Par exemple, pour Nemeth, la fraction (1.1) s'écrit de la façon suivante :

$$?x-1/x+1# \quad (1.3)$$

Où, « ? » et « # » indiquent le début et la fin de la fraction respectivement.

L'utilisation des différents symboles, afin de marquer différentes catégories de blocs (fraction, racine carrée, exposant, etc.), est importante pour une présentation linéaire. Les données sont généralement lues de gauche à droite et caractère par caractère d'une manière séquentielle. Si les parenthèses régulières « (» et «) » ont été utilisées au lieu de « ? » et « # », alors le non-voyant connaîtra la nature de l'expression seulement quand il lit le symbole de la fraction soit « / ». L'utilisation des symboles spéciaux pour donner au lecteur un aperçu de l'expression (i.e. remplace le coup d'œil chez un voyant) est une caractéristique importante de la conception de notations mathématiques braille (Karshmer, Gupta et Pontelli, 2008). Cela aide beaucoup le lecteur à comprendre l'expression (Stevens et Edwards, 1994a). En effet, un aperçu permet surtout à un lecteur aveugle de prévoir sa stratégie de lecture, et par conséquent de planifier la navigation. D'autre part, un bon aperçu de la structure globale d'une expression mathématique peut être utile dans certains cas comme la recherche d'une formule particulière (dont l'utilisateur connaît l'aspect général) parmi une liste d'expressions.

La Figure 1.13 montre les différentes transcriptions de la fraction (1.2), ainsi que les exemples utilisés qui sont inspirés de (Archambault et al., 2005). La transcription de l'expression varie selon la notation braille utilisée. Les caractères qui marquent le début et la fin du bloc varient de la première version du braille français à la deuxième. Dans Nemeth, on utilise un caractère pour le début de la fraction et un autre pour la fin. Cependant, dans ItalBra, plusieurs caractères noirs (i.e. x et 1) sont représentés par deux caractères braille.

Il existe plusieurs outils permettant la traduction de tout type de document (LaTeX, MathML, texte, et autres) vers le braille linéaire. Les principaux travaux dans ce domaine sont:

- **BraMaNet** (*BRAille MATHématique sur le NET*) (Schwebel, 2000) transcrit les expressions codées en MathML en une chaîne de caractères correspondant à l'expression mathématique en braille français. Un document mathématique écrit avec l'éditeur d'équation de Word ou avec MathType sera d'abord exporté en MathML et ensuite il sera converti en braille;

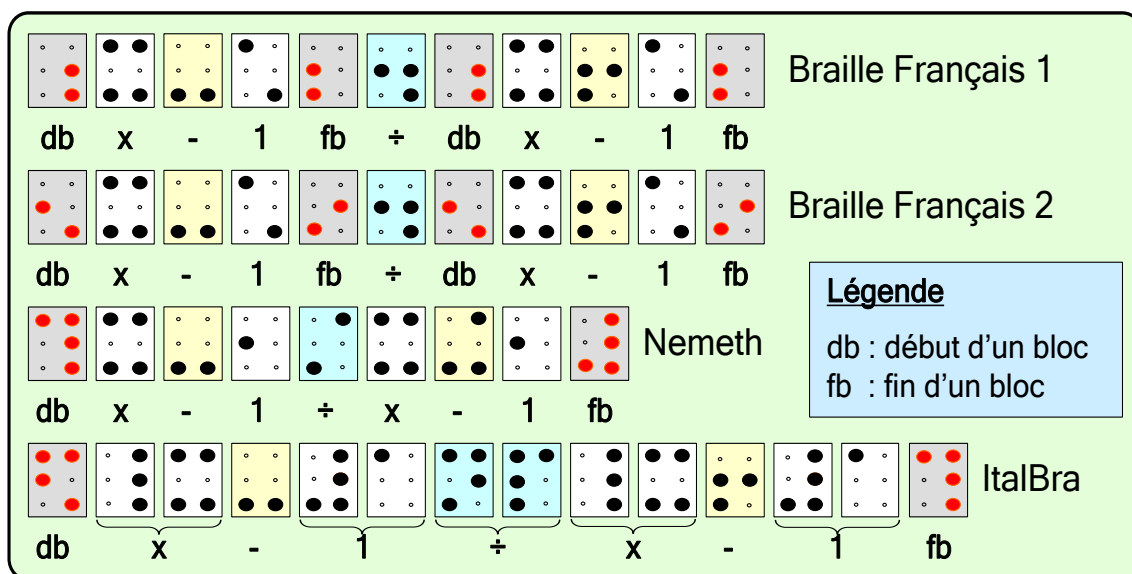


Figure 1.13 La transcription d'une fraction dans 4 différentes notations.

- **LaBraDoor** (*LaTeX-to-Braille-Door*) (Batusic, Miesenberger et Stöger, 1998) a été conçu pour convertir des documents LaTeX en braille (Marburg). Par la suite, une nouvelle version (Batusic, Miesenberger et Stöger, 2003) a été développée pour la conversion inverse (Braille vers le texte en noire);
- **Multi-Language Mathematical Braille Translator** (Moço et Archambault, 2004; Moço et Archambault, 2003b) est basé sur les approches développées dans le projet UMA (*Universal Mathematics Accessibility*) (Karshmer et al., 2004). Il permet la conversion entre plusieurs notations braille (le Français, Marburg et Nemeth);
- **LAMBDA** (Schweikhardt et al., 2006) permet de présenter les données (principalement écrites en MathML) en braille et sous forme audio. Il est basé sur l'utilisation conjointe d'une présentation linéaire et d'un éditeur mathématique prévu pour les besoins éducatifs des étudiants et des professeurs.

D'autres outils et projets ont été développés dans ce domaine. Nous pouvons citer Math2Braille (Crombie et al., 2004), VICKIE (Moço et Archambault, 2003a) et bien d'autres

encore. Chacun de ces outils transcrit les formules codées en (MathML ou LaTeX) en une chaîne de caractères correspondant à l'expression mathématique en braille. Ils offrent souvent la possibilité de choisir la notation mathématique braille. De plus, la plupart supporte les trois formats populaires visuels (MathML, LaTeX et MathType).

Cependant, lors de la conversion, plusieurs problèmes peuvent être envisagés. Pour la même notation, plusieurs sémantiques sont possibles. Par exemple, la notation 2km peut être « le produit des 2 et les variables k et m) » ou « deux kilomètres » ou ... Dans BraMaNet, les unités de mesures sont bien identifiées s'il existe un espace qui sépare le chiffre de l'unité « 2 km » mais pas dans l'autre cas « 2km ». Pour les symboles trigonométriques, logarithmiques, etc., il est impossible de savoir quand se termine l'argument lorsque les parenthèses ne sont pas écrites au moment de l'édition de l'expression avec MathML.

Le braille et les symboles en relief

Cette présentation des symboles a été développée au sein du projet DotsPlus²¹ (Gardner, 1993). L'idée est de garder la même mise en page des expressions mathématiques que pour l'écriture en noir, et l'on remplace les caractères alphanumériques par leurs équivalents Braille. Cette méthode a l'avantage de garder la « bi-dimensionnalité » de la formule. L'alphabet de 64 caractères utilisé dans le braille standard est suffisant pour coder les caractères alphanumériques, et les autres symboles mathématiques seront transcrits en relief. Dans ce projet (Gardner, Ungier et Boyer, 2006; Preddy et al., 1997), les graphiques et les courbes sont aussi pris en charge. Ils sont embossés sur du papier pour permettre à un utilisateur non-voyant d'y accéder.

La Figure 1.14 montre une formule sous sa forme bidimensionnelle (forme habituellement utilisée par les voyants), la transcription en braille linéaire (braille français) et celle en DotsPlus.

²¹ DotsPlus : <http://dots.physics.orst.edu/dotsplus.html>. Consulté le 4/4/2009.

La présentation linéaire compte plus de caractères que les deux autres formes. Par exemple, on utilise deux caractères pour désigner l'indice et deux autres pour indiquer les limites d'un bloc. Cependant, DotsPlus simplifie la présentation en gardant sa structure bidimensionnelle et de ce fait évite les caractères additionnels.

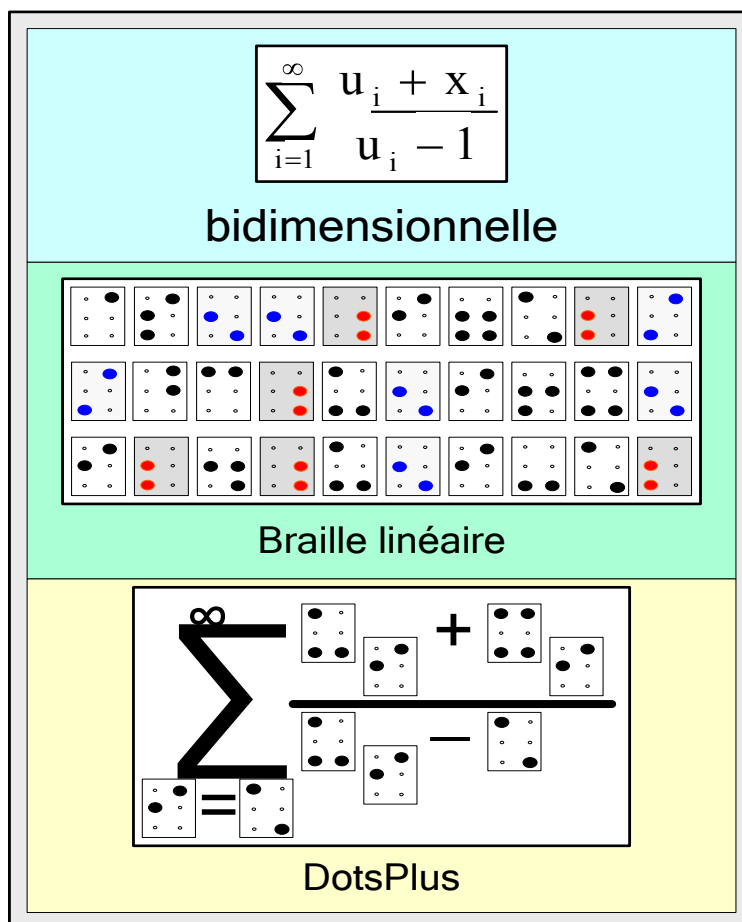


Figure 1.14 Une expression écrite sous trois différentes formes.

Le principal problème de cette méthode est l'apprentissage qu'elle nécessite. En effet, plusieurs études (Favre, 2003; Gardner, 1998; Podevin, 2002) ont montré que l'apprentissage de la reconnaissance tactile des formes en relief peut être long. Il faut dire aussi, que les personnes ayant déjà vu (non aveugle de naissance) peuvent se fier aux symboles embossés et les insérer dans le code braille car elles peuvent les conceptualiser. Par contre, les gens aveugles de naissance vont trouver plus de difficultés à conceptualiser ces symboles.

Cette méthode n'est pas adaptée à une utilisation dans le cadre de l'informatique, parce qu'elle nécessite des dispositifs capables d'afficher dynamiquement des images en relief (par exemple, DMD 120060) mais ces périphériques ne sont pas accessibles facilement et ils coûtent pour l'instant trop cher. Dans le projet DotsPlus, les imprimantes TIGER (Walsh et Gardner, 2001) sont employées pour l'impression en relief.

DotsPlus est un système qui sert juste à donner accès, en lecture, aux mathématiques. En effet, l'utilisateur non-voyant ne peut pas composer lui-même les expressions. Dans cette perspective, l'utilisateur a besoin d'utiliser un autre média et un code différent. Il doit alors apprendre deux codes (un pour lire et l'autre pour écrire).

Le braille et la présentation arborescente

Cette méthode correspond à une architecture graphique où les éléments non terminaux sont associés à une descendance. Ainsi, dans un arbre apparaît la notion de père et de fils. Pour adapter la notion de l'arbre aux formules mathématiques, il suffit de faire correspondre des éléments non terminaux à des opérateurs et les éléments terminaux à des opérandes. On propose à la personne de se déplacer dans l'arbre à l'aide de commandes simples pour lire les sous-expressions qui seront affichées en braille linéaire sur un terminal adapté.

MAVIS²² (Mathematics Accessible to Visually Impaired Students) (Karshmer et al., 1998) est le système qui emploie cette technique. Il la dénomme le traitement par « chunking » et les sous expressions sont appelées des « chunks ». Avec cette méthode, on peut déterminer même le niveau d'abstraction et on peut décider de lire ou non les détails de la formule ou bien se limiter à un aperçu global. MAVIS est une application qui s'adresse aux enseignants de mathématiques et à leurs étudiants aveugles. En effet, il permet aux professeurs de préparer leurs documents en LaTeX, de les traduire en braille (la notation américaine Nemeth

²² MAVIS : <http://www.nmsu.edu/~mavis/>. Consulté le 4/4/2009.

est employée) et de les rendre accessible via un terminal braille. Ainsi, l'étudiant lit et édite ses formules mathématiques en braille.

La principale notion apparaissant dans MAVIS est la notion de chunks. Dans ce projet, on découpe les expressions mathématiques selon le même principe qu'en informatique linguistique. On sépare les termes en chunks. Dans le cas d'une égalité, les trois principaux chunks vont être : le membre de gauche, le signe égal et le membre de droite. Ce découpage permet, pour un étudiant aveugle, une navigation plus simple au sein d'une formule. À chaque fois qu'il rencontre un nouveau chunk il peut, à l'aide des boutons d'une souris, soit le passer (un simple clic permettant de descendre dans l'arbre syntaxique), soit l'expliquer (deux clics pour lire le chunk courant).

Ainsi, l'utilisateur contrôle la manière dont l'expression lui est présentée. Il est possible de déterminer l'ordre selon lequel les termes de la formule vont être lus. L'utilisateur peut lire l'expression en entier ou demander un aperçu de la formule. C'est une méthode qui favorise l'interaction entre l'utilisateur et la machine. Cependant, elle perd son utilité avec des documents imprimés.

La Figure 1.15 montre une présentation arborescente de la formule (1.4).

$$\frac{2+x^2}{5} \quad (1.4)$$

L'inconvénient majeur de cette méthode est que la notation de l'arbre n'est pas courante et demande un apprentissage. Par ailleurs, cette méthode nécessite la maîtrise du braille linéaire (connaissance de Nemeth Code dans le cas de MAVIS), de LaTeX (pour produire les documents mathématiques), et de la notation mathématique préfixée (la notation sous forme d'arbre n'est pas connue de tous).

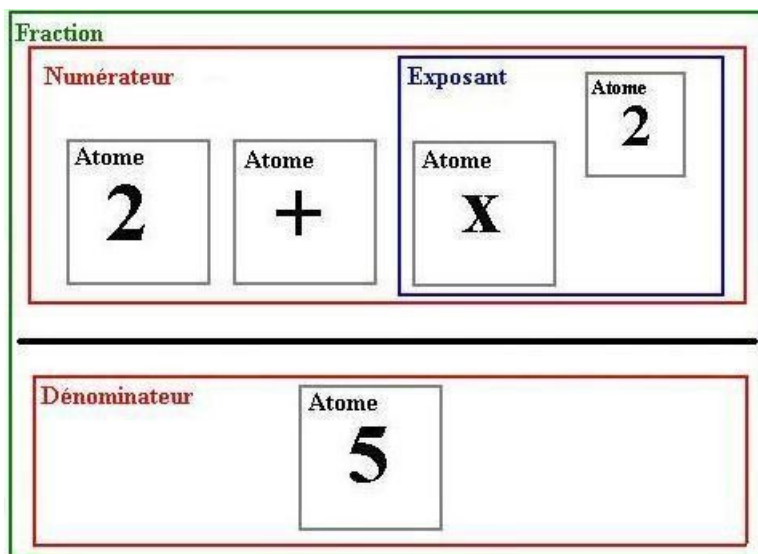


Figure 1.15 Un arbre qui représente une fraction.

Le braille mathématique spatialisé

Les voyants adoptent un formalisme bidimensionnel car il permet d'avoir en lecture un aperçu de la formule, de trouver facilement des repères visuels, de mémoriser rapidement la formule et de l'éditer sans difficulté. Pour la transcription en braille, une question se pose : si l'on adopte cette bi-dimensionnalité, peut-on aussi donner cet aperçu de la structure de la formule?

La spatialisation offre immédiatement des repères, et même si l'on ne se souvient plus du niveau de décomposition, il suffit de parcourir rapidement les « grands opérateurs » pour retrouver celui que l'on recherche. Ainsi, présenter l'expression, pour le non-voyant, de manière plus spatialisée, lui donnera un aperçu général de la formule et cela aide à faciliter la mémorisation de l'expression. Cette présentation en 2D peut lui offrir de meilleurs repères spatiaux et une meilleure structuration de la formule. Dans le cas d'une expression complexe, cela devient intéressant. Mais, pour une expression simple, il faut mieux rester avec une autre présentation, comme la linéaire ou l'arborescente.

L'idée de cette présentation a été développée dans le cadre du projet EasyMaths (Podevin, 2002). L'idée est alors de garder en partie les concepts de spatialisation de l'écriture en noir en les adaptant aux non-voyants pour gagner en clarté sur la présentation des formules. On décompose les expressions complexes en sous expressions plus simples à l'aide de pointeurs spatiaux et d'arborescences. Les concepts de pointeur, substitution et décomposition forment la partie majeure de la représentation des formules. Le symbole carré ■ représente physiquement l'indicateur de pointeur et signale que quelque chose est développée ailleurs sur la feuille. La Figure 1.16 montre la formule²³ (1.5) en utilisant une présentation spatialisée.

$$\text{prod}(a, b) = \frac{\prod_{j=1}^n (a_j + b_j)}{(\sum_{j=1}^{n-1} a_j)(\sum_{j=1}^{n-1} b_j)(a_n + b_n)} \quad (1.5)$$

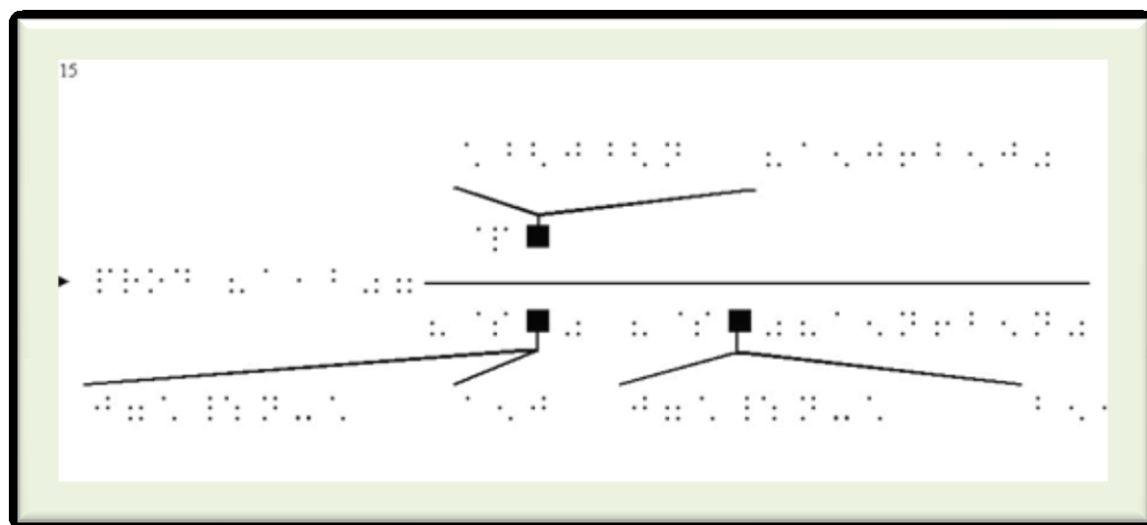


Figure 1.16 Une formule présentée par EasyMath lors de l'évaluation du système.

Cette méthode résout en partie le problème de la bi-dimensionnalité. En effet, grâce aux pointeurs, on évite d'écrire un grand nombre de caractères qui ne font pas réellement partie de la formule et ne sont en fait que pour clarifier la mise en forme. Par exemple, la lecture des fractions, dont le numérateur et le dénominateur sont longs, est simplifiée car l'utilisateur

²³ C'est la formule no. 15 présentée lors de l'évaluation du projet EasyMath.

n'est plus obligé de faire correspondre les parenthèses qui délimitent l'un par rapport à l'autre. Ici, c'est la barre de fraction, transcrite en relief, qui l'indique clairement.

Par contre, on ne peut pas éviter l'apprentissage tout comme les nouvelles méthodes. Le fait d'introduire des symboles et des graphiques nécessite d'avoir un périphérique dynamique (rafraîchissable). Par ailleurs, des documents écrits à l'aide de cette méthode exigent beaucoup d'espace. Enfin, l'utilisation du symbole ■ comme indicateur de pointeur n'offre pas à l'utilisateur une idée du terme substitué. Nous pensons alors que les sons non parlés pourraient aider dans ce contexte.

Après avoir vu les principales approches braille/tactiles, nous procédons à une synthèse des approches sonores.

1.4.3 Les approches sonores

Il existe deux types de méthodes sonores : le premier type permet une lecture précise de la totalité de la formule, alors que le deuxième n'offre qu'un aperçu de l'expression afin d'avoir une idée de sa complexité et de son aspect général.

La lecture séquentielle à haute voix

La lecture, à haute voix, de la formule est une autre forme de présentation des données. Cette présentation a été utilisée depuis longtemps vu qu'elle est facile et n'exige pas une phase d'apprentissage longue. En effet, la parole est une forme de communication naturelle. Par contre, si seule la lecture à voix haute est utilisée, cela peut devenir très ambigu pour l'auditeur. Il existe deux approches pour éviter l'ambiguïté :

L'utilisation d'indicateurs lexicaux : on ajoute des informations supplémentaires pour délimiter les blocs dans une expression. Ces indicateurs correspondent aux parenthèses dans l'écriture en noire. Par exemple, la fraction (2.1) sera lue comme suit :

Début numérateur x moins 1 fin numérateur divisé par début dénominateur x plus 1 fin dénominateur.

Cette approche a été adoptée dans la plupart des outils qui présentent les expressions mathématiques vocalement, et elle a été reconnue efficace quand on traite des expressions ayant des blocs importants (racine carrée, fraction, sin, cos, etc.). D'autre part, des études comme (Stevens, Edwards et Harling, 1997) ont démontré les aspects négatifs causés par les indicateurs lexicaux. Ils sont surtout associés à la surcharge de la mémoire du lecteur.

L'utilisation de la prosodie : le changement est dû au synthétiseur de parole qui tente de conserver le caractère multidimensionnel de l'information. Pour cela, la lecture n'est pas monotone. Plus précisément, la façon dont l'expression mathématique est lue « la tonalité (emphase), rythme (pause) et la hauteur de la voix (aiguë, médium ou grave) » donne des informations qui aident à comprendre le sens de cette expression. La formule est explicitée grâce à la prosodie. Des mots sont également rajoutés parfois pour une meilleure compréhension.

Dans ce contexte, le projet MathTalk (Stevens et Edwards, 1994a; Stevens et Edwards, 1994b) a implémenté cette approche. Il repose sur deux idées principales: la navigation au sein d'une expression mathématique et la possibilité d'accéder à un aperçu général de la formule. Un simple enregistrement sonore (magnétophone, livres enregistrés, etc.) ne permet pas de « naviguer » dans l'expression. C'est pour cela que MathTalk met à la disposition de l'utilisateur plusieurs fonctions de navigation. Cette navigation, indirecte et directe, par commandes vocales ou par raccourcis clavier, est basée sur la structure syntaxique de l'expression mathématique. Il y a huit actions possibles ("aller au début", "expliciter", etc.), couplées à huit cibles différentes ("exposant", "numérateur", etc.) permettant d'atteindre rapidement n'importe quel terme de la formule. Avec la possibilité de naviguer au sein de l'expression mathématique, l'utilisateur contrôle mieux le processus de lecture. Un langage a été élaboré afin de bien formuler les expressions mathématiques. En lecture, la méthode de parcours semble être efficace. Cependant, elle n'est valable que si les expressions

mathématiques sont formulées correctement, c'est-à-dire que le formalisme imposé par le langage soit respecté. Le lecteur n'est pas obligé d'avoir connaissance de ce langage, mais l'auteur des formules doit l'être.

MathTalk ne s'occupe pas de la saisie. C'est pour cela que MATHS²⁴ (Mathematical Access for Technology and Science) (Dix, 1996; Stevens, 1996) a été développé. Cet outil repose sur une extension de MathTalk. Les auteurs cherchaient à étendre ce programme à la manipulation des expressions via la reconnaissance de la parole ou via le clavier. La saisie des formules mathématiques est quelque chose de primordial, non seulement pour communiquer avec autrui, mais pour servir d'aide-mémoire. Il est donc nécessaire de concevoir un outil permettant la lecture ainsi que la saisie d'expressions mathématiques. Pour la saisie de textes ou d'expressions, on utilise un clavier conventionnel, la parole et le braille. Pour la sortie, MATHS emploie la synthèse de parole, du braille, un écran (pour les voyants ou mal voyants) et un ensemble dans des sons non parlés. L'interaction par un terminal braille se fait en entrée comme en sortie.

Dans les deux projets suivants, ASTER (Raman, 1994; 1998) et AudioMath (Ferreira et Freitas, 2004; 2005) l'utilisation de la prosodie pour désigner la structure de l'expression est employée. Le premier traite les documents écrits en LaTeX, alors que le deuxième permet la lecture des documents codés en MathML.

D'autres projets comme MathGenie (Gillan et al., 2004; Karshmer, Bledsoe et Stanley, 2004), VoiceXML (Reddy, Annamalai et Gupta, 2004) et MathPlayer²⁵ (Soiffer, 2005; 2007) permettent la lecture et la navigation dans des expressions codées en MathML. Dans ces systèmes, l'utilisateur a l'avantage d'utiliser des « repères audio » pour dénoter une sous-expression lors de la navigation. Ces repères peuvent être utiles pour que l'utilisateur puisse ajuster la lecture de l'expression à son rythme et à ses besoins.

²⁴ MATHS : <http://www.cs.york.ac.uk/math/math.html>. Consulté le 4/4/2009.

²⁵ MathPlayer : <http://www.dessci.com/en/products/mathplayer/>. Consulté le 4/4/2009.

Cependant, avec des formules longues, cette approche (la lecture à haute voix) n'est pas très avantageuse. Bien qu'apportant de nombreuses améliorations par rapport à une lecture avec une voix de synthèse basique (sans prosodie), l'empan mnésique²⁶ peut ne pas être suffisant pour que l'utilisateur se souvienne du début de la formule lorsqu'il arrive à la fin de celle-ci. Le principal inconvénient de ces deux outils est qu'ils nécessitent en entrée des documents numériques bien formalisés (MathML ou LaTeX), ce qui n'est pas évident pour tous les utilisateurs.

Les sons non parlés (earcons)

Les earcons sont des sons obtenus par utilisation de fréquences musicales organisées selon une grammaire plus ou moins complexe faisant intervenir des changements de hauteur et de rythmes (Blattner, Sumikawa et Greenberg, 1989). Les earcons sont employés normalement pour fournir de l'information à l'utilisateur concernant une action, un objet manipulé, etc.

Des études (Brewster, 1998; Brewster, Wright et Edwards, 1993) ont suggéré l'utilisation d'earcons pour associer les sons non parlés aux différents opérateurs. Par exemple, une fraction complexe est associée à deux longues notes avec une tonalité constante séparée par deux pauses (Stevens, Edwards et Harling, 1997). Pour générer un earcon (Blattner, Sumikawa et Greenberg, 1989), le système parcourt la formule et remplace les caractères par des sons venant d'instruments différents, et joués à différentes hauteurs, fréquences et longueurs.

Les projets MathTalk, MATHS et MAVIS utilisent cette méthode pour donner un aperçu des formules mathématiques. De plus, les earcons sont utilisés pour signaler le début et la fin d'un bloc (racine, numérateur, etc.). Les méthodes utilisant la décomposition en arbre (MAVIS et MathTalk) planifient la manière dont l'ordinateur va présenter la formule. Les earcons ne permettent pas de prendre connaissance de la formule en détails, mais juste un

²⁶ Quantité limitée d'environ $7 (\pm 2)$ unités d'information pouvant être retenues en mémoire à court terme pour une période restreinte de moins d'une minute. (Le grand dictionnaire)

aperçu. C'est le reproche qu'on peut faire à cette méthode. Cette méthode nécessite également un apprentissage difficile. L'utilisateur doit avoir une oreille très attentive et musicale. Il doit apprendre à reconnaître qu'un son donné représente un opérateur donné, que telle note signifie un indice ou un exposant, etc.

Décomposition en arbre

Les expressions mathématiques peuvent être représentées par des arbres comme nous l'avons vu dans les approches braille et en particulier avec le chunking dans MAVIS. La différence est que, cette fois-ci, « l'affichage » ou bien le média de sortie n'est plus le terminal braille, il est remplacé par le système de synthèse vocale avec l'utilisation de la prosodie.

Les commandes vocales permettant la navigation au sein de l'expression sont composées de deux parties : action (lire, aller, sortir, etc.) et cible (terme, expression, numérateur, etc.). Par exemple dans MathTalk, la commande « lire expression » permet de lire la totalité de l'expression, alors que la commande « Terme suivant », si l'on est sur le premier facteur d'une multiplication, passera au deuxième. Ce langage de commande, certes assez simple, suppose que l'utilisateur aveugle ait une représentation mentale bien construite de la formule. À titre d'exemple concret, quand l'utilisateur dit "aller à la racine carrée" cela semble aisé pour un voyant puisque, l'ayant sous les yeux, il sait où elle se trouve. Mais l'aveugle peut ne pas savoir où elle se trouve dans la formule, pouvant ainsi ne plus avoir de repère, ou pouvant ne plus se souvenir de son existence. Enfin, on peut aussi remarquer que peu d'aveugles sont familiers avec la notion d'arbre.

1.5 Les approches pour l'édition des expressions

Nous entendons par accès, aussi bien la lecture que l'écriture d'expressions mathématiques. Dans la littérature, il existe peu de logiciels qui permettent aux utilisateurs non-voyants d'éditer un texte d'une façon aisée et directe (Bernareggi, 2007). En effet, la plupart des logiciels développés sont des outils de transcription automatique en braille ou de synthèse

vocale de textes disponibles sous divers formats électroniques (texte, MathML, LaTeX, etc.). Un étudiant qui désire modifier une équation mathématique n'a pas d'autre choix que de modifier le document original puis de le convertir en braille et/ou audio, en fonction de l'outil utilisé (BraMaNet, AudioMath, ASTER, etc).

La saisie des formules mathématiques est quelque chose de primordial, non seulement pour communiquer avec autrui, mais aussi pour servir d'aide-mémoire. Il est donc nécessaire d'offrir aux utilisateurs un outil de saisie d'expressions. Permettre à l'utilisateur d'éditer lui-même le rend autonome. De plus, l'écriture permet à l'étudiant de faire les calculs plus facilement et d'une manière efficace et rapide.

Le problème majeur des outils qui présentent les expressions mathématiques est l'utilisation d'un format pour écrire les données et un autre pour les afficher. Cela perturbe l'utilisateur et le contraint à apprendre deux codes. En effet, bien que cela soit techniquement possible, utiliser un code pour lire et un autre pour écrire est fastidieux. En effet, ce type d'interface ne correspond pas au formalisme WYSIWYG (what you see is what you get). Nous pensons que le même formalisme devrait obligatoirement être utilisé en lecture et en écriture : un formalisme WYWIWYR (what you write is what you read).

De plus, il semble plus difficile d'écrire en 2D qu'en 1D (Podevin, 2002; Schweikhardt et al., 2006) d'autant plus que quelque soit la discipline étudiée (littérature, mathématiques, musique), les non-voyants lisent habituellement de façon linéaire. Si on propose une façon d'écrire les formules, il faut donc que cela reste accessible. S'il est plus efficace d'écrire les formules en 1D, alors il n'est pas envisageable d'écrire en 1D tout en proposant une lecture en 2D (Edwards, McCartney et Fogarolo, 2006). Pour cette raison, lors du développement du projet Lambda, l'idée d'imprimer les documents en format DotsPlus a été rejetée. Les concepteurs du projet voulaient conserver le même format pour la lecture, la manipulation et l'écriture des formules.

Le problème de la manipulation d'expressions mathématiques, tel que l'accès direct à un terme, apparaît clairement avec l'édition des formules. En effet, si l'utilisateur a pris connaissance de la formule et veut juste éditer un terme spécifique, il doit avoir des moyens qui lui permettent d'y accéder sans être obligé de parcourir et lire le reste de la formule. Pour un voyant, il est très facile d'y accéder et de manipuler la formule dans un laps de temps très court par contre ce processus exige des efforts, du temps et de la concentration de la part du non-voyant.

Avec les approches sonores, on peut dire que l'édition est difficile. L'écriture à l'aide d'un système de reconnaissance vocale, n'est pas si facile car les logiciels de reconnaissance de la parole ne sont pas toujours robustes et les expressions mathématiques ne tolèrent aucune erreur. Dans les formules mathématiques, une erreur sur un seul caractère peut conduire à une interprétation complètement erronée de la formule. Par contre, les systèmes de reconnaissance vocale peuvent bien être utiles pour les commandes de navigation courtes.

Des travaux ont été réalisés dans l'objectif de permettre à des utilisateurs non-voyants de lire et d'éditer des expressions mathématiques. Nous décrivons et analysons, ci-après, les plus importants.

1.5.1 Lambda

Le système Lambda est basé sur l'utilisation conjointe d'un code linéaire, pouvant être immédiatement converti en utilisant MathML, et d'un éditeur mathématique prévu pour les besoins éducatifs des étudiants et des professeurs. C'est un système flexible et entièrement configurable. L'éditeur permet l'écriture d'expressions mathématiques sous forme linéaire, de manière rapide et efficace ainsi qu'une gestion active (écriture, lecture, manipulation, calcul, etc.) à l'aide de périphériques Braille et/ou de synthèse vocale.

L'utilisateur non-voyant peut écrire et éditer les expressions mathématiques et en même temps l'interface génère un code MathML qui permet à l'enseignant de visualiser les expressions en noir. De même, l'enseignant écrit les formules en noir (MathML) et

l'interface le transcrit en braille automatiquement. De ce fait, ce système permet à chaque utilisateur (étudiant aveugle, enseignant voyant) de lire et d'écrire facilement sans être obligé d'apprendre plusieurs codes (un pour l'entrée, un pour la sortie) et l'utilisateur peut aussi choisir la forme qui lui convient.

Les commandes vocales d'édition de ce système sont de trois sortes : les commandes générales (delete character, delete line, delete word, undo, etc.), les commandes avancées (delete the next n words, repeat m times the last n commands, etc.), et les commandes spéciales (copy into buffer or delete a fraction or part of fraction, etc.). Ces commandes sont similaires à celles permettant la navigation au sein de l'expression mathématique.

1.5.2 Le projet Infty

Le projet Infty²⁷ vise principalement les documents mathématiques imprimés. Il permet de numériser ces documents puis de les convertir en formats accessibles par les non-voyants. Le projet se décompose en plusieurs modules :

- **InftyReader** (Fukuda, Ohtake et Suzuki, 2000) est la partie du logiciel qui permet de numériser une page imprimée et d'en reconnaître chaque caractère. Ce module utilise une avancée technologie OCR (Optical Character Recognition) pour la reconnaissance des formules mathématiques imprimées.
- **InftyEditor** (Suzuki et al., 2004) est généralement utilisé pour éditer les documents numérisés par InftyReader. L'utilisateur peut modifier les expressions, ajouter ou supprimer des formules. Les documents édités par InftyEditor sont sauvegardés dans des fichiers XML. Ces fichiers peuvent ensuite être transcrits en Braille et/ou présentés vocalement.
- **InftyConverters** (Suzuki et al., 2004) est utilisé pour convertir les fichiers générés par InftyEditor (i.e. XML) en plusieurs formats : LaTeX, MathML, Braille Japonais, et autres.

²⁷ Le projet Infty : <http://www.inftyproject.org/en/index.html>. Consulté le 4/4/2009.

- **ChattyInfty** (Komada et al., 2006) est une extension du module InftyEditor. Il permet la lecture à haute voix des expressions, éditées avec InftyEditor, à l'aide d'un système de synthèse vocale (Microsoft speech API²⁸). Des données supplémentaires (i.e. début bloc, fin bloc, etc.) sont ajoutées pour éliminer les sources d'ambiguïté.

1.5.3 WinTriangle

WinTriangle²⁹ (Gardner et al., 2002) est un outil qui vise le traitement des fichiers RTF (**Rich Text Format**). Son objectif principal est de permettre aux voyants et non-voyants de s'échanger les documents sans que personne ne soit obligé d'apprendre les formats de l'autre. Il est ainsi capable d'afficher les données sur l'écran pour les voyants, en braille pour les non-voyants ainsi que de les présenter en utilisant un système de synthèse vocale. Le module LaTeX2Tri (Thompson, 2005) permet d'importer les documents mathématiques écrits en MathML, LaTeX, etc. WinTriangle est utilisé dans plusieurs universités. Les professeurs préparent les documents (i.e. les devoirs) en format lisible pas les voyants. Les étudiants non-voyants ont donc la possibilité d'éditer ces documents pour les remettre au professeur dans un format lisible par les voyants.

Nous avons constaté que le problème de l'accès en écriture nécessite la résolution au préalable du problème de l'accès en lecture pour que l'on ait une seule présentation permettant l'écriture et la lecture. Le même formalisme doit être utilisé en lecture et en écriture : un formalisme WYWIWYR.

1.6 Comparaison et analyse des approches

Plusieurs études comparatives ont été menées auprès des utilisateurs non-voyants pour évaluer les approches précédentes via les systèmes développés (Favre, 2003; Podevin, 2002). Aucune de ces études n'a présumé que telle ou telle approche est la meilleure. En effet,

²⁸ Microsoft Speech technology: <http://www.microsoft.com/speech/speech2007/default.msp>. Consulté le 4/4/2009.

²⁹ WinTriangle : <http://wintriangle.org/>. Consulté le 4/4/2009.

chaque approche possède des points forts ainsi que des inconvénients et cela varie d'un utilisateur à un autre, selon son profil (son niveau collégial, universitaire, ses connaissances des mathématiques, l'âge où il a perdu la vue, les habitudes avec les différents médias, etc.). Dans ce qui suit, nous présentons une comparaison entre les différentes approches présentées:

- La présentation bidimensionnelle des expressions mathématiques écrites en noire est conservée par plusieurs projets (DotsPlus et EasyMath). Elle est utile pour les personnes qui, avant de perdre leur vue, ont déjà employé une méthode de présentation de formules fortement semblable. Malgré l'apprentissage nécessaire pour la reconnaissance des caractères en noir (dans le cas de DotsPlus), les personnes non-voyantes apprécient le fait d'avoir une structure spatiale stricte dans l'expression. Elle permet, d'une part, de réduire le nombre de données supplémentaires (même si on ajoute les symboles mathématiques noirs) et, d'autre part, de lever l'ambiguïté posée par les autres présentations.
- La lecture des formules mathématiques à voix haute avec prosodie, utilisée principalement par AudioMath, MahtTalk et ASTER, n'exige pas un apprentissage très difficile de la part de l'utilisateur aveugle. Cependant, celle-ci ne favorise pas la navigation au sein de l'expression mathématique. Dans les études, on note que la présence de la prosodie dans tous les systèmes sonores est très bien accueillie, voire même jugée indispensable par certains utilisateurs.
- Les earcons ou les sons non parlés offrent un aperçu global et rapide de l'expression mathématique. Par contre, l'utilisateur doit passer par une phase d'apprentissage. Les données apportées par cette méthode ne sont pas précises. Elle a été utilisée avec les logiciels MathTalk, MAVIS et MATHS. Nous pouvons aussi les utiliser pour une lecture rapide sans prosodie afin de donner un aperçu global de l'expression et, cela, sans avoir à passer par la phase d'apprentissage. Dans ce cas (c.-à-d. pour l'aperçu), les approches sonores sont plus avantageuses que les approches tactiles.
- La plupart des sites et applications Web actuels utilisent MathML pour écrire les formules mathématiques. D'où l'importance des outils utilisant des traducteurs de MathML vers le braille (BraMaNET) ou vers un système de synthèse de parole (AudioMath). Comme peu

des non-voyants maîtrisent le braille, la nécessité d'un projet comme AudioMath et des approches sonores en général est évidente.

- Pour les gens qui maîtrisent le braille, l'utilisation du braille linéaire pour la présentation des formules mathématiques demeure efficace. Cette présentation devient lourde quand l'expression comporte beaucoup de blocs (i.e. parenthèses, exposants, etc.).
- La navigation dans les arbres syntaxiques est simple pour les gens habitués tels que les informaticiens, mais pas évidente pour les personnes qui ne connaissent pas la notation des arbres. Cette approche a été développée dans MAVIS, MathTalk et MATHS. Elle est utilisée conjointement avec la commande vocale pour permettre à l'utilisateur de surveiller le processus de lecture. En effet, l'utilisateur contrôle la manière dont l'expression lui est présentée. Il lui est possible d'avoir connaissance des termes de l'expression dans l'ordre qui lui convient, et même de n'accéder qu'à un aperçu de l'expression.
- Le braille mathématique n'est pas une approche qui engendre beaucoup d'erreurs. Par contre, on note que c'est celui pour lequel le temps moyen de lecture est le plus important. Cela tend à prouver que toutes les autres approches apportent, chacune à sa manière, un plus à la rapidité de lecture et donc un confort pour l'utilisateur.

1.7 Résumé : les approches pour l'accès aux informations par les non-voyants

Le tableau 2.1 montre la relation entre les formes de présentation et les médias utilisés au sein des outils existants.

Le tableau 2.2 montre les méthodes utilisées par ces outils pour présenter des données aux déficients visuels.

Tableau 1.1 Les médias utilisés en entrée et en sortie dans les outils développés pour accéder aux mathématiques.

Média Outils	Entrée				Sortie					forme	
	Clavier	Terminal Braille	Overlay	Reconnaissance vocale	Terminal Braille	Relief (imprimante)	Synthèse vocale	DMD	Sons (Earcons)	Audio	tactile
ASTER	+						+		+	+	
AudioMath	+						+			+	
BraMaNet	+				+						+
DotsPlus	+				+	+					+
EasyMath	+		+			+	+	+		+	+
Infty	+	+		+	+		+			+	+
LaBraDoor	+				+						+
MATHS	+	+		+	+		+		+	+	+
MathTalk	+						+		+	+	
MAVIS	+	+			+		+		+	+	+

Cette partie du rapport nous a permis de faire une comparaison entre les différentes approches et de voir que chacune a ses avantages et ses inconvénients. Ces approches ne sont pas toujours équivalentes mais sont souvent complémentaires. D'où l'idée de concevoir un système multimodal qui permet de combiner les différentes approches pour présenter les expressions mathématiques sur plusieurs modes et sous les différentes formes possibles. Dans la section suivante, nous présentons la multimodalité et son apport dans les systèmes destinés aux non-voyants.

Tableau 1.2 Les outils développés dans la littérature pour présenter les mathématiques aux non-voyants.

Projets	Présentation de l'expression	Représentation interne	Remarques
ASTER	Lecture dirigée	Classification hiérarchique en termes d'objet, arbre étiqueté, règles AFL	
AudioMath	Lecture avec prosodie	Linéaire MathML	
BraMaNet	Braille linéaire	Linéaire MathML	
DotsPlus	Maintien de la disposition spatiale	Sous Ms-Word, il s'agit de la « fonte » particulière TIGER	Impression rapide
EasyMath	Maintien de la structure bidimensionnelle de l'expression		Il exige l'utilisation d'un « overlay ».
Infty	Braille linéaire & synthèse vocale	Document XML	
LaBraDoor	Braille linéaire	Linéaire LaTeX	
MATHS	Audio glance, lecture dirigée par l'utilisateur et Braille linéaire	Éditeur SGML	Navigation séquentielle et techniques de navigation basées sur la structure du document lu.
MathTalk	Audio glance et lecture dirigée par l'utilisateur		
MAVIS	Découverte pas à pas	Arbre d'analyse syntaxique, chunks et XML.	

1.8 La multimodalité

La multimodalité permet à un utilisateur d'interagir avec la machine en exploitant différentes modalités de communication (vocale, auditive, tactile, visuelle, etc.). Dans un système multimodal, il est possible d'exploiter plusieurs modalités simultanément ou pas, qu'elles soient combinées ou non, de manière complémentaire ou pas lors de l'interaction entre l'utilisateur et la machine. Ces systèmes visaient à améliorer l'interaction homme-machine aussi bien au niveau de la performance et de la flexibilité que celui du confort (Bellik, 1995).

Contrairement à un système multimédia, un système multimodal est apte à analyser et à interpréter les informations acquises (Coutaz et Caelen, 1991). En effet, lorsqu'un système multimodal reçoit des informations, provenant de plusieurs modalités, il en fait une analyse pour en donner une interprétation.

En entrée (dans le sens de l'utilisateur vers la machine) : La multimodalité naît de l'association de plusieurs modalités comme la parole, le geste, l'utilisation du clavier et de la souris, etc.

En sortie (dans le sens de la machine vers l'utilisateur) : La multimodalité correspond à la combinaison de plusieurs modalités : texte, bruit, musique, braille, vibration, etc. À noter que la modalité en sortie est définie par la structure d'information telle qu'elle est perçue par l'utilisateur et non pas telle qu'elle est représentée au sein de la machine (Bellik, 1995).

Richard Bolt (Bolt, 1980) a conçu la première application multimodale. Celle-ci permettait de créer et de manipuler des objets graphiques à l'aide de la parole et du geste. Ce travail a été publié sous le nom « Put That There » et est devenu le paradigme des interfaces multimodales. L'utilisation et le développement de ce type de systèmes ne s'est répandu que dix ans après la publication de ce travail, probablement pour des raisons technologiques.

À partir des années 90, les recherches sur les applications multimodales ont commencé à prendre de l'ampleur. Nous citons ici une liste non exhaustive des recherches effectuées dans ce domaine. Les champs d'application varient d'une recherche à une autre. Nous pouvons trouver des applications de base comme un éditeur de texte (Vo et Waibel, 1993), un navigateur internet (Caelen et Bruandet, 2001), des interfaces pour les dessins et un notebook (Nigay et Coutaz, 1993), des logiciels pour l'aménagement intérieur (Ando, Kitahara et Hataoka, 1994), des interfaces de loisirs (Reithinger et al., 2003), des logiciels pour opérer une ligne de production en usine (Cohen et al., 1989), des applications pour la manipulation d'objets (Koons et Sparrell, 1994; Thorisson, Koons et Bolt, 1992), des assistances pour les cartes touristiques (Cheyer et Julia, 1995; Denda, Itoh et Nakagawa,

1997) ainsi que des applications dans l'avionique et le militaire (Bastide et al., 2005; Cohen et al., 1997; Coutaz et al., 1995; Duncan et al., 1999). La multimodalité a également fait l'objet de plusieurs thèses (Bourguet, 1992), (Nigay, 1994), (Bellik, 1995), (Martin, 1995), (Catinis, 1998), (Buisine, 2005), (Jacquet, 2006), (Rousseau, 2006), (Djenidi, 2007). D'autant plus que d'autres chercheurs travaillent encore sur les applications de la multimodalité dans plusieurs disciplines.

L'apport de la multimodalité a été souligné dans plusieurs recherches telles que (Cohen et al., 1989) qui ont démontré les intérêts de l'utilisation synergique des modalités par rapport à « l'unimodalité » (c-à-d. l'utilisation séparée de chaque modalité). Dans un système multimodal, il est possible d'exploiter les forces d'une modalité pour compenser les faiblesses d'une autre. C'est ce qui est appelé dans (Oviatt, 1999) « la désambiguïsation mutuelle ». En ce qui nous concerne, dans (Bellik, 1995), (Burger et Hadjadj, 1999) et (Jacquet et al., 2004) les auteurs ont montré les bénéfices de la multimodalité dans les applications destinées aux non-voyants.

1.9 Utilisation des modalités

1.9.1 TYCOON

Pour Martin (Martin, 1996; Martin, 1995), la multimodalité consiste à faire coopérer des modalités variées, aussi bien en analyse qu'en synthèse, c'est-à-dire de la transmission de l'information, respectivement de l'utilisateur vers le système et du système vers l'utilisateur. Ainsi, un flot d'informations peut être analysé ou synthétisé par plusieurs modalités agissant simultanément, en parallèle, ou l'une après l'autre. Dans (Martin, Veldman et Béroule, 1995; Martin, 1995), les auteurs proposent la taxonomie TYCOON (TYpes de COOpérationN). Six types de coopération entre modalités sont distingués dans TYCOON :

- **La complémentarité** : au sein d'un même énoncé, différentes informations sont transmises sur différentes modalités pour réaliser une tâche donnée.

- **La redondance** : au sein d'un même énoncé, la même information est transmise par différentes modalités.
- **La concurrence** : différentes modalités sont utilisées en parallèle pour réaliser des actions distinctes.
- **L'équivalence** : désigne la possibilité de choisir entre plusieurs modalités pour formuler un énoncé particulier.
- **La spécialisation** : on utilise toujours la même modalité pour exprimer une action particulière.
- **Le transfert** : un énoncé produit par une modalité est analysé par une autre modalité.

Dans cette classification, le parallélisme n'est pas mis en valeur. En effet, les trois types de la coopération (équivalence, spécialisation et transfert) concernent la coopération des modalités successives. Dans ce contexte, d'autres travaux ont fourni d'autres classifications de la coopération des modalités. Nous avons employé le travail de (Catinis, 1998) pour résumer ces taxonomies.

1.9.2 Propriétés CASE

D'autres travaux (Caelen, 2000; Nigay et Coutaz, 1993) établissent une relation entre les modalités (du point de vue du système), et distinguent quatre grands types, représentés par les propriétés CASE (Concurrente, Alternée, Synergique, Exclusive), intégrant la notion de temporalité :

- **La multimodalité concurrente** : différentes modalités sont utilisées en parallèle pour réaliser des tâches distinctes.
- **La multimodalité alternée** : une seule modalité est utilisée à la fois pour produire une action mais plusieurs modalités sont exploitées pour réaliser une même tâche.
- **La multimodalité synergique** : différentes modalités sont utilisées en parallèle pour réaliser une seule tâche.

- **La multimodalité exclusive** : une tâche est exécutée à la fois et en utilisant une seule modalité.

1.9.3 Propriétés CARE

Contrairement aux propriétés CASE qui se concentrent sur l'utilisation des modalités pour réaliser une ou plusieurs tâches, les propriétés CARE (Complémentarité, Assignment, Redondance, Équivalence) (Coutaz et al., 1995) permettent de décrire les relations entre les modalités pour la réalisation d'une seule tâche (Nigay, 1994). À noter que les propriétés CARE définissent la modalité par les deux éléments constitutifs : le dispositif et le langage d'interaction. Elles sont définies comme suit :

- **La complémentarité** : l'utilisateur emploie différentes modalités en même temps, au sein d'un même énoncé, pour réaliser une tâche donnée. Chaque modalité est nécessaire pour comprendre l'énoncé.
- **L'assignment** : l'utilisateur emploie toujours la même modalité pour exprimer une action particulière.
- **La redondance** : l'utilisateur transmet simultanément le même énoncé via plusieurs modalités.
- **L'équivalence** : l'utilisateur a le choix entre plusieurs modalités pour exprimer une action particulière.

La notion de l'assignment de CARE se rapproche de la notion de la spécialisation de TYCOON. Toutefois, dans CARE, l'assignment porte sur une donnée particulière en tenant compte des deux éléments de la modalité qui sont le dispositif et le langage d'interaction. C'est alors un cas particulier de la spécialisation qui s'applique à un type de données (Horchani, 2007).

Les différents niveaux d'abstraction ne sont pas explicités dans la dimension de parallélisme de CASE (Bellik, 1995). C'est pourquoi Bellik a affinée cette classification et a explicité les différents niveaux d'abstraction du parallélisme.

1.9.4 Fusion/Fission

La coopération entre les modalités peut conduire à une fusion (modalités en entrée) et fission (modalités en sortie). Coutaz (Coutaz et al., 1995) et Nigay (Nigay, 1994) distinguent trois types de fusion (fission) selon le niveau d'abstraction de l'information :

- **La fusion sémantique** : Elle constitue la combinaison de commandes pour en obtenir une nouvelle. Exemple: dessiner un cercle et colorier un objet = une commande qui permet de dessiner un cercle selon différentes couleurs.
- **La fusion syntaxique** : Elle correspond à la combinaison d'unités d'information pour obtenir une action. Exemple: l'unité « effacer ça » n'a de sens qu'après sa combinaison avec une autre unité indiquant l'objet à supprimer.
- **La fusion lexicale** : Elle se définit comme étant la combinaison d'actions physiques pour obtenir une action au niveau signal. Exemple: (windows) l'enfoncement conjoint des touches «CTRL» et «ESC» sont fusionnés par le système en un seul évènement.

1.9.5 Les types de multimodalités selon Bellik

Bellik a repris la classification de Coutaz et Nigay et a distingué les critères de fusion selon trois paramètres :

- La production des énoncés (séquentielle, parallèle).
- Le nombre de médias par énoncé (un, plusieurs).
- L'usage des médias (exclusif, simultané).

À partir de ces trois paramètres, Bellik a identifié sept types de multimodalités. La huitième combinaison ne peut pas être produite (voir la Figure 1.17) :

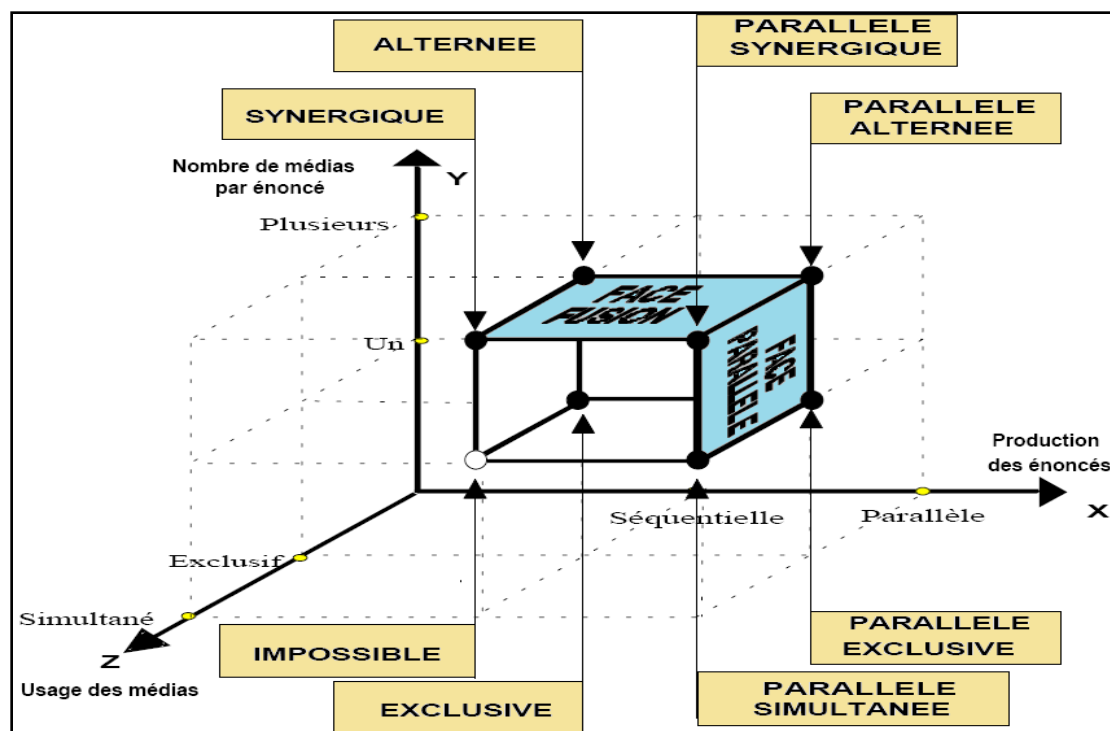


Figure 1.17 Les différents types de multimodalité selon Bellik.

Tirée de Bellik (1995, P.31)

- **Multimodalité exclusive :** un énoncé et un seul média à la fois (pas de multimodalité).
- **Multimodalité alternée :** une tâche est réalisée avec plusieurs médias, mais ils sont employés alternativement.
- **Multimodalité synergique :** une tâche est réalisée avec plusieurs médias employés simultanément.
- **Multimodalité parallèle exclusive :** plusieurs tâches indépendantes sont réalisées en parallèle. Dans une même tâche, un seul média est utilisé. Un seul média est actif à un moment donné.
- **Multimodalité parallèle alternée :** plusieurs tâches sont réalisées en parallèle avec plusieurs médias, mais un seul média est actif à un moment donné.
- **Multimodalité parallèle synergique :** plusieurs tâches sont réalisées en parallèle. Dans une même tâche, plusieurs médias sont employés simultanément.

- **Multimodalité parallèle simultanée** : plusieurs tâches indépendantes sont réalisées en parallèle. Dans une même tâche, un seul média est employé. Les médias de différentes tâches peuvent être utilisés simultanément.

La classification de Bellik se base sur une analyse quantitative et temporelle des données, elle ne tient pas compte de la sémantique des énoncés (Zanello, 1997).

1.10 Taxonomie des modalités

Dans la littérature, plusieurs travaux ont été faits pour classifier les modalités. Nous citons ici les travaux de Bernsen (Bernsen, 1993). Il propose de classifier les modalités de sortie (sens de la machine vers l'utilisateur) en les associant à des modalités de base appelées « modalités pures ». Bernsen a identifié cinq propriétés pour caractériser les modalités : Langagière / Non Langagière, Analogique / Non Analogique, Arbitraire / Non Arbitraire, Statique / Dynamique et Média d'expression.

En ce qui nous concerne, nous adaptons la taxonomie établie par Berrami (Berrami, 2001). Les modalités sont classifiées selon leur mode principal. En effet, dans (Rousseau, 2006), l'auteur permet à une modalité d'être associée à plusieurs modes dont un est principal. Par exemple, le Braille est considéré principalement comme une modalité tactile (mode TPK), mais il peut être aussi lu par les voyants (le mode visuel est ici un mode secondaire).

La Figure 1.18 est inspirée du travail de (Jacquet, 2006). Elle illustre, sous forme d'un arbre (diagramme de classes en UML³⁰), la taxonomie des modalités (appelé arbre taxonomique). Dans cet exemple, seuls les deux modes utilisables par les non-voyants sont pris en compte. Les modes sont représentés par les nœuds intérieurs. Cependant, les feuilles de l'arbre représentent les modalités qui permettent de présenter l'information à l'utilisateur, tandis que les attributs fixent l'instance de la présentation (Rousseau, 2006). En d'autres termes, les attributs permettent de préciser les paramètres de la présentation comme le volume, le nombre de caractères par ligne, etc.

³⁰ Unified Modeling Language.

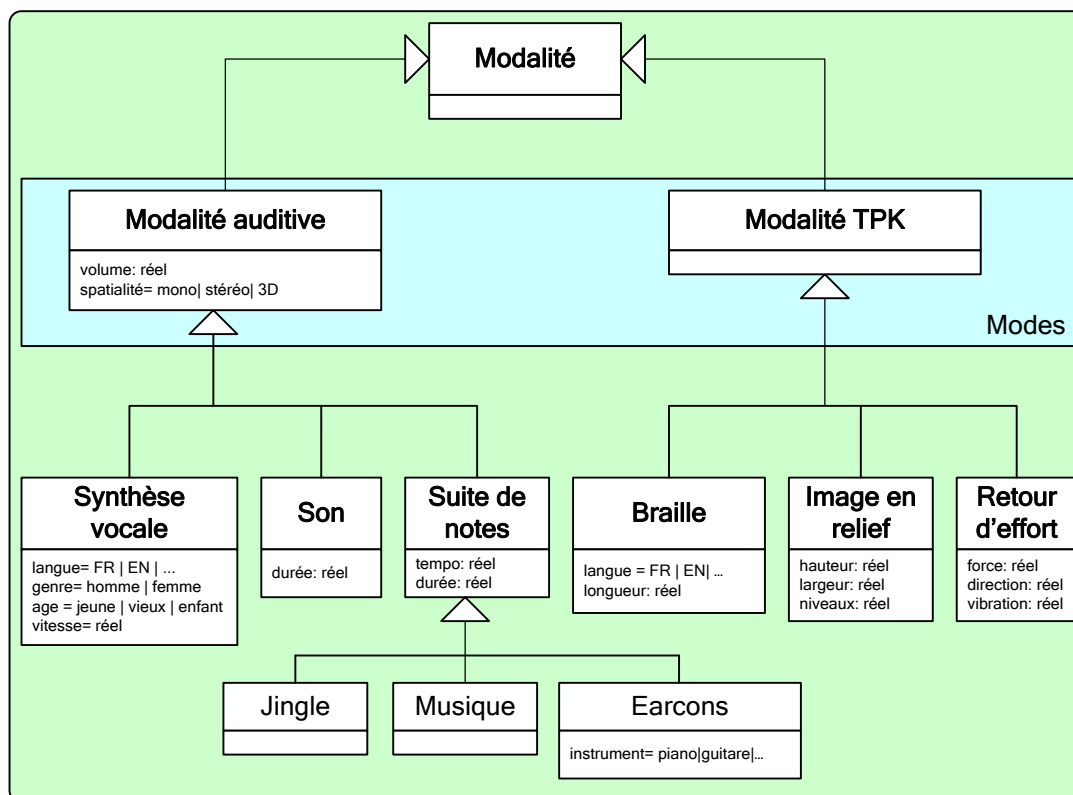


Figure 1.18 Diagramme UML présentant la taxonomie des modalités en sortie. Seuls les deux modes utilisables par les non-voyants sont pris en compte.

1.11 La présentation multimodale de l'information

Cette question s'impose surtout dans les systèmes multimodaux qui traitent des modalités en sortie. En effet, quand une information doit être présentée, le système doit choisir la ou les modalités appropriées. Mais, quels critères faut-il suivre pour sélectionner les modalités parmi l'ensemble de modalités utilisables dans le système?

Dans (André, 2000), on identifie un certain nombre de critères qui aident à faire élire la ou les modalités appropriées :

- Les caractéristiques de l'information à présenter.
- Les caractéristiques des modalités disponibles dans le système.

- Les préférences et les caractéristiques de l'utilisateur.
- Les caractéristiques de la tâche à réaliser par l'utilisateur.
- Les caractéristiques et les limitations des ressources.

Dans le modèle WWHT (What-Which-How-Then) proposé par (Rousseau, Bellik et Vernier, 2005), le choix de la modalité correspond à la phase « Which ». Il s'agit d'identifier les couples (modalité, média) adaptés au contexte d'interaction. Il faut donc prendre en considération l'état de l'utilisateur, son environnement et les ressources du système. Les auteurs ont utilisé une analogie avec le mode politique pour sélectionner (élire) les modalités appropriées (voir la Figure 1.19). Le processus d'élection comporte un modèle comportemental à base de règles (électeurs), qui attribue des points (voix) aux modes et modalités (candidats) en fonction de l'état du contexte de l'interaction (situation économique, etc.). Dans ce processus d'élection, l'élection pure est le meilleur couple (modalité, média) choisi. Cependant, l'élection composée représente les couples redondants ou complémentaires au premier choix.

Dans (Jacquet, 2006), est introduite la notion de profil qui est un ensemble de pondérations pour indiquer les capacités et les préférences des modalités et leurs attributs. Ce profil est défini par un arbre de pondération (voir la Figure 1.20). On parle donc d'ajouter des pondérations à l'arbre taxonomique, pour exprimer les capacités, les préférences et les contraintes de l'entité décrite (utilisateur, dispositif ou unité sémantique).

Une pondération est un nombre réel de l'intervalle $[0;1]$. Elle peut être située à deux niveaux différents :

- Au niveau d'un nœud : Une pondération à 1 signifie que les modalités du sous-arbre sont les modalités préférées alors qu'une pondération à 0 signifie que les modalités correspondantes ne sont pas acceptées. Une pondération entre 0 et 1 permet de déterminer le niveau de préférence de chaque modalité.

- Au niveau d'un attribut : une fonction mathématique est définie pour indiquer la pondération de chaque valeur possible de l'attribut. La signification des pondérations est la même que précédemment.

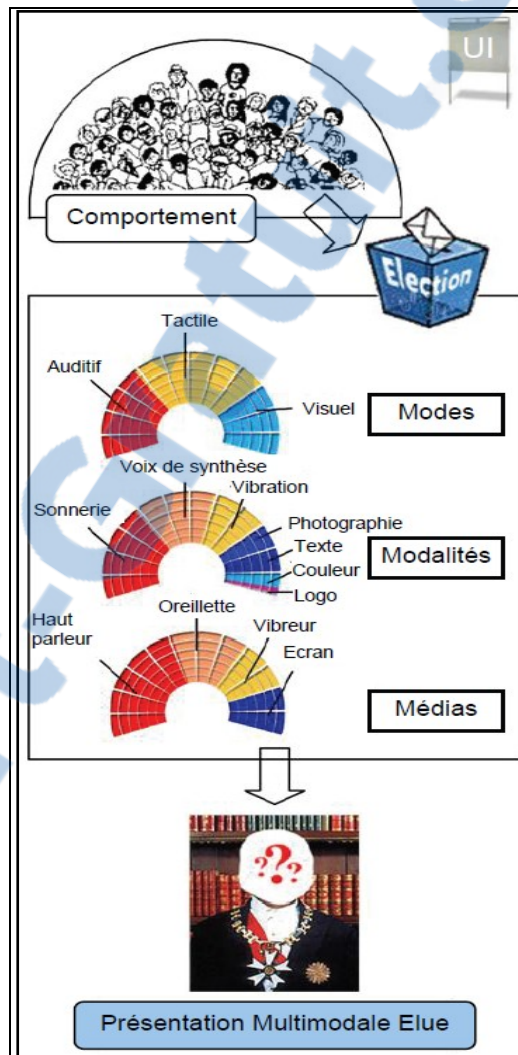


Figure 1.19 Le concept d'élection selon Rousseau.

Tirée de Rousseau (2006, P. 43)

La Figure 1.20, tirée de (Jacquet, 2006), montre un exemple d'arbre de pondération qui correspond à un profil d'un utilisateur malvoyant. Ce type d'utilisateur préfère les modes auditifs (pondération 1) aux modes visuels (pondération 0.2). Les attributs de ces modalités sont pondérés selon des fonctions continues ou discrètes. Par exemple, les valeurs de

l'attribut « langue » sont pondérées par une fonction discrète (EN 1.0, FR 0.4 et DE 0) et celles de l'attribut « vitesse » sont spécifiées par une fonction continue.

Finalement, la modalité appropriée est celle ayant obtenu la plus forte pondération. Il suffit de l'instancier en utilisant les valeurs d'attributs ayant elles aussi obtenu les plus fortes pondérations pour présenter l'information à l'utilisateur selon ses préférences.

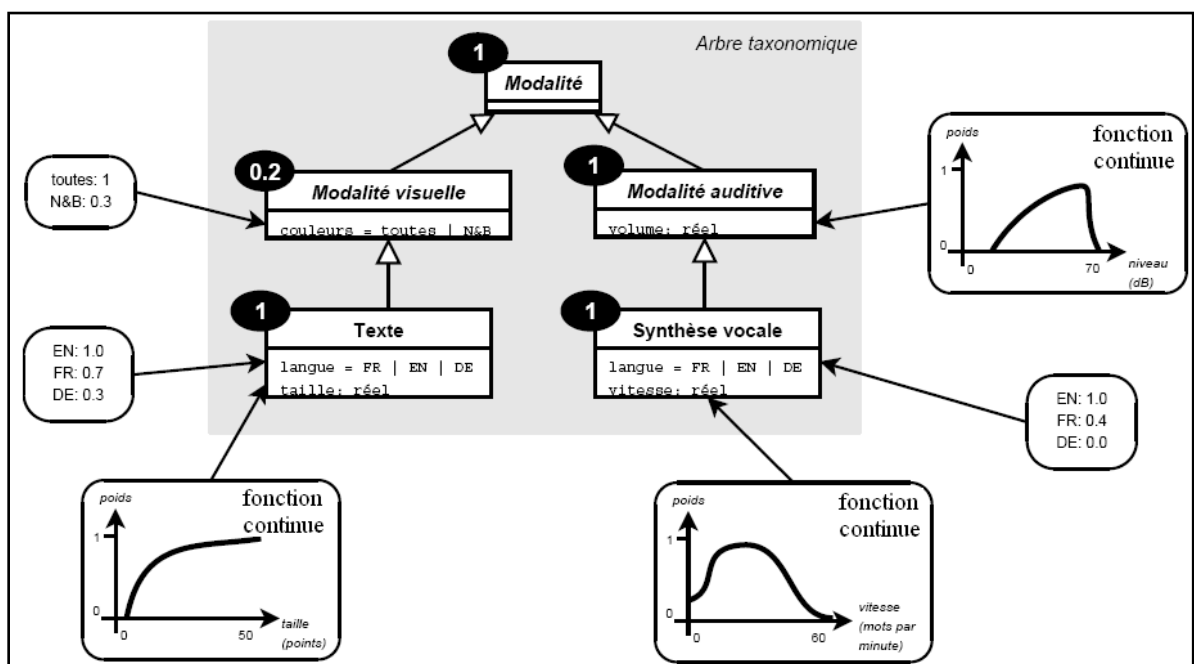


Figure 1.20 Exemple de profil (arbre de pondération).

Tirée de Jacquet (2006, P. 120)

En ce qui nous concerne, la sélection des modalités, des médias et/ou des formes de présentations est faite en fonction du contexte d'interaction actuel de l'utilisateur. Nous définissons également une fonction qui quantifie le degré de la convenance entre le contexte et la modalité appropriée. Ainsi, nous proposons un mécanisme d'automatisation pour la sélection des formes de présentations les plus appropriés au contexte de l'interaction de l'utilisateur. Il s'agit d'une technique d'apprentissage automatique qui permet de rendre la sélection des modalités, des médias et formes de présentations ainsi que leur adaptation automatique.

1.12 Architectures des systèmes intelligents

Dans la littérature, il existe plusieurs types de modèles d'architectures utilisés dans la conception des systèmes multimodaux. Nous présentons un résumé de ces architectures.

1.12.1 Seeheim

Seeheim est un modèle de type linguistique (grammaire du langage d'interaction) qui décompose l'interface en trois couches (voir la Figure 1.21). Chaque couche correspond à un niveau d'abstraction et possède un rôle spécifique (Pfaff, 1985; Pfaff et Hagen, 1993).

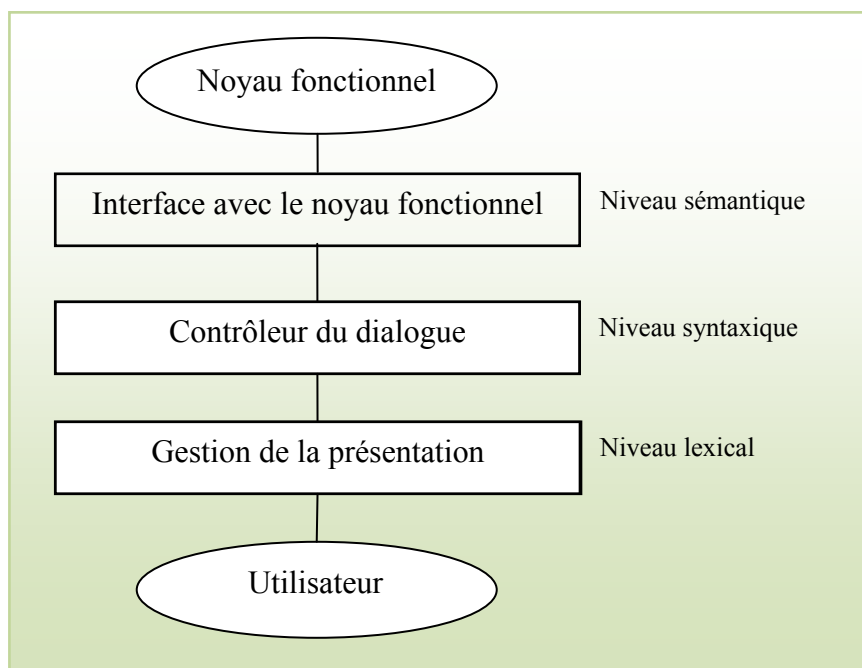


Figure 1.21 Le modèle de Seeheim.

Chaque couche est représentée par un module particulier :

- L'interface avec le noyau fonctionnel représente le niveau sémantique. Il est responsable de la traduction des informations en provenance du contrôleur du dialogue et inversement en informations compréhensibles dans le domaine de l'application.

- Le composant contrôleur du dialogue correspond au niveau syntaxique. Il est responsable de la gestion du dialogue. Il vérifie la cohérence structurelle des expressions.
- Le composant présentation correspond au niveau lexical. Il est responsable de la présentation des données reçues du contrôleur de dialogue via des médias de sortie. Il doit également traduire les données reçues, du média d'entrée, en unités lexicales et les transmettre au contrôleur de dialogue.

1.12.2 MVC

Le modèle MVC (Modèle-Vue-Contrôleur) a été mis au point dans le cadre du langage de programmation SmallTalk (Krasner et Pope, 1988).

Il organise l'interface homme-machine en trois facettes (voir la Figure 1.22) :

- Le modèle représente le comportement de l'application (traitements des données, etc.).
- La vue représente l'interface avec laquelle l'utilisateur interagit.
- Le contrôleur gère la gestion des événements pour mettre à jour la vue ou le modèle et les synchroniser.

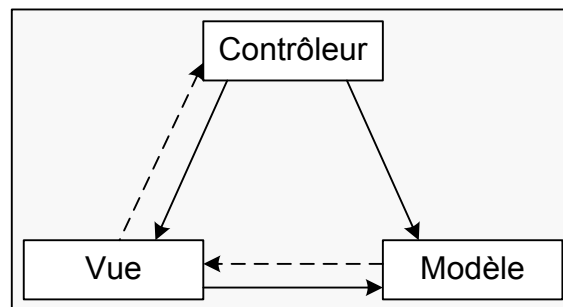


Figure 1.22 Le modèle MVC.

La Figure 1.22 résume les différentes interactions entre le modèle, la vue et le contrôleur. L'utilisateur fait une action; le contrôleur l'analyse puis demande au modèle approprié

d'effectuer les traitements nécessaires. Enfin, le contrôleur s'assure que la vue est actualisée pour afficher l'état du nouveau modèle.

MVC est un modèle qui a séparé la gestion des entrées de celle des sorties dans chacun de ses agents. Cependant, cette séparation, qui est considérée très avantageuse par rapport à la « souplesse », rend la mise en application parfois difficile. Effet, la plupart des entrées liées au contrôleur sont aussi liées directement à l'affichage. En absence d'un protocole de communication entre les composants, plusieurs outils ont fusionné la vue et le contrôleur en une seule entité.

1.12.3 ARCH

Le modèle ARCH est un résultat de l'atelier « UIMS Tool Developers Workshop 92 » (Bass et al., 1992). Ce modèle se fonde sur le modèle Seeheim avec les notions de noyau fonctionnel, de contrôleur de dialogue et de présentation (Figure 1.23). Il comprend cinq composants :

- Le noyau fonctionnel désigne la partie non interactive de l'application. Il définit ainsi les concepts propres au domaine de l'application.
- L'adaptateur de domaine représente un intermédiaire entre le contrôleur de dialogue et le noyau fonctionnel où il ajuste les différences entre les formalismes utilisés par ces deux composants pour la représentation des objets conceptuels. Il se charge aussi de la détection des erreurs sémantiques.
- Le contrôleur de dialogue gère le flux d'information entre le composant de présentation et adaptateur du domaine. Il est également responsable du maintien de la cohérence entre les différentes vues.
- Le composant de présentation joue le rôle de médiateur ou un composant d'amortisseur entre le contrôleur de dialogue et le composant d'interaction. Il maintient une représentation logique des objets conceptuels qui est indépendante de la plate-forme (i.e. Windows, OS., etc.).

- Le composant d'interaction est le responsable de la mise en œuvre des interactions physiques avec l'utilisateur. Il désigne l'ensemble des objets d'interactions d'une boîte à outils, ainsi que les communications avec les médias.

Le modèle ARCH utilise trois types d'objets décrivant les informations transmises entre les composants :

- Les objets du domaine contiennent des données utilisées par le noyau fonctionnel et par l'adaptateur du domaine.
- Les objets de présentation sont des objets d'interaction virtuels décrivant des événements produits par l'utilisateur et des données de présentation pour l'utilisateur, indépendamment des médias utilisés dans l'interaction.
- Les objets d'interaction implémentent des techniques d'interaction et de visualisation.

Le poids relatif des composants et la répartition de leurs fonctionnalités varient selon les types d'applications interactives et les choix impératifs fixés par les développeurs (Dragicevic, 2004).

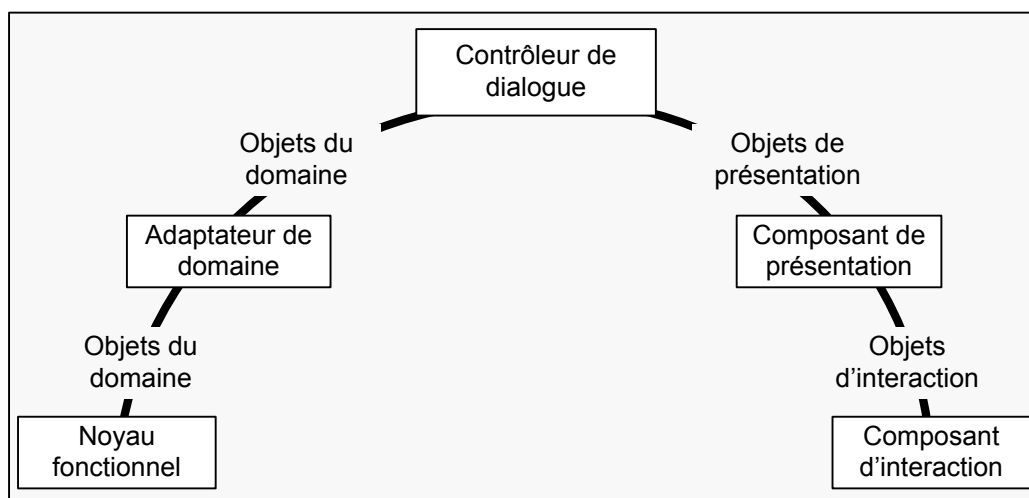


Figure 1.23 Les composants du modèle ARCH.

1.12.4 PAC

Le modèle PAC (Présentation, Abstraction, Contrôle) s'articule autour d'une entité de base nommée agent PAC (Coutaz, 1987). Cet agent est la brique de base d'une architecture construite sur ce modèle. Chaque agent comprend trois composants (voir figure 1.24):

- La présentation définit le comportement en entrée et en sortie de l'agent vis-à-vis de l'utilisateur. Elle regroupe les aspects vue et contrôle du modèle MVC.
- L'abstraction représente la partie sémantique de l'agent. Elle peut être considérée comme l'aspect modèle dans un MVC.
- Le contrôle regroupe les fonctionnalités d'interaction entre la présentation et l'abstraction, permet la communication avec les autres agents.

Le modèle PAC est récursif. Les agents peuvent donc être appliqués à tous les niveaux d'affinement et d'encapsulation d'un système interactif. Une application interactive peut ainsi être organisée de façon hiérarchique comme le montre la Figure 1.24. Nous pouvons noter que ce modèle unifie en quelque sorte les modèles à agents et les approches à couches de type Seeheim. Enfin, comme tous les systèmes multi-agents, PAC supporte le parallélisme et il reste un bon modèle pour les architectures distribuées.

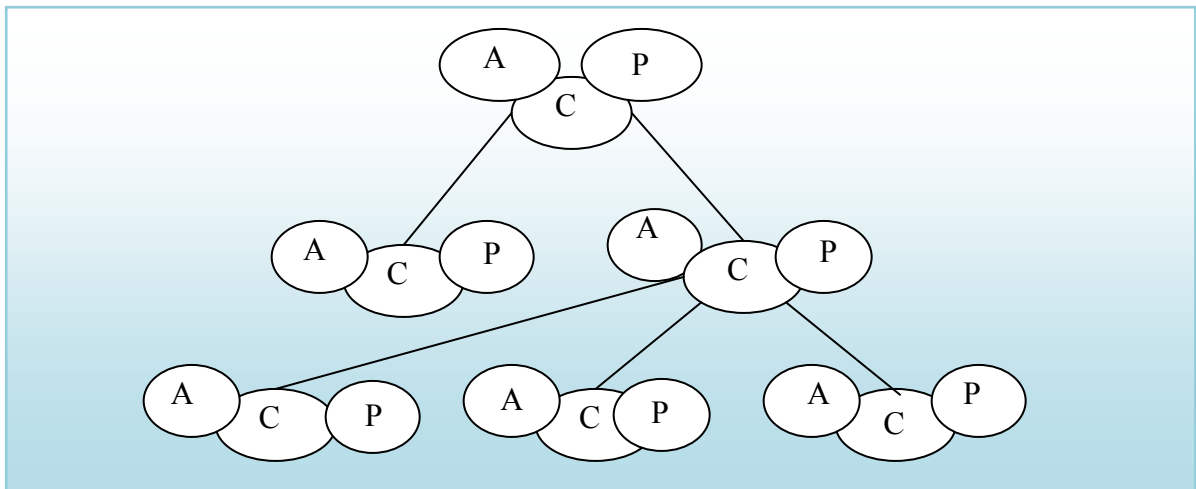


Figure 1.24 Une architecture PAC.

1.12.5 PAC-Amodeus

Cette architecture préconise la décomposition fonctionnelle d'ARCH et intègre les capacités d'affinement et de prise en compte du parallélisme de PAC. Comme le montre la Figure 1.25, cette architecture reprend les composants du modèle ARCH et décrit le contrôleur de dialogue avec une hiérarchie d'agents PAC (Nigay et Coutaz, 1993).

Nigay a trouvé que l'organisation générale du modèle ARCH est très similaire à celle d'un agent PAC à une décomposition fonctionnelle près. Elle a proposé de rassembler le tout en une architecture hybride: PAC-Amodeus (Nigay, 1994).

PAC-Amodeus possède une collection d'agents indépendants pour gérer le parallélisme et le traitement de l'information à plusieurs niveaux d'abstraction. Cet ensemble d'agents coopérant permet une meilleure répartition des tâches et donc un meilleur traitement.

Nigay et ses collaborateurs (Nigay, Coutaz et Salber, 1993) ont développé l'application MATIS (Multimodal Airline Travel Information System) basée sur cette architecture. Ils utilisent le modèle du creuset pour effectuer la fusion sur la base de trois critères : le temps, la complémentarité et le contexte du dialogue.

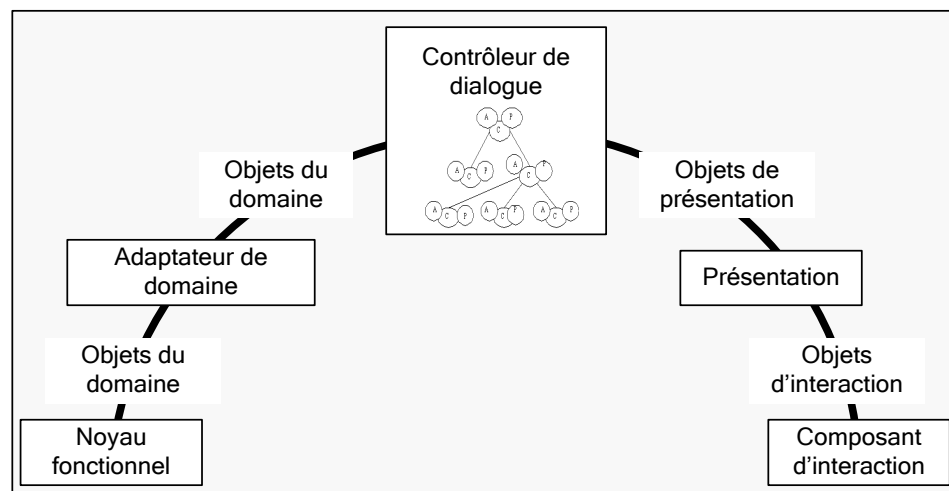


Figure 1.25 Le modèle PAC-Amodeus.

Malgré leurs apports, les modèles de référence tels que PAC-Amodeus n'ont pas décrit les techniques d'interaction. PAC-Amadeus ne décrit pas la distribution des techniques d'interaction entre le composant de présentation, le composant d'interaction, et le protocole de communication (Catinis, 1998).

1.12.6 L'architecture O. A. A.

L'idée de l'architecture O. A. A. (Open Agent Architecture) (Cohen et al., 1994) est à base d'agents. Elle fournit une structure pour construire des systèmes ouverts, distribués, extensibles, mobiles. Cette approche assure la collaboration et la communication entre des agents autonomes par un composant central « le facilitateur ». La Figure 1.26 montre un aperçu du prototype « office assistant » (Balvet, Grisvard et Bisson, 2002). Les principales caractéristiques de ce modèle se résument par (Julia, 1995; Moran et al., 1997) :

- **L'ouverture** : Ce modèle supporte des agents écrits dans divers langages de programmation (C, C++, Java, etc.) sur diverses plates-formes (Windows, Linux, OS, etc.).
- **La distribution** : Les agents d'une même application peuvent être distribués sur plusieurs machines (sur un réseau intra/internet). Cette distribution permet l'exécution des différents agents de l'application en même temps sur différentes machines. Les agents communiquent entre eux en ICL (Interagents Communication Language) et TCP (Transmission Control Protocol).
- **L'extensibilité** : Des agents peuvent être ajoutés ou retirés du système en fonction du contexte, de l'activité, selon les besoins de l'utilisateur ou des ressources disponibles, etc. Ces ajouts ou retraites apportent des changements à l'application pendant que celle-ci est en cours d'exécution. Ainsi, l'architecture supporte des changements dynamiques en offrant ainsi les services de nouveaux agents ou arrêtant les services des agents retirés.
- **La collaboration** : la tâche est réalisée grâce à la coopération des différents agents concernés. L'interface fonctionne par des agents autonomes qui coopèrent entre eux pour

réaliser une activité. À noter que l'utilisateur lui-même est vu comme un agent du système.

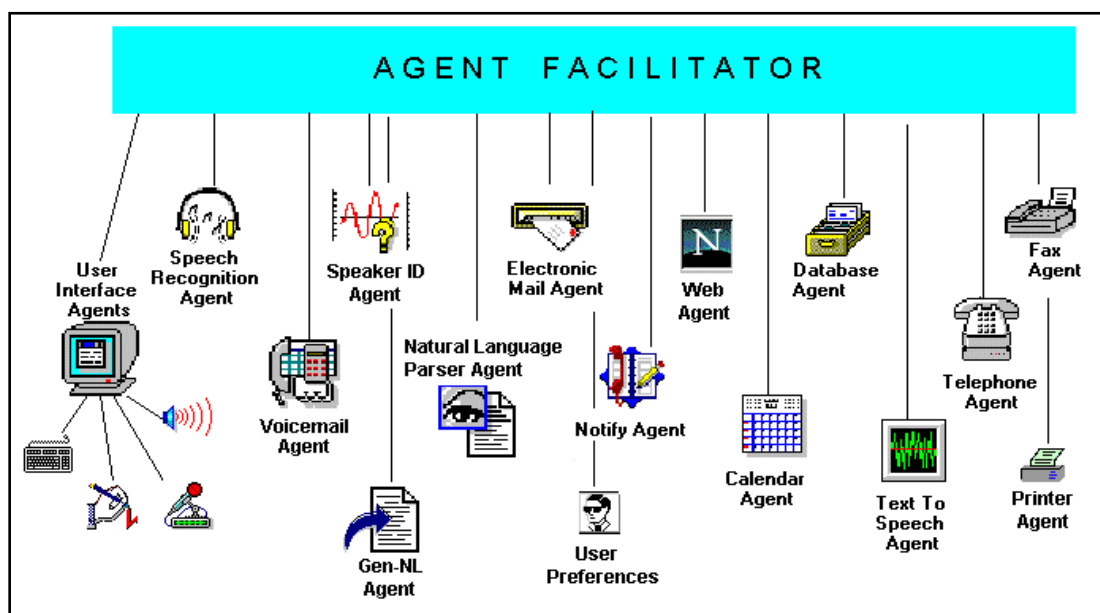


Figure 1.26 Un exemple de l'architecture O.A.A, le modèle communauté d'agents formant le «office assistant».

Tirée de Balvet (2002, P. 94)

- **La mobilité** : résulte du fait que l'architecture peut être répartie. Il est ainsi possible d'assigner la charge des traitements (les calculs lourds) à des machines puissantes (serveur, cluster, etc.) et de laisser l'affichage au terminal de l'utilisateur qui peut être mobile tel qu'un PDA (Personal Digital Assistant), un téléphone cellulaire, etc.
- **La réutilisation** : les agents sont autonomes. Il est possible de les réutiliser et de les intégrer à la communauté des agents d'un autre système. Cette approche permet de réduire l'effort nécessaire lors du développement d'une nouvelle application.

Malgré les avantages d'un tel modèle d'architecture, le facilitateur peut néanmoins être la cause d'un problème de surcharge s'il reçoit beaucoup d'informations à traiter simultanément.

1.12.7 Le modèle à couches

Dans ce modèle, l'application est structurée en plusieurs couches dont chacune a un rôle spécifique. Dans cette approche, chaque couche propose un ensemble de services qu'elle offre à la couche immédiatement supérieure.

Le découpage du système en plusieurs couches permet de faire évoluer les fonctionnalités d'une couche sans provoquer des changements dans les autres couches.

Ce modèle d'architecture a été surtout utilisé dans les applications réseaux. De plus, cette approche peut être utilisée en conjonction avec les autres modèles d'architectures tel que le modèle multi-agents. Cette conjonction contribue à dépasser les contraintes du système d'exploitation et les contraintes du matériel et des pilotes de périphériques (Catinis, 1998). La Figure 1.27 présente un exemple d'une architecture multicouche dont les fonctionnalités de chaque couche sont assurées par des agents.

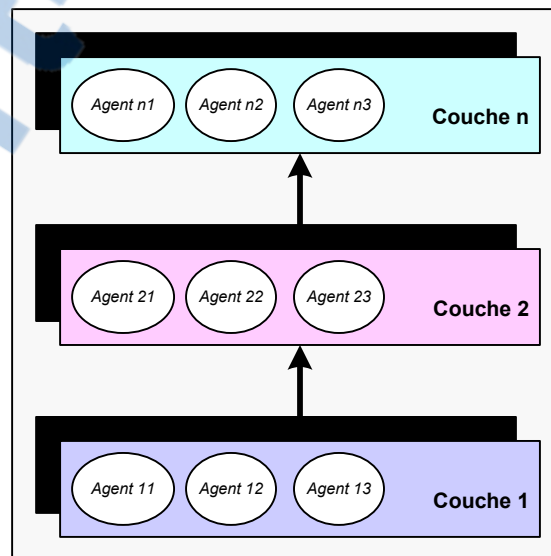


Figure 1.27 Un exemple de l'architecture multicouche en conjonction avec le modèle multi-agent.

1.12.8 Choix de l'architecture

Nous constatons que les modèles de type linguistique ne sont pas adaptés pour concevoir un système multimodal « intelligent ». En effet, ce genre de modèle est destiné, à l'origine, pour concevoir des interfaces langagières. Cela rend son adaptation aux interfaces modernes assez difficile. En ce qui concerne les modèles à agents (PAC –Amodeus et MVC), ceux-ci permettent le traitement parallèle mais ne décrivent pas la distribution des techniques d'interaction entre le composant de présentation, le composant d'interaction, leurs rôles et le protocole de communication (Catinis, 1998). Aussi, faire fonctionner ce genre de système, sur un terminal à faibles capacités, n'est pas possible. Cependant, une architecture telle que « OAA » permet d'exécuter les applications sur des terminaux mobiles. Ce modèle est plus avantageux surtout que la mobilité est un aspect important des systèmes multimodaux adaptatifs et a fait le sujet de plusieurs études (Oviatt, 2000; Rousseau et al., 2006; Serrano et al., 2006; Van Belle et al., 2001).

En ce qui nous concerne, nous désirons concevoir un système multimodal intelligent qui permet à l'utilisateur mobile d'accéder à ses données en tenant compte des ressources du système et des capacités de l'utilisateur. Nous avons choisi d'opter pour une architecture multicouche pour concevoir notre système multi-agent (SMA) parce qu'une telle approche est plus pertinente pour un système qui doit prendre en considération le contexte de l'interaction lors de la présentation des données aux utilisateurs. Ce genre de système est facilement modifiable pour correspondre aux besoins des utilisateurs. En effet, les agents possèdent des propriétés qui sont très utiles et exploitées dans les SMA. L'autonomie, la spécialisation et le parallélisme des agents rend possible la conception des systèmes qui traitent les informations perçues sur plusieurs canaux. Chaque agent (chaque média est représenté par un agent) ne réagit qu'aux informations qui lui sont destinées et, ceci, indépendamment de ce qui peut être reçu ailleurs. Il est ainsi possible de réaliser la fusion multimodale des informations via des agents spécialisés (Djenidi et al., 2003). Les avantages les plus visibles de ce modèle se manifestent par la rapidité de réaction du système et par sa

capacité d'intégrer des nouveaux composants et de continuer à fonctionner, même si une partie du système n'est plus active.

1.13 Interfaces multimodales pour les non-voyants

Le problème principal de la conversion des données visuelles vers des données non visuelles équivalentes est parfois complexe en raison de la grande quantité d'informations à présenter. En effet, une grande quantité d'informations peut être assez facilement présentée visuellement (exemple, une image affichée sur l'écran). Cependant, les formes non visuelles ne peuvent que présenter une faible quantité d'informations simultanément. Pour pallier ces difficultés, il est judicieux d'avoir un système qui permet de présenter les données à l'utilisateur via plusieurs modalités (auditive, tactile). Le système multimodal, riche de ces modalités et apte à les combiner, peut offrir une solution intéressante pour combler le vide laissé par l'absence du mode visuel.

En s'appuyant sur les remarques précédentes, notre approche consiste donc à combiner plusieurs modalités :

- En entrée (de l'utilisateur vers la machine) : exploiter la voix (système de reconnaissance vocale), le clavier traditionnel, terminal braille, un Overlay, etc.
- En sortie (de la machine vers l'utilisateur) : avoir un retour tactile (braille, mise en relief), auditif (parole et / ou sons non parlés), haptique (souris à retour d'effort).

Il serait intéressant d'exploiter simultanément ces modalités dans un but de complémentarité et de redondance. Certaines informations vont être plus efficacement transmises via une seule modalité, d'autres nécessiteront plusieurs modalités pour être facilement accessibles, chaque modalité transmettant tout ou en partie de l'information. La définition de nouveaux scénarios d'interaction non visuels multimodaux peut constituer un apport considérable pour l'augmentation de l'efficacité, la convivialité, la facilité d'apprentissage et l'utilisation de l'interface non visuelle.

Dans MATHS (Stevens, 1996; Stevens, Edwards et Harling, 1997) par exemple, la multimodalité correspond à l'utilisation de la synthèse vocale (la notion de sortie audio spatialisée), de sons non parlés, du braille (en utilisant la notation de Stuttgart (Schweikhardt, 1987)) et d'un écran (pour les utilisateurs mal voyants). La saisie de textes ou d'expressions mathématiques se fait en utilisant un clavier traditionnel, un système de reconnaissance vocale et le braille.

Dans MEDITOR (Bellik, 1995), l'utilisateur a toujours l'impression d'agir directement sur les objets de l'interface, et perçoit instantanément, par des messages de rétroaction (feedback) tactiles et/ou sonores, les modifications que subissent ces objets. L'utilisateur lit en touchant les caractères brailles affichés par le terminal et il peut simultanément énoncer des commandes de navigation en utilisant un système de reconnaissance vocale (page suivante, début du texte, etc.). Les informations supplémentaires peuvent être communiquées soit par des bips, soit par la synthèse vocale ou par le point 8 des cellules braille, ou bien encore par la combinaison de ces trois éléments.

Dans EasyMath (Podevin, 2002), le système est conçu pour être capable d'interagir avec l'utilisateur via le clavier traditionnel, un overlay et la reconnaissance vocale en entrée ; le braille, le relief et la synthèse vocale sont exploitées en sortie. En terme de multimodalité, l'information présentée par différentes modalités est donc redondante (braille et lecture audio), en entrée, elle est complémentaire (clic et parole). À noter que les commandes multimodales n'ont pas été testées parce qu'elles n'étaient pas prêtes lors des tests menés. Le principe de l'interaction est le suivant : l'utilisateur lit la formule avec le doigt, puis s'il désire obtenir la signification de ce qu'il touche, il peut cliquer sur le caractère pour l'entendre. À l'aide des commandes vocales, l'utilisateur peut lire les cases suivantes, précédentes, etc. L'utilisateur peut demander la lecture intégrale de l'expression mathématique et il peut l'interrompre à n'importe quel instant.

Les exemples de systèmes multimodaux décrits (et d'autres) ont permis de constater que la multimodalité peut augmenter considérablement l'efficacité et la facilité d'apprentissage d'une interface non visuelle.

1.14 Résumé : les interactions multimodales

La multimodalité a un intérêt évident dans le cadre des applications destinées aux personnes déficientes visuelles. La combinaison judicieuse de plusieurs modalités permet généralement d'obtenir une interaction plus riche que chacune d'elles prise séparément. De plus, le fait de pouvoir utiliser plusieurs méthodes de présentation simultanément peut permettre à un professeur de mathématiques « voyant » d'enseigner à un étudiant non-voyant en lisant les formules en noir sur l'écran, tout en donnant la possibilité à l'étudiant de les lire en braille ou selon une autre méthode.

Cette multimodalité ouvre le champ à une interaction homme-machine « intelligente » :

En entrée, la machine doit être dotée de connaissances et de capacités de raisonnement sur ces connaissances qui lui permettront d'une part, de faciliter et de fiabiliser les entrées de l'utilisateur et d'autre part, de prendre en considération non seulement les informations reçues mais également le contexte de l'interaction et la ou les modalités à travers lesquelles elles ont été transmises. Ceci permettra à la machine, par exemple, de déterminer les modalités optimales pour un utilisateur et de s'adapter en fonction de ses préférences. Un système multimodal doit permettre à l'utilisateur d'utiliser le ou les modes d'interaction les mieux adaptés à ses préférences, à son niveau d'expertise et à la nature de la tâche à accomplir (Catinis, 1998).

En sortie, la multimodalité a ajouté une nouvelle dimension à la problématique de l'interaction Homme-Machine. En effet, dans les interfaces actuelles, toutes les formes de sorties sont prédéfinies avec leurs modalités. Le problème qui se pose aujourd'hui est de doter la machine de capacités intelligentes qui lui permettront d'accepter de nouvelles formes de sorties de manière dynamique, sans avoir à changer l'architecture de l'application. Il ne faut pas seulement présenter l'information à l'utilisateur, mais aussi pouvoir la présenter via la meilleure combinaison de modalités en fonction du contexte courant. Cela implique que le système doit être capable de connaître les moyens de communication (les modalités) les plus

efficaces pour chaque utilisateur et chaque type de données manipulées et de pouvoir sélectionner de façon pertinente entre les médias et les formes de présentation disponibles.

1.15 Conclusion du chapitre 1

Ce chapitre a permis de donner une vue d'ensemble des recherches menées dans le domaine de la présentation d'expressions mathématiques aux non-voyants. Après avoir posé quelques définitions et présenté les méthodes d'accès aux informations utilisées par les non-voyants, nous avons présenté les concepts de base de la multimodalité, les taxonomies d'intégration des modalités, les problématiques spécifiques à ce domaine de recherche et les architectures des systèmes intelligents.

À la suite de cette étude, nous avons abordé les problèmes auxquels les non-voyants font face à travers l'analyse de quelques systèmes existants. Nous avons souligné l'importance de la multimodalité dans les interfaces non visuelles. Ce chapitre s'est conclu sur notre problématique initiale, à savoir la conception d'un système intelligent qui s'adapte au contexte de l'interaction. Cette vision nous a conduits vers la conception d'un système multi-agent apte à détecter et traiter le contexte de l'interaction. Les systèmes existants pour l'accès aux mathématiques se focalisent sur des approches traditionnelles (une seule forme de présentation est possible, elle est prédéfinie), oubliant bien souvent l'aspect dynamique du système et négligeant le contexte de l'interaction.

Enfin, l'adaptabilité d'un système signifie qu'il sera configurable par l'utilisateur et permettra d'utiliser la plupart des approches existantes de façon séparée ou groupée. Son adaptativité viendra de sa capacité à s'auto configurer en fonction de l'évolution du niveau de l'utilisateur, de ses performances et habitudes. Dans ce contexte, nos recherches sont focalisées sur la conception d'un système capable de se servir de l'ensemble des approches et d'offrir à l'utilisateur des interfaces multimodales qui pourront, à terme, se configurer dynamiquement (Djenidi et al., 2003) à partir de la connaissance du profil de l'utilisateur et de la nature de l'expression.

Nous nous attarderons maintenant à organiser notre problématique en commençant par proposer un système multi-agent pour la présentation d'expressions mathématiques à des utilisateurs non-voyants. Ce système prend en compte le contexte de l'interaction ainsi que la nature de l'expression. Le chapitre suivant est donc consacré à définir l'architecture et à indiquer les concepts employés pour réaliser un tel système.

CHAPITRE 2

INFRASTRUCTURE OF AN ADAPTIVE MULTI-AGENT SYSTEM FOR PRESENTATION OF MATHEMATICAL EXPRESSION TO VISUALLY-IMPAIRED USERS

Ali Awde¹, Chakib Tadj¹, Yacine Bellik²

¹Département de Génie Électrique, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

²Laboratoire LIMSI-CNRS, Université de Paris-Sud
B.P. 133, 91043 Orsay, France

Ce chapitre a été accepté pour une publication dans la revue International Society for Advanced Science and Technology (ISAST) Transactions on Intelligent Systems, To be published in 2009.

Résumé

L'infrastructure de notre système adaptatif vise à présenter, de manière adaptée, les expressions mathématiques pour les non-voyants. Sa conception est basée sur un système multi-agent qui détermine la forme de présentation qui convient à l'utilisateur en prenant compte le contexte de l'interaction (c.-à-d. l'utilisateur, son environnement et son système) ainsi que la nature de l'expression.

Le système proposé est un système multi-agent, réparti en couches multiples. Le découpage du système en couches permet de modifier une couche sans affecter les autres couches. Le système est conçu pour être adaptatif et tolérant aux fautes. Il est également capable de s'adapter aux besoins de l'utilisateur et aux changements affectant le contexte d'interaction (défaillance matérielle ou logicielle). Dans le cas où c'est un média qui tombe en panne, notre système est capable de remplacer le média défectueux par un nouveau (si son remplaçant est disponible). Sinon, le système régénère une nouvelle modalité et une nouvelle

forme de présentation selon la nouvelle situation. Cela permet à l'utilisateur de continuer son activité sans aucune intervention explicite de sa part.

Les simulations de notre SMA ont été effectuées sur la plate forme multi-agent JADE (Java Agent Development Framework). Ce travail représente notre contribution aux recherches dans le domaine de l'aide aux utilisateurs non-voyants. Notre objectif est de rendre l'accès aux mathématiques plus simple pour les non-voyants et de les aider à acquérir une certaine autonomie.

Mots clés: Système adaptatif, système tolérant aux fautes, mathématiques pour les non-voyants, système multi-agent.

Abstract

The infrastructure of our adaptive system is aimed at correctly presenting a mathematical expression to visually impaired users. Its design is based on a multi-agent system that determines the appropriate presentation format based on the given interaction context (i.e. combined user, environment and system contexts) and the expression complexity as well as the user preferences. The architecture of our infrastructure is layered, thus encapsulating the components of the various layers. The system design is intended to be adaptive, fault tolerant and is capable of self-adaptation under varying conditions (e.g. missing or defective components). In case of failure of media device, our system is capable of replacing the faulty component with a new one (if a replacement is available). If replacement is not possible, the system re-determines the new modality and presentation format apt for the new configuration. The human intervention is greatly reduced in our system as it is capable of self-configuration, and learning. In our work, agent communication simulation has been carried out on the Java Agent Development Framework (JADE) platform. This work is an original contribution to the ongoing research in helping the visually-impaired users to become autonomous in using the computing system. Our aim is to improve the computing productivity of a visually-impaired user.

Index Terms: Adaptive system, fault-tolerant system, mathematics for visually-impaired users, multi-agent system.

2.1 Introduction

Mathematics is a fundamental foundation of science. Understanding science is impossible without knowing mathematics. Mathematics for visually-impaired users, however, is a challenging task due to the following reasons: First, the visual mathematical representation is bi-dimensional and the interpretation of a mathematical expression is related to one's knowledge of the expression's individual spatial components. Second, the conversion of a multi-dimensional structure to a non-visual representation is a difficult problem. For example, the representation of a mathematical expression in Braille requires supplementary information to denote some components in order for the blind users to read the expressions easily. Also, the conversion of mathematics into an audio format is often ambiguous. Third, the vocabulary terms used by sighted people in a mathematical document are quite large compared with the amount of data accessible by a visually-impaired user. For example, in a standard 6-dot Braille, we can encode 64 characters. This number of symbols is, however, insufficient to represent all frequently used mathematical symbols. Also, large number of symbols is a big challenge to blind users. For instance, Braille characters are often embossed into a paper which is a static media; hence, user will not be able to manipulate the data easily. Indeed, some flexible methodologies are needed to allow blind users to access and navigate terms in a mathematical expression easily.

This paper presents the challenges that underlie in designing such infrastructure that provides presentation of mathematical expressions to visually impaired users, and how we address the problem cited above by proposing an intelligent multimodal computing system that interacts with the user, capable of choosing modalities and media devices based on a given interaction context. The format for the presentation of a mathematical expression is selected based on available media devices, user's preferences and context of the mathematical expression. The proposed infrastructure is a multi-agent system. The design of this multi-layered

infrastructure enables every layer's calculation and decision making be hidden from other layers. In this way, the possible propagation of ripple effect during any stage of software life cycle becomes limited and restricted only within the boundaries of the concerned layer. Various layers communicate among themselves via parameters passing. In this work, we discuss the design of each agent and its responsibilities. Also, we present our fault tolerant system design and a JADE simulation of agent communication.

This work is our contribution to make mathematics more accessible to visually-impaired users. It is aimed at providing some autonomy to blind users when dealing with mathematical expressions. The rest of this paper is structured as follows. Section 2.2 provides a brief review of the state of the art related to our work; Section 2.3 lists down the technical challenges in this work and essays our approach to address each one of them. Section 2.4 presents the individual components of our adaptive multimodal computing system. In Section 2.5, the principles used for the knowledge acquisition of our learning agent are discussed. Examples are provided in Section 2.6 while Section 2.7 presents some formal specifications related to the functionalities of the system. The future works and conclusion are presented in Section 2.8.

2.2 Review of the State of the Art

To a visually-impaired user, understanding mathematical expression requires repeated passage over the expression, sometimes skipping some secondary information, only to revert back to it again and again until the user fully grasps the expression. A complicated task like this is detailed in (Stöger, Miesenberger et Batusic, 2004). Some tools, however, have been developed to lessen the complexity of performing similar task, among them being the MathTalk (Edwards et Stevens, 1994), Maths (Cahill et al., 1995), DotsPlus (Preddy et al., 1997), EasyMath (Podevin, 2002), and AudioMath (Ferreira et Freitas, 2004; 2005). MathTalk and Maths convert a standard expression into audio information. In Maths, the user can read, write and manipulate mathematics using a multimedia interface containing speech, Braille and audio. Raman developed Aster (Raman, 1998), a program that takes a Latex

document and reads it loud using several audio dimensions that make up the different components of the expression. VICKIE (Moço et Archambault, 2003a) and BraMaNet (Schwebel et Goiffon, 2005) are transcription tools that convert mathematical document (written in Latex, MathML, HTML, etc.) to Braille representation. DotsPlus is a tactile method of printing documents that incorporates both Braille and graphic symbols (e.g. \prod , \sum , etc.) In EasyMath, regardless of using Braille or overlay keyboard, its main focus is to keep the general structure of mathematical expressions intact.

None of these tools, however, is complete. Studies have been made for evaluating these tools based on users' needs (Garlini et Fogarolo, 2003; Podevin, 2002). Results indicate that users are neither independent nor able to do their homework (in case of students) without the help of sighted people. Indeed, each tool has its own set of usage limitations. For example, Aster transforms only a LaTeX document into an output suitable for speech while AudioMath transforms only a MathML input document into a speech output. Our approach, therefore, is to get the strength of each tool, integrate each one of them into our work in order to build a system that (1) broadens the limits of utilization, (2) provides the user with opportunities to access as many document types as possible, and (3) presents data output in as many suitable formats as possible after considering user situation and the special symbols within the expression. This work is an essential contribution because we offer all types of data presentation formats yet requires minimum intervention from the user.

HOMERE (Lécuyer et al., 2003) is a multimodal system that allows visually-impaired users to use haptic/touch and audio modalities to explore and navigate virtual environments. In comparison, our approach is better because there are no pre-defined input-output modalities; the selected modalities are chosen according to their suitability to user's context. To visually-impaired users, multi-modality is even more important as it provides them equal opportunities to use informatics like everybody else. In determining the appropriate modality, the user situation plays an important role. In our work, the notion of user context includes additional handicaps and user preferences on the priority rankings and parameter settings of media devices and presentation formats.

An agent is some software that senses its environment and is capable of reaction, proactivity, and social interaction. A group of agents in a system forms a Multi-Agent System (MAS) (Wooldridge, 2002). Agents and MAS (Ferber, 1995; Weiss, 1999) have been widely used in many applications, from relatively small systems such as email filters up to large, open, complex, mission-critical systems such as air traffic control (Jennings et Wooldridge, 1998). Generally, it is preferred over traditional techniques (i.e. functional or object-oriented programming) because the latter is inadequate in developing tools that react on environment events. Significant works on MAS for visually impaired include (Awde, Tadj et Bellik, 2008; Bourbakis et Kavraki, 2001). For example, Tyflos (Bourbakis et Kavraki, 2001) could help a visually impaired user to be partially independent and able to walk and work in a 3-D dynamic environment. Our work, in contrast, uses agents to detect user context, and other data in order to assist the system to determine the media and modalities that are appropriate for the user.

2.3 Technical Challenges

In this section, we resume the problems in designing a system that will present a mathematical expression to visually impaired users, pose specific technical challenges that need to be solved and describe our approach to address those challenges.

In our proposed infrastructure, we envision a system that is rich in media devices and modality selection, data formats, techniques in converting one data format to another, and an adaptive interface that allows user to manipulate mathematical data. To design such a system, a solution must address key requirements, cited below:

Requirement 1: Provide a multiagent system that coordinates all its components in an orderly manner, providing autonomy to the user and is able to adapt automatically to a given interaction context. How do we design a multi-layered, multiagent system that satisfies the system requirement? What mechanisms must be adopted to make the system tolerant from faults?

Requirement 2: Provide a mechanism that allows selecting the appropriate media device supporting the selected modality. How do we design a system that satisfies the user preferences? How the media device will be detected? Which media device will be activated to support a specific modality?

Requirement 3: Provide an infrastructure for analysis and conversion of a mathematical expression, embedded within a document (in MathML format) into its corresponding encoded format and then into its presentation format. How to convert an expression written in MathML format into an expression in encoded format and into an expression using a presentation format such as the Braille, DotsPlus, EasyMath and audio?

Requirement 4: Provide a mechanism that allows visually impaired users to manipulate terms in a mathematical expression. How do we design a system component that allows the user to add, modify and delete mathematical terms, and to search the expression for a term in random manner?

The rest of this paper addresses the technical challenges by providing specific solutions to the system requirements cited above.

2.4 The Components of an Adaptive Multimodal Computing System

2.4.1 Our Adaptive Multimodal Computing System for Mathematical Expression Presentation to Visually Impaired Users

Figure 2.1 shows the layered view of an adaptive multimodal computing system that presents mathematical expressions to visually-impaired users. The layers and their roles are as follows: (1) *Physical Layer* – contains all the physical entities of the system, including devices and sensors. The raw data from this layer are sampled and interpreted and forms the current instance of interaction context. (2) The *Context Gathering Layer* – here, the interaction context (of tuple $\langle \text{user}, \text{system}, \text{environment} \rangle$) is detected; (3) The *Control and Monitoring Layer* – it controls the system, coordinating the detection of user's interaction

context, the mathematical expression, its presentation and/or manipulation; (4) The *Data Analysis Layer* – here, the presentation format of the mathematical expression is selected based on available resources and user’s situation; (5) The *Data Access Layer* – in this layer, mathematical expression may be searched or edited by the user; (6) The *Presentation Layer* – here, the mathematical expression is presented through an optimal presentation format.

Figure 2.2 shows our model of a multi-agent adaptive multimodal system for visually-impaired users. The agents’ functionalities in the system’s layers are detailed in the next sections.

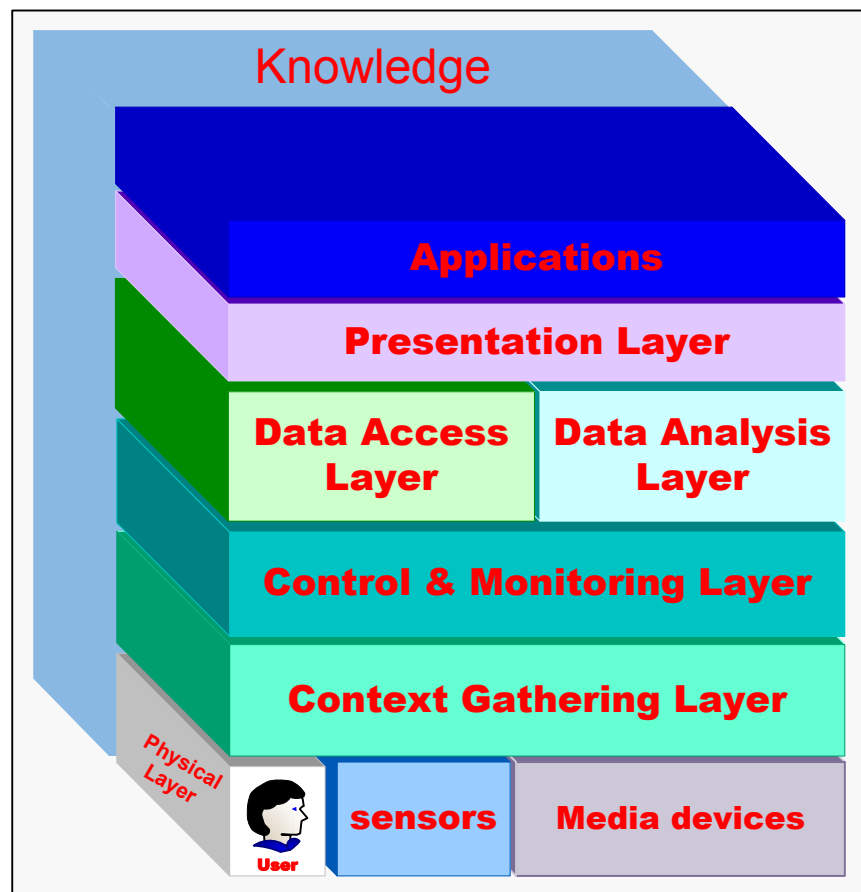


Figure 2.1 The architectural abstraction of a generic MM computing system for visually-impaired users.

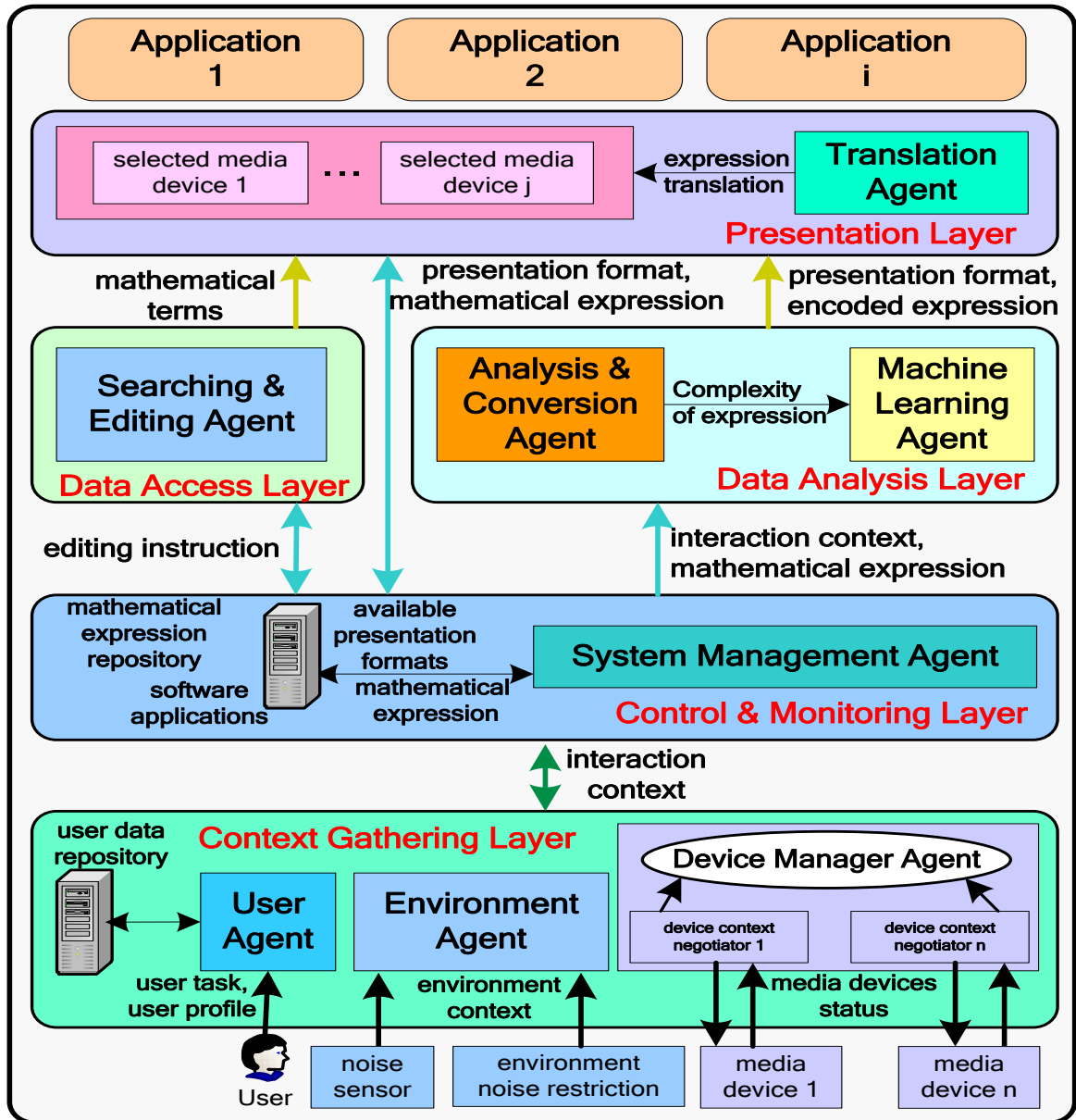


Figure 2.2 Architectural layer view of the multimodal computing system.

2.4.2 Modality and Media

In our work, we adopt the concepts of media and modality that are defined by Bellik in (Bellik, 1995).

1. **Modality** is defined by the information structure as it is perceived by the user (e.g. text, speech, Braille, etc.).
2. **Media** is defined as a device used to acquire or deliver information or data (e.g. screen, terminal Braille, mouse, keyboard, etc.).

Here, *Vocal* and *Tactile* modalities are possible since we address visually impaired users. Also, in general, interaction is possible if there exists at least one modality for data input and at least one modality for data output. Given a modality set $M = \{V_{in}, T_{in}, V_{out}, T_{out}\}$ wherein $V_{in} = \text{vocal input}$, $V_{out} = \text{vocal output}$, $T_{in} = \text{tactile input}$ and $T_{out} = \text{tactile output}$ then interaction is possible under the following condition:

$$\mathbf{Interaction\ Possible} = (V_{in} \vee T_{in}) \wedge (V_{out} \vee T_{out}) \quad (2.1)$$

where the symbols \wedge and \vee denote logical AND and OR, respectively. There are usually more than one media that support a specific modality (see section 2.2.3).

2.4.3 The Context Gathering Layer

In this layer, the user's interaction context is identified. There are three agents that obtain the context of the *user*, the *environment*, and the *system* which collectively form the interaction context. Here, we present briefly the interaction context, for more details see (Awde et al., 2009).

The User Context

In this work, the user context (UC) is a function of user profile (including any handicap) and preferences. The *user agent* (UA) detects the user's profile and preferences. We have a *user data repository* where user's task (i.e. the mathematical expression) is stored. In such repository, there is a *user profile* (UP) for the user, which contains, among others, the user's username, password, his computing devices and identifications (i.e. IP addresses) and his

special needs (i.e. handicap). This information is useful for determining the suitable modalities. For example, being mute prevents the user from using vocal input modality.

The Environment Context

The *environment's context* (EC) detected by Environment Agent (*EnvA*), is the assessment of a user's workplace condition. In this work, EC is based on the following parameters: (1) the workplace's *noise level* – identifies if it is quiet/acceptable or noisy, and (2) the *noise level restriction* – identifies whether a workplace imposes mandatory silence or not. For example: in a library where silence is required, sound producing media (e.g. speaker) needs to be muted or deactivated.

The noise level is interpreted by EnvA from the sampled raw data of a sensor. In our work, *50 dB or less* is considered “*acceptable*” while *51 dB or more* is considered “*noisy*”.

For environment noise restriction, we have a database of pre-defined places (e.g. library, park) and their associated noise restrictions (e.g. library: silence required, park: silence optional). User can update and modify some database records.

The System Context

In this work, the *system context* (SC) implies the user's computing device and the available media devices. SC is managed by the *Device Manager Agent* (DMA). The computing device (e.g. PC, laptop, PDA, cellular phone) affects the modality selection. For example, using a PDA or cell phone prevents user from using tactile input or output modality. On the other hand, some of the most commonly-used media devices suitable for blind users are: (i) *Keyboard*; (ii) *Microphone*; (iii) *Speech Recognition* (iv) *Speech Synthesis* (v) *Speaker*; (vi) *Headset*; (vii) *Braille Terminal* (viii) *Overlay or Concept Keyboard* (ix) *Tactile Printer or Embosser*.

Every media device has its own *device context negotiator* (DCN). A DCN is an agent that detects the media device's context; it is a link between the actual media device and the DMA. Every DCN has the following attributes (see Figure 2.3): its *name*, *class*, *characteristics*, *status*, and *confidentiality*. "*Name*" identifies the media device it manages (e.g. "Braille context negotiator" detects the context of a Braille terminal). "*Class*" identifies its form of modality. "*Characteristics*" identify the unique features of the device. "*Status*" identifies if the device is on, off, sleep or disabled. "*Confidentiality*" is the perceived reliability of the device and is denoted as high, medium or low.

Context negotiator	Device 1	Device 2	Device 3
Name	<i>Braille Terminal</i>	<i>Speech recognition</i>	<i>Speech synthesis</i>
Class	<i>Tactile input, Tactile output</i>	<i>Vocal input</i>	<i>Vocal output</i>
Characteristics	<i>no. of characters per line</i>	<i>language no. maximum of characters detected /min</i>	<i>language; age; gender; no. of characters /min</i>
Status	<i>On Off Sleep Disabled</i>	<i>On Off Sleep Disabled</i>	<i>On Off Sleep Disabled</i>
Confidentiality	<i>High Medium Low</i>	<i>High Medium Low</i>	<i>High Medium Low</i>

Figure 2.3 Attributes of device context negotiator for Braille terminal, speech recognition and synthesis.

2.4.4 The Control and Monitoring Layer

System Management Agent (SMA) monitors and reports on the system configuration and application activity. When a fault occurs, SMA should deal with errors and find solution to keep system working (see section 2.5).

Upon detection of interaction context, the control and monitoring layer via its SMA examines the mathematical expression in the user's task. It determines the available presentation

formats by sensing the presence of conversion software in the computer. It also determines if the current mathematical expression and interaction context are already listed in the system's *mathematical expression repository* (MER). If so, then the expression is sent to Presentation Layer for its presentation. Otherwise, the system must learn how to present this expression by sending all the available information to the Data Analysis Layer.

The MER is a *private* database that keeps all mathematical expressions that have already been encountered. All expressions are stored in a tabular form; each table entry contains (1) the original expression in MathML format, (2) the user interaction context, and (3) the translated expression. Figure 2.4 shows the MER contents in generic format. Hence, an entry in MER prevents the unnecessary repetition of calculations and analysis that were done when the condition was first encountered. The MER is scalable. Its contents are time-bounded (i.e. similar to electronic mail); its information is periodically updated, and very old records are deleted.

No.	MathML Expression	User Interaction Context	Translated Expression
1	MathML <Exp.1>	<user context a>, <environment context a>, <system context a>	Braille <Exp.1>
2	MathML <Exp.2>	<user context b>, <environment context b>, <system context b>	Braille <Exp.2>
3	MathML <Exp.3>	<user context c>, <environment context c>, <system context c>	Braille <Exp.3>
::	::	::	::
n-1	MathML <Exp. n-1>	<user context a>, <environment context b>, <system context j>	DotsPlus <Exp. n-1>
n	MathML <Exp. n>	<user context n>, <environment context n>, <system context n>	EasyMath <Exp. n>

Figure 2.4 The mathematical expression repository, in generic format.

2.4.5 The Data Analysis Layer

Here, we present the analysis and learning methods that are invoked in determining the optimal presentation of a mathematical expression based on a given interaction context.

The Visually-Impaired User's View of a Mathematical Expression

To a visually-impaired user, a *simple* mathematical expression (e.g. quadratic equation) becomes complex due to the presence of elements such as the *subscript*, *exponent*, *mathematical symbols* (e.g. Σ , Π , etc.), and the expression's *dimension* (i.e. complex numerator, denominator). In informatics, a mathematical expression is generally written using the syntax of *MathML* or *LaTeX* which are inappropriate format for visually impaired users. For example, a simple fraction in Figure 2.5 is shown with its equivalence in MathML, LaTeX, Braille and its linear representations.

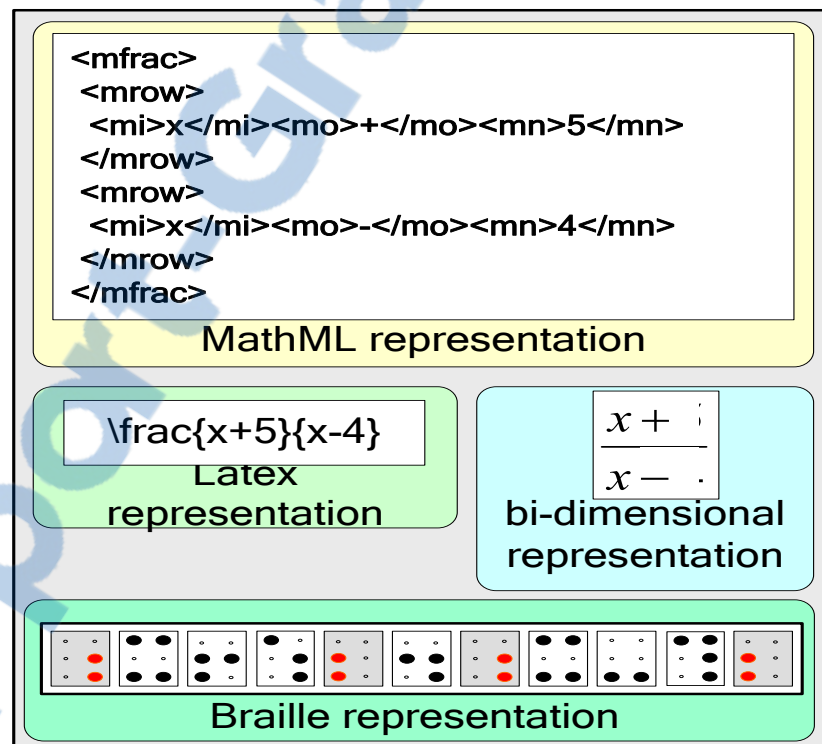


Figure 2.5 A fraction in bi-dimensional form and its corresponding equivalent in LaTeX, MathML and Braille.

Representation of Mathematical Operation in Different Formats

Presentation formats use different methods to represent a mathematical operation. Using *Braille*, there is a unique symbol for every operation. Using *speech*, an operation is uttered using a specific word (e.g. “+” is “add”, “-” is “minus”, etc). Using *DotsPlus*, an operation is represented by a unique symbol in a tactile form. Using *EasyMath*, every basic operation (e.g. +, -, ×, ÷, etc.) is represented by a unique symbol similar to its Braille representation. For special operation (e.g. \rightarrow , Π , \log , \int , \iint , \iiint , etc), however, the representation is in tactile form. Note that the representation of special operations in DotsPlus and EasyMath are not the same. For example, the + operation symbol is represented by a Braille symbol whereas in DotsPlus it is represented as “+” in embossed tactile form.

The Expression Complexity

The complexity of the expression affects the choice of the format of presentation. In case of simple expressions (see Figure 2.6 , expression (b)), the user will choose simple presentation format such as Braille or audio. Note that when the expression is complex, user has to choose more complex presentation format such as DotsPlus’s presentation (e.g. expression (a) in Figure 2.6). Hence, the complexity of the expression is important for determining the suitable presentation format. In (Awde, Bellik et Tadj, 2008), authors proposed a method to determine the complexity of the expression based on: the depth of syntax tree, number of operands and operators.

The Analysis and Conversion Agent

Figure 2.7 shows the functionalities of Analysis and Conversion Agent (ACA). It receives a mathematical expression (in MathML format) from SMA. Using *grammar rules and dictionary*, the expression is analyzed lexically. The result yields a list of lexemes. A lexeme is a parameter within an expression which may be an operand or an operator. Given the lexemes, the *parser* analyzes the expression parameters (i.e. operands, operators and syntax

tree) then sends parameters to the *Expression Evaluator* to determine its complexity. The *parser* sends then the expression to the *Expression Encoder* to be translated into its encoded format.

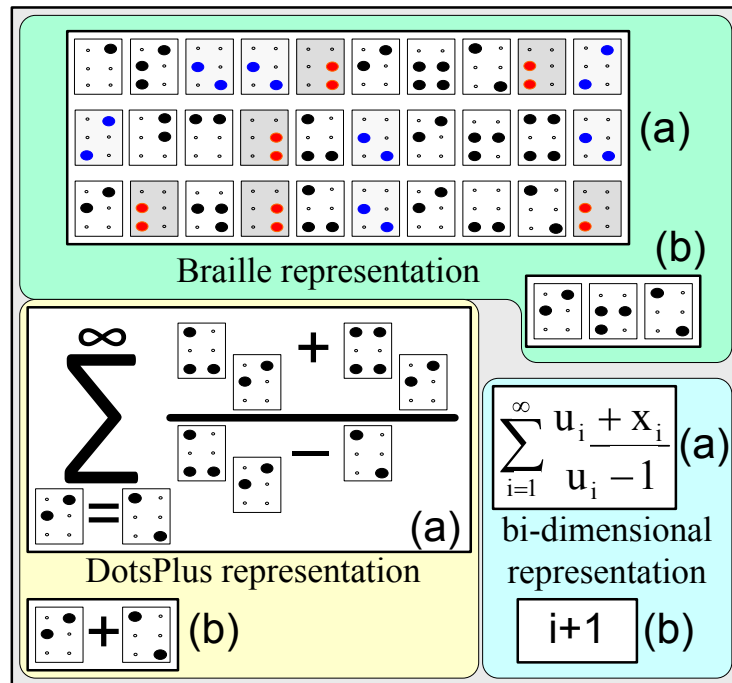


Figure 2.6 Two sample expressions in bi-dimensional form and its corresponding equivalent in Braille and DotsPlus format.

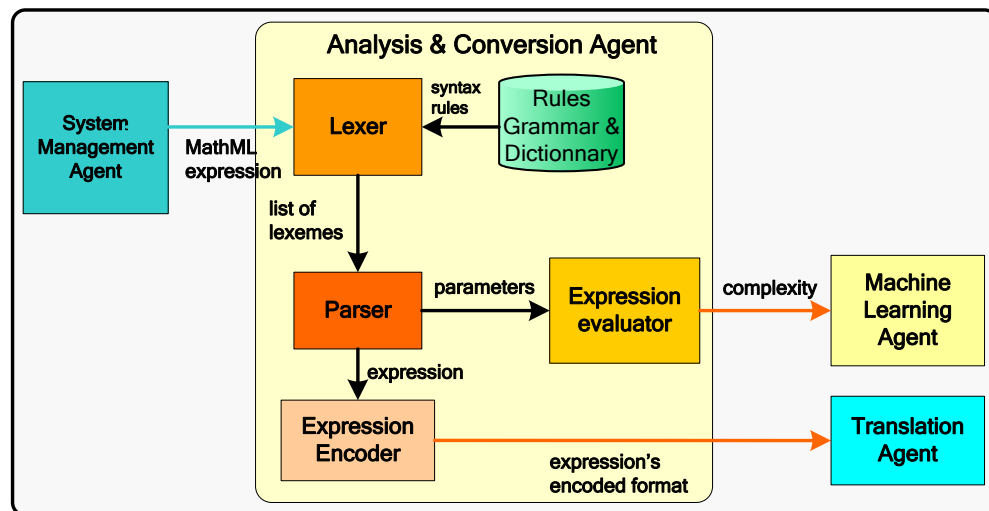


Figure 2.7 The Analysis and Conversion Agent.

As an example, Figure 2.8 shows the fraction defined in Figure 2.5, as a specimen expression. In (step 1), the expression is sent to the Lexer. In (step 2), using the XML grammar, the expression is decomposed into a list of lexemes; the list is then sent to the parser. In (step 3), the operations and operands in the expression are sent to expression evaluator and encoder. Together, in (step 4), the evaluator deduces the complexity of the expression (e.g. simple) while the encoder produces the encoded expression. Finally, in (step 5), the MLA is informed about the complexity of the expression, while the encoded expression is forwarded to the Translation Agent (TA).

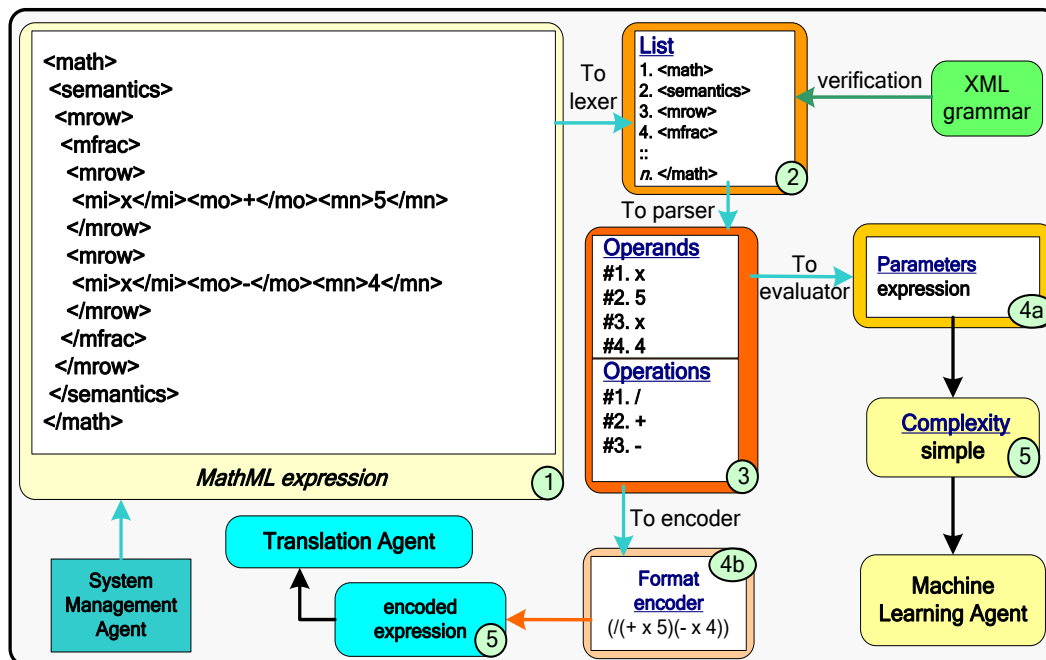


Figure 2.8 A sample analysis of a specimen fraction.

Determining a Mathematical Expression's Presentation Format

The MLA selects the presentation format based on interaction context and the complexity of the expression. The selection of the appropriate presentation format and the learning process are detailed in (Awde et al., 2009). Here, we present briefly the functionalities of MLA that are depicted in Figure 2.9.

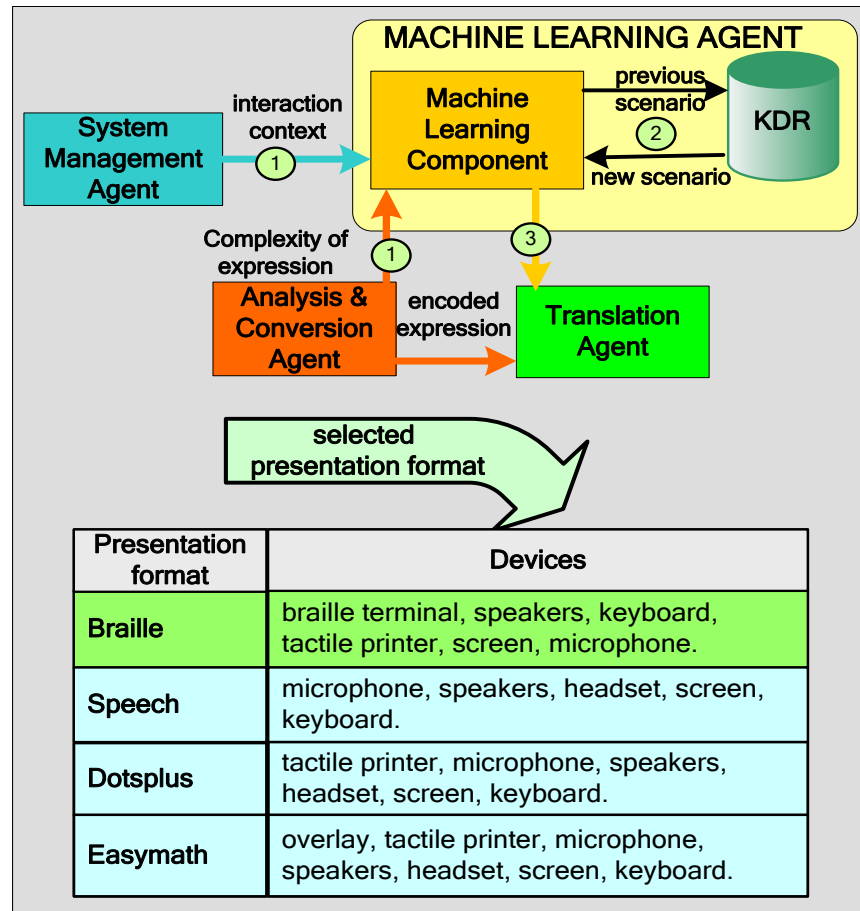


Figure 2.9 The Machine Learning Agent and its interaction with other system components.

In (step 1), MLA receives interaction context from SMA. The ACA informs the MLA of the complexity of the expression as discussed in previous section. The interaction context and the complexity of the expression input to the *machine learning component* (MLC) forms the pre-condition scenario. In (step 2), information about the corresponding post-condition is searched. If it is empty (i.e. it is a new scenario) then the MLA determines the modality that is appropriate for the interaction context. Next, it determines which available media devices support the chosen modality. Using the complexity of the expression, the MLA finally decides the optimal presentation format. The chosen presentation format becomes the post-condition scenario and is recorded in the *scenario repository* (SR). Otherwise, the user condition is not a new scenario; hence the MLA simply retrieves the entry in the post-

condition scenario. In (step 3), the chosen presentation format (e.g. Braille) is forwarded to the TA. The suitable media devices also are activated as shown in the bottom of Figure 2.9.

2.4.6 The Presentation Layer

Here, we present a mathematical expression presentation in a chosen format.

The Translation Agent

Using the chosen presentation format sent by MLA, the *Translation Agent* (TA) converts the encoded expression into its final presentation. As shown in Figure 2.10, the TA controller forwards the encoded expression to format translator(s) (i.e. *Braille*, *EasyMath*, *DotsPlus*, and *Speech*). See Figure 2.11, for the translation of specimen fraction into each of the 4 formats. The translated expression is then forwarded to the selected media devices and to SMA. The SMA then updates its mathematical expression repository by storing the new information (i.e. MathML expression, user interaction context, and the translated expression), implying that a new scenario (i.e. the current one) is encountered.

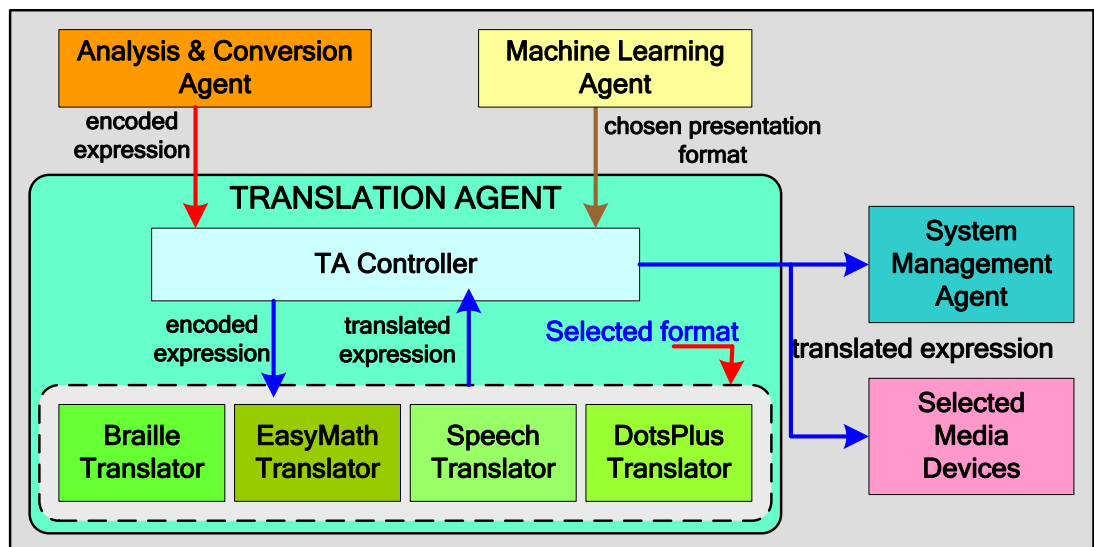


Figure 2.10 The Translator Agent and its components.

Encoded Expression	$((+ x 5)(- x 4))$
Presentation format	Translated expression
Braille	
EasyMath	
Speech	Open parenthesis x plus five close parenthesis divided by Open parenthesis x minus four close parenthesis
DotsPlus	

Figure 2.11 The translation of a specimen fraction into 4 formats.

Parameters Setting of Presentation Formats

TA must know the parameters of every presentation format if it is to produce a correct translation of a mathematical expression. In general, values of these parameters are set by the user himself. For example, for speech presentation, parameters such as language, gender, and speed of audio message are presented as per user specification. In Figure 2.12 (Right), the parameters of 4 presentation formats (and their sample values) are shown. The presentation format’s parameters settings are part of user preferences as contained in the user profile.

Media Device	Parameter Setting	Presentation Format	Format parameters
<device 1>	<parameter I1> = <value I1> ... <parameter It> = <value It>	Speech	language = French, age = young, gender = male, speed = 20 words/s
::	::		
<device n>	<parameter n1> = <value n1> ... <parameter nx> = <value nx>	Braille	notation = French braille, No. of character per line = 40
Keyboard	language = French Canadian	EasyMath	braille notation = French braille, size of tactile line = medium
Speaker	volume = medium bass = off	DotsPlus	braille notation = French braille, size of tactile line = medium

Figure 2.12 Parameters settings for presentation formats and some media devices.

Parameters Setting of Selected Media Devices

The selected media devices can also be configured so that their settings suit the user's needs. Figure 2.12 (Left) shows the generic format of media devices. As an example, the parameters of keyboard and speaker are set, as shown. During system activation, the parameters of each media device are set according to its media setting record by the device context negotiator. These settings, a part of user preferences, are relayed to the *Control Panel* of the operating system to effect the changes.

2.4.7 The Data Access Layer

The Searching and Editing Agent

The *Searching and Editing Agent* (SEA) allows navigation in and manipulation of a mathematical expression, controlling how mathematical terms are read and edited. SEA is composed of the following components: (1) the *Searcher*, (2) the *Editor*, and (3) the *MathML Generator*. SEA allows either sequential or random term access. Vocal commands for accessing mathematical terms are adopted because of their proven efficiency (i.e. cases of MathTalk and Meditor (Bellik, 1995)). Via vocal commands, the user directly accesses an object, and perceives the modifications on the object via tactile and/or sound feedback. Figure 2.13 illustrates the functionalities of SEA. Using user interface, the user can issue a vocal command (e.g. *go to <term 5>*). In (step 1), SMA sends user command to the Searcher. In (step 2), the Searcher verifies the validity of the command using grammar rules. If valid, the command is executed, and the result is sent to selected device(s) for presentation (step 3). Otherwise, no command is executed.

When the user edits an expression, the command (e.g. *Delete <term x>*) is sent by SMA to the Editor (step a). Upon command validation (step b), expression editing is executed and result is presented through selected device(s) (step c). This modification produces a new

expression (step d). The MathML Generator produces the new MathML code and sent it to SMA for presentation in the user interface (step e). In Figure 2.13 (Right), sample SEA commands are shown in tabular form. Note that when the presentation format is DotsPlus, search and modification of mathematical terms are not possible because the data are all embossed on paper (i.e. static media).

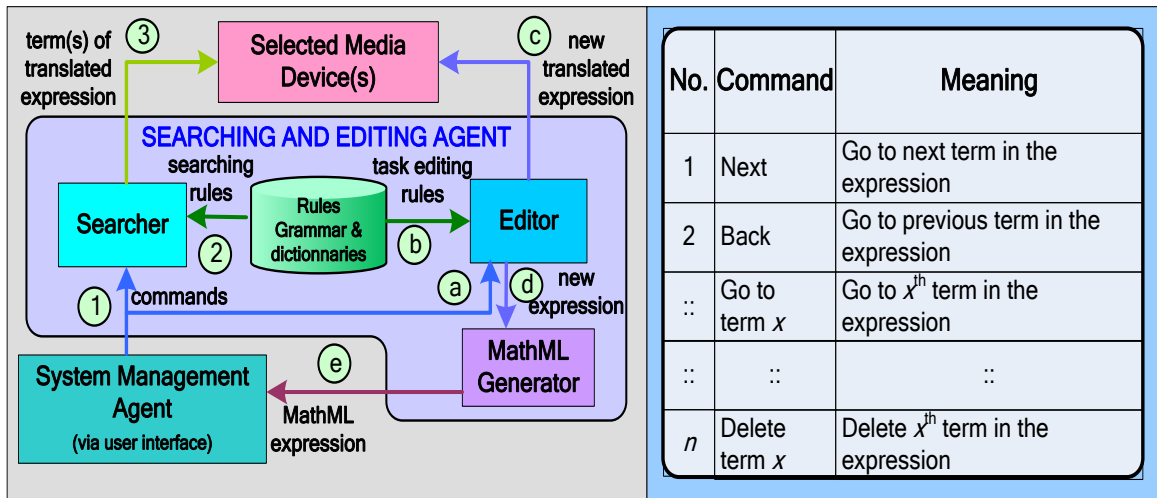


Figure 2.13 Searching agent and editor agent cooperates with Device manager agent.

2.5 Fault Tolerant System

2.5.1 General Principle of a Fault-tolerant System

In general, a fault-tolerant system is about the ability of a computing system to continue operating properly in the event of the failure of some of its components. It is designed to be able to handle several possible failures that may occur both in software-related faults such as communication between components or hardware-related faults such as input or output device failures.

In a fault-tolerant system, we have to proceed in three steps: 1) determine the set of faults to be tolerated; 2) choice of techniques which provide solution to the identified faults; and 3) test or experiment the efficiency of adopted techniques.

2.5.2 Our Multi-agent Fault-Tolerant System

The architecture of our system is designed to resist failure. When one or more faults (an agent or device is missing or defective) occur, the system would resist failure by self-reconfiguring. SMA reacts immediately to replace the failed component based on learned knowledge and users preferences (media devices priorities). Here, we can support 2 possible sources of failure: agent failure and defective or missing device.

A Failed Agent

In case of agent, SMA can perform various actions to prevent system crash. These actions are inspired from CONIC (Kramer et Magee, 1998) and AAA (Kumar et Cohen, 2000). The state of the agent determines action nature to be executed by the system. An agent can be in 1 of 4 states (see Figure 2.14): 1) *Idle* – the agent is ready to execute actions. 2) *Running* – it is active and ready to accept and react with others queries. 3) *Disconnected* – when the agent is still alive but it does not reply correctly. 4) *Stopped* – the agent is missed and cannot be repaired.

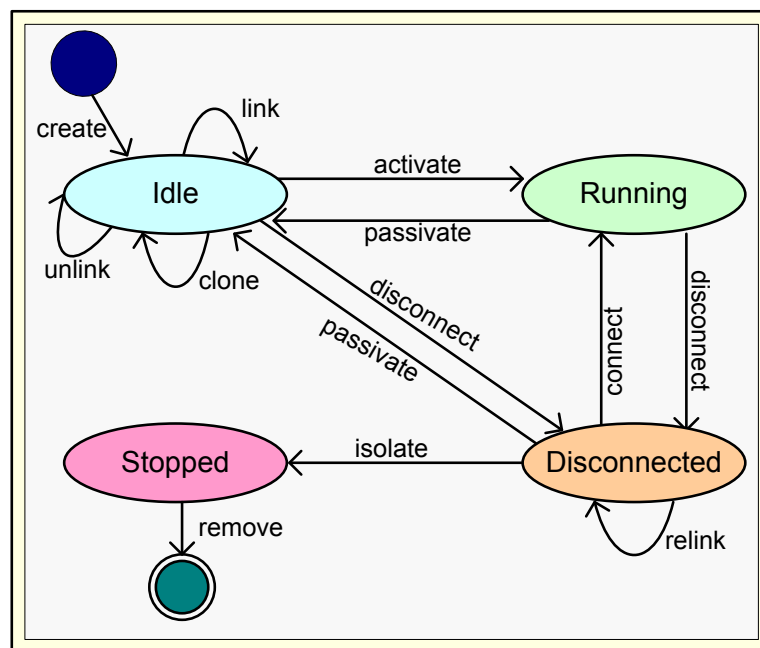


Figure 2.14 All possible states of agent and primitives in our system.

In our system, the configuration is realised by using some primitives as shown on Figure 2.13

- 1) *create* – it is the primitive that allows adding an agent to the configuration,
- 2) *clone* – it is used to clone an agent in the configuration,
- 3) *link* – it creates a connection between 2 agents,
- 4) *unlink* – it removes the connection between 2 agents,
- 5) *re-link* – it reconstitutes the connection between 2 agents,
- 6) *isolate* – when an agent is not repairable, it is isolated to be deleted,
- and 7) *remove* – it destroys the stopped agent.

For example, if the agent A is failed, SMA tries to repair it based on its state. If there is a problem of communication between agents (i.e. agent is disconnected), primitives such as link, unlink and re-link are useful. If problem persists SMA assigns the works to the duplicate of the agent A that is already added to configuration at the beginning (i.e. using clone primitive). So the failed agent is isolated and then it is removed from the configuration.

A Missing or Defective Device

Usually, when a media device is malfunctioning or absent (i.e. *failed*), the system searches the device (that is classified in the same group of modality) which is next in priority. If it is found, the replacement device is activated and the search is over. Otherwise, the system keeps searching for a replacement through *priority ranking order*.

Let there be a *media devices priority table* (MDPT) (see Tableau 2.1) containing media devices grouped according to the modality they support and arranged by priority ranking. When our system implements a modality, it selects the media device(s) that is/are ranked top in priority. It is also through the MDPT that the system searches for a replacement to a failed media device.

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 \mathbf{r} by the user. What happen

to the rankings of other devices \mathbf{d}_i , ($1 \leq i \leq n$, 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 $\mathbf{Priority}(\mathbf{d}_{\text{new}})$ set to \mathbf{r} then the priority of the other device i , ($1 \leq i \leq n$) denoted $\mathbf{Priority}(\mathbf{d}_i)$, remains the same. The second possibility is the priority rankings of all media devices ranked \mathbf{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: $\forall i, \exists r: \mathbf{r}; \forall \mathbf{d}_i, \exists \mathbf{d}_{\text{new}}: \mathbf{Devices} \mid (\mathbf{Priority}(\mathbf{d}_{\text{new}}) = \mathbf{r} \wedge \mathbf{Priority}(\mathbf{d}_i) \geq \mathbf{r}) \Rightarrow \mathbf{Priority}(\mathbf{d}_i)' = \mathbf{Priority}(\mathbf{d}_i) + 1$.

Tableau 2.1 A sample Media Devices Priority Table (MDPT).

Modality	Media Devices by Priority			
	1	2	3	n
Vocal input (V_{in})	microphone, speech recognition			
Vocal output (V_{out})	speaker, speech synthesis	headset		
Tactile input (T_{in})	keyboard	Braille terminal	overlay	
Tactile output (T_{out})	Braille terminal	tactile printer		

When a media device fault is detected and replaced by another one, the selection of the appropriate presentation format may be affected. Then ML must search the optimal presentation format based on the new situation. The process of selection of presentation format is discussed in (Awde et al., 2009).

2.6 Example of Simulation with Jade

JADE (Bellifemine, Caire et Greenwood, 2007) (Java Agent Development framework) is a software framework and middle-ware aimed at developing multi-agent applications conforming to FIPA standards. JADE is an Open Source project and has been coded in Java.

Using this framework, a programmer should code his agents in Java. JADE provides an implementation for the following components:

- Agent Management System (AMS). This agent is responsible for controlling access to the platform, authentication and registration of participating agents.
- Directory Facilitator (DF). This agent provides a yellow page service to the agents in the platform.
- Agent Communication Channel (ACC). This agent provides a white page service. It also supports inter-agent communication and inter-operability within and across different platforms.

When a JADE platform is launched, AMS and DF are immediately created and ACC module is set to permit communication between agents by set of messages. The agent platform can be distributed on several hosts. AMS and DF live in the main-container that is an agent and it contains the RMI registry used internally by JADE. The other agents created should be connected to the main- container.

The components of a multi-agent system implemented with JADE communicate with each other using flexible and efficient messaging services. According to the FIPA specification, agents communicate via asynchronous message (i.e. Agent Communication Language ACL messages) and the communication between agents involves an exchange of ACL messages. ACL is conceived in a formal language that avoids any ambiguity.

Figure 2.15 shows a Graphic User Interface (GUI) generated by the general management console for a JADE that is called RMA (Remote Management Agent). RMA provides control of all registered agent within platform, acquires information about the platform and executes GUI commands as create new agent, kill agent, etc. Through the RMA, a sniffer agent which is an important tool of JADE for monitoring and checking ACL messages exchanged among agents. When we sniff one agent or more, every message incoming/outgoing to/from agent is tracked and displayed in the Sniffer Agent's GUI in a diagram similar to UML sequence

diagrams. There are other useful tools in JADE (Dummy Agent, Introspector) for monitoring and debugging the multi-agent systems. In this example, we present a simple simulation of 2 agents of our system: SMA and DMA.

In our simulation, we demonstrate the communication and the coordination between SMA and DMA. Usually, DMA communicates with all negotiators that are responsible to determine the status of each device and fills the MDPT. User must only specify its preferences. However, we use a user interface for DMA in order to fill the MDPT as shown in Figure 2.15.

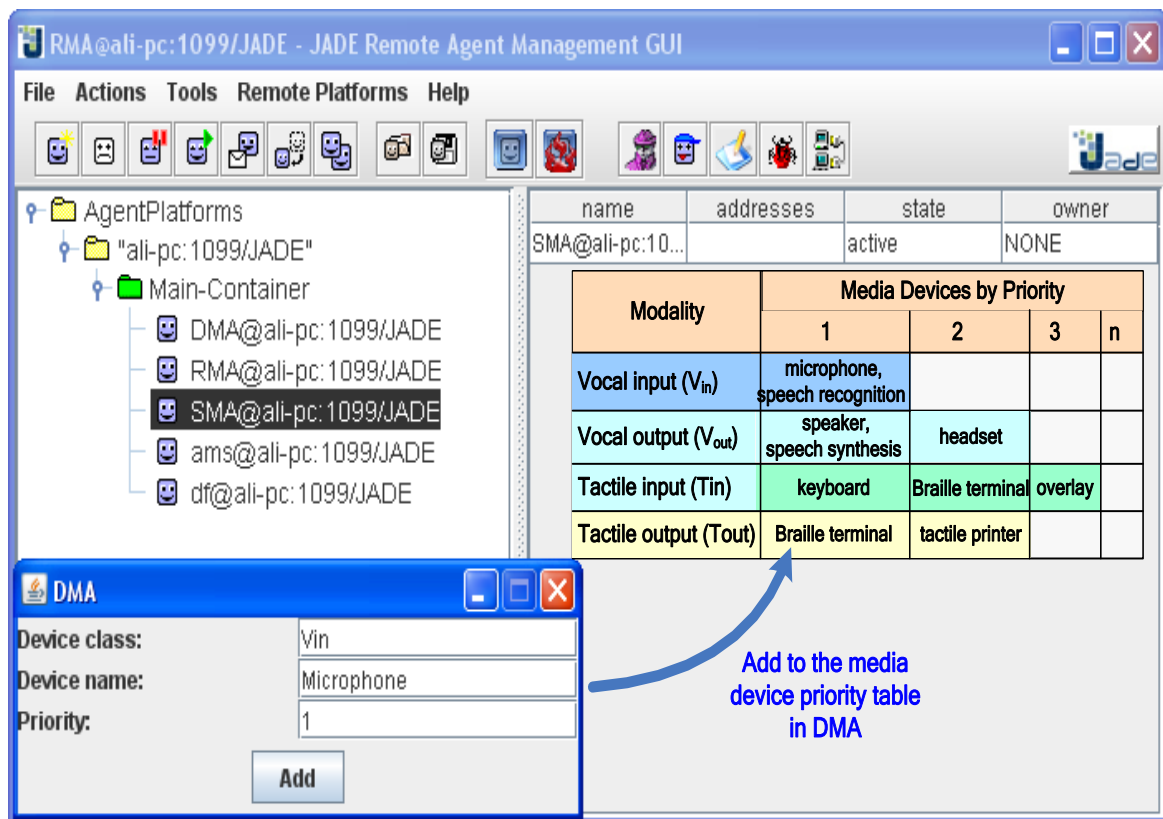


Figure 2.15 A simple Jade simulation of SMA and DMA, also a sample MDPT.

At the beginning, SMA has a modality and searches to select the media device that is top-ranked in priority. To do that, SMA communicates with DMA as shown on Figure 2.16. Messages (a and b) represent the case when there are no device of modality T_{out} available in

the MDPT. Messages (1,2,3 and 4) present a typical scenario when DMA has at least one device of the modality searched by SMA (i.e. here it is T_{out}). First, SMA looks to select the best available media device supporting T_{out} (a or 1) to DMA. If it is found, as in our example, it replies with a proposal message (2) that contains name (Braille Terminal) and rank (10) of the found media. Otherwise, it replies with a refuse message (b) to inform SMA that no device is available supporting T_{out} . In case of proposal message, SMA replies with an acceptance (3) and DMA informs (4) SMA that the media device (i.e. Braille Terminal) is ready to be used. Also, these steps are executed in finding replacement to a failed device.

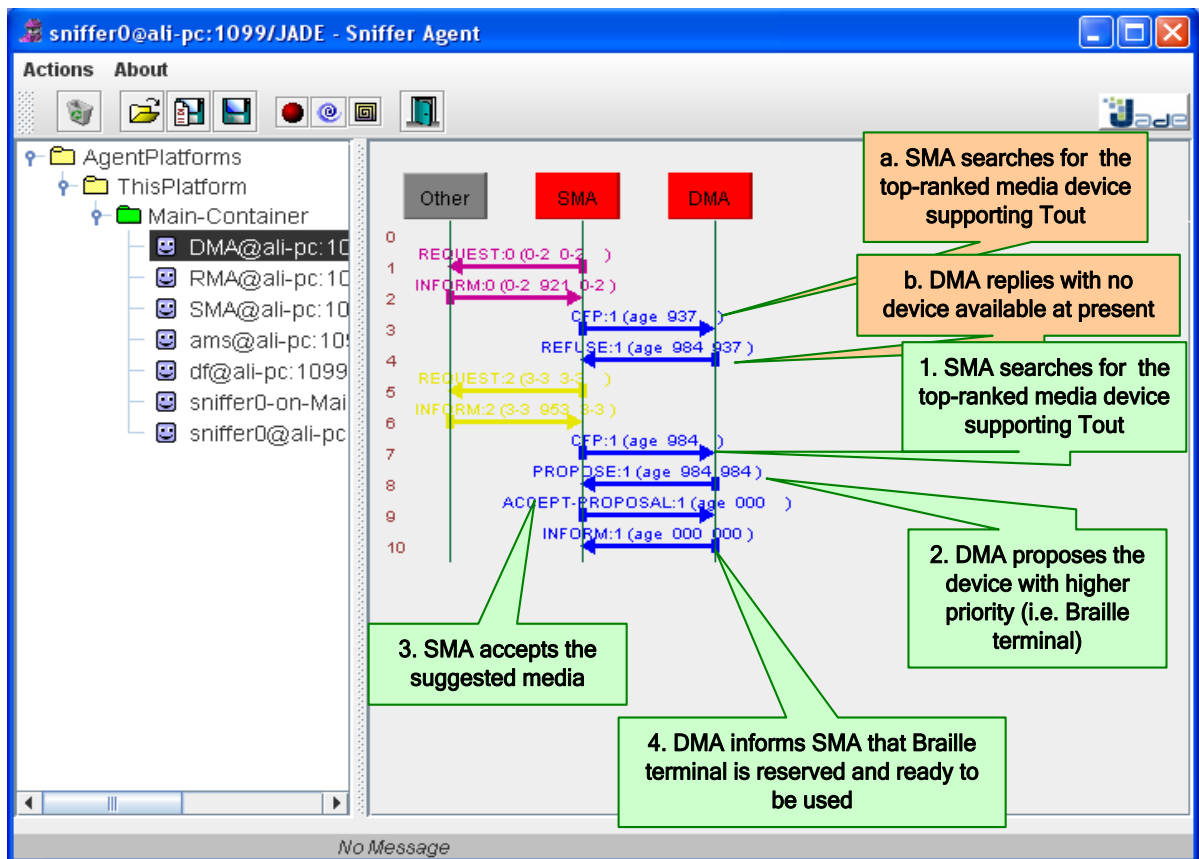


Figure 2.16 Sniffer agent monitors and checks ACL messages exchanged among agents SMA and DMA.

2.7 Conclusion

Our ongoing research is focused on providing computing infrastructure to visually-impaired user through multimodality. One area of such domain is the infrastructure for mathematical presentation to blind users which this paper addresses. In this work, we presented an infrastructure supporting the presentation of mathematical expressions. Our multi-agent system considers the interaction context (i.e. combined user's, environment's and system's contexts) as well as the nature of the mathematical expression itself and of the user's preferences.

In this paper, we have presented the architecture of our adaptive multi-agent system. Also, we have presented the agents' functionalities in the system's layers. For each layer, we have shown the agents and their behaviour.

The architecture of our infrastructure is layered, thus encapsulating the components of the various layers. It is adaptive that it is capable of determining the best configuration (modality, media, and presentation) for the user. In case of failure of media device, our system is capable of shutting down the faulty component and replacing it with a new one (if a replacement is available). If replacement is not possible, the system re-determines the new modality and presentation format apt for the new configuration. The human intervention is greatly reduced in our system as it is capable of self-configuration, and learning. This system feature promotes autonomy to visually-impaired users, thus enhancing their information processing productivity.

In order to demonstrate the behaviour of the agents, a simulation has been carried out on the Java Agent Development Framework (JADE) platform. This simulation has confirmed the efficiency of our system design.

This work is our continuing contribution to advance research on making informatics more accessible to handicapped users. Our future works involve the prototyping of this

infrastructure and simulating its performance using several computing platforms. Such prototype will also be tested on visually-impaired users with other varying interaction contexts.

2.8 References

- Awde, A., Y. Bellik et C. Tadj. 2008. « Complexity of Mathematical Expressions in Adaptive Multimodal Multimedia System Ensuring Access to Mathematics for Visually Impaired Users ». *International Journal of Computer and Information Science and Engineering*, vol. 2, n° 2, p. 103-115.
- Awde, A., M. D. Hina, C. Tadj, Y. Bellik et A. Ramdane-Cherif. 2009. « An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually- Impaired Users ». *Journal of Multimedia (JMM)*. Vol.4, No. 4, 2009.
- Awde, A., C. Tadj et Y. Bellik. 2008. « Un système multi-agent pour la présentation d'expressions mathématiques à des utilisateurs non-voyants ». In *21ième Conférence Canadienne de génie électrique et génie informatique*. Niagara Falls, Ontario, Canada: IEEE Canada.
- Bellifemine, F. L., G. Caire et D. Greenwood. 2007. *Developing Multi-Agent Systems with JADE*. Wiley, 300 p.
- Bellik, Y. 1995. « Interfaces multimodales : concepts, modèles et architectures. ». Thèse de doctorat en informatique, Paris, Université de Paris-Sud XI Orsay.
- Bourbakis, N. G. and D. Kavraki. *An Intelligent Assistant for Navigation of Visually Impaired People*. in the *2nd IEEE International Symposium on Bioinformatics and Bioengineering Conference*. 2001.
- Cahill, H., C. Lineham, J. McCarthy, G. Bormans et J. Engelen. 1995. « Ensuring Usability in MATHS ». In *The European Context for Assistive Technology, Assistive Technology Research Series*. p. 66-69. Amsterdam: IOS Press.
- Edwards, A. D. N., et R. D. Stevens. 1994. « A Multimodal Interface for Blind Mathematics Students ». In *INSERM'94*. p. 97 - 104. Paris, France.
- Ferber, J. 1995. *Les systemes multi-agents*. Paris: InterEditions.
- Ferreira, H., et D. Freitas. 2004. « Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project ». In *9th International Conference on Computer*

- Helping People with Special Needs (ICCHP)*. p. 678-685. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Ferreira, H., et D. Freitas. 2005. « AudioMath: Towards Automatic Readings of Mathematical Expressions ». In *Human-Computer Interaction International (HCII)*. Las Vegas, Nevada, USA.
- Garlini, P., et F. Fogarolo. 2003. *LAMBDA: Linear Access to Mathematics for Braille Device and Audio Synthesis - Analysis of the User's Needs*. Padova, Italy: University of Padova.
- Jennings, N. R., et M. J. Wooldridge. 1998. « Applications of Intelligent Agents ». In *Agent Technology: Foundations, Applications, and Markets*, sous la dir. de Jennings, Nicholas R., et Michael J. Wooldridge. p. 3-28. Springer-Verlag: Heidelberg, Germany.
- Kramer, J., et J. Magee. 1998. « Analysing Dynamic Change in Software Architectures: A Case Study ». In *Proceedings of the International Conference on Configurable Distributed Systems* Washington, DC, USA: IEEE Computer Society.
- Kumar, S. and P. R. Cohen. *Towards a Fault-Tolerant Multi-Agent System Architecture*. in *The fourth international conference on Autonomous agents 2000*: ACM Press.
- Lécuyer, A., et al. *HOMERE: a Multimodal System for Visually Impaired People to Explore Virtual Environments*. in *Proceedings of the IEEE Virtual Reality 2003*. Washington, USA: IEEE Computer Society.
- Lightfoot, D. 2001. *Formal Specification Using Z*, 2nd. McMillan Press.
- Moço, V., et D. Archambault. 2003. « VICKIE: A Transcription Tool for Mathematical Braille ». In *7th European Conference for the Advancement of Assistive Technology in Europe (AAATE)* (September 2003). Dublin, Ireland: IOS press.
- Podevin, A. 2002. « Accès aux formules mathématiques par des personnes non-voyantes : étude et définition d'une méthode adaptée ». DEA, Université de CAEN.
- Preddy, M., J. Gardner, Steve Sahyun, Oregon State University et Dave Skrivanek. 1997. « Dotsplus: How-to Make Tactile Figures and Tactile Formatted Math ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los-Angeles, CA, USA
- Raman, T. V. 1998. *Audio System for Technical Readings*, 1410. Berlin, Germany: Springer-Verlag.

- Schwebel, F. and R. Goiffon. *BraMaNet: Quelques règles simples à connaître pour qu'un aveugle puisse lire vos documents mathématiques et vos pages web.* in *Journées nationales Caen*. 2005. Caen, France.
- Stöger, B., K. Miesenberger et M. Batusic. 2004. « Mathematical Working Environment for the Blind Motivation and Basic Ideas ». In *9th International Conference on Computers Helping People with Special Needs (ICCHP)*. p. 656-663. Springer Verlag, Lecture Notes in Computer Science (LNCS).
- Weiss, G. 1999. « Multiagent systems ». MIT-Press.
- Wooldridge, M. 2002. *An Introduction to Multiagent Systems*. Chichester, England: Wiley.

CHAPITRE 3

AN ADAPTIVE MULTIMODAL MULTIMEDIA COMPUTING SYSTEM FOR PRESENTATION OF MATHEMATICAL EXPRESSIONS TO VISUALLY- IMPAIRED USERS

Ali Awde¹, Manolo Dulva Hina^{1,2}, Chakib Tadj¹

¹Département de Génie Électrique, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

²PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines
45, avenue des États-Unis, 78035 Versailles Cedex, France

Yacine Bellik³

³Laboratoire LIMSI-CNRS, Université de Paris-Sud
B.P. 133, 91043 Orsay, France

Amar Ramdane-Cherif^{2,4}

⁴LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines,
Bâtiment Descartes, 45 avenue des États-Unis - 78000 Versailles, France

Ce chapitre est accepté pour publication dans la revue Journal of Multimedia (JMM). Vol. 4,
No. 4, 2009.

Résumé

Présenter les expressions mathématiques aux utilisateurs non-voyants est une tâche difficile. Les solutions proposées dans la littérature présentent les expressions mathématiques aux déficients visuels sous une seule forme de présentation et ne prennent pas en considération le contexte de l'interaction ni la nature de l'expression à présenter.

Dans cet article, nous présentons les expressions mathématiques en utilisant plusieurs formes de présentation et en tenant compte de l'utilisateur, de son environnement et de son système ainsi que de la nature de l'expression elle-même.

La solution présentée dans ce travail est plus efficace lorsque les utilisateurs sont familiers avec plusieurs formes de présentation. Contrairement aux solutions précédentes, la nôtre prend en considération la situation de l'utilisateur et propose une présentation selon sa

convenance et ses capacités. Cette contribution vise à améliorer l'accessibilité de l'information pour les non-voyants.

Mots clés: Système adaptatif, mathématiques pour les non-voyants, système multimodal, interaction homme-machine.

Abstract

Presenting mathematical expressions to visually-impaired users is a challenging task because, unlike text, a mathematical expression is bi-dimensional and has some distinguishing characteristics. At present, the state-of-the art solutions present mathematical expressions to blind users in only one presentation format and with no consideration of user's context. In this work, we present mathematical expressions in more than one presentation format and consider the context of the user, his environment and this computing system as well as the nature of the expression itself and of the user's preferences. The solution presented in this work is most efficient when users are familiar with many presentation formats. Unlike the current state-of-the art solutions, ours takes into account the user's situation and present a solution that is suitable to his context and capacity. This work is our contribution to an ongoing research to make informatics more accessible to handicapped users.

Index Terms: Visually-impaired users, mathematical presentation, multimodal multimedia computing, machine learning, human-machine interaction.

3.1 Introduction

Recently, various systems have been specifically developed for visually-impaired people so they can use computing systems. A number of such systems have been developed in order to allow users to access mathematical expressions. The presentation of mathematical expressions to blind users is a challenge due the following reasons: First, the visual mathematical representation, unlike that of text, is bi-dimensional and the interpretation of a mathematical expression is related to one's knowledge of the expression's spatial

components. Second, the conversion of a multi-dimensional structure to a non-visual representation is not straightforward. For example, the representation of an expression, say in Braille, requires additional information to explain a specific term (for example, an exponent). Also, the conversion of a mathematical expression into a certain format, say audio, is often ambiguous. Third, the vocabulary terms used by sighted people are quite large compared to the amount of data that can be made accessible to a visually-impaired user. For example, a standard 6-dot Braille can encode only 64 characters. This number of characters is not enough to represent all frequently-used mathematical symbols. Hence, a mathematical symbol itself is made represented by a combination of these characters. However, large quantity of symbols is a challenge to blind users. Finally, the manipulation of data for visually-impaired user is not as convenient as it is for sighted people. For example, Braille characters are often embossed into a paper. Once an expression is printed in a static media, the user cannot easily change the data.

Indeed, to visually-impaired users, a mere understanding of a mathematical expression requires repeated passage on the expression in which the user sometimes skips some secondary information, only to revert back to them again and again until he fully grasps the meaning of the expression. A complicated task like this is detailed in (Stöger, Miesenberger et Batusic, 2004). Hence, some tools were developed to lessen the complexity of performing a similar task. Among these tools are MathTalk (Edwards et Stevens, 1994), Maths (Cahill et al., 1995), DotsPlus (Preddy et al., 1997), EasyMath (Podevin, 2002), and AudioMath (Ferreira et Freitas, 2004; 2005). MathTalk and Maths convert a standard algebraic expression into its audio equivalent. In using Maths, the user can read, write and manipulate mathematics by using a multimedia interface that contains speech, Braille and audio. Aster (Audio System for Technical Readings) (Raman, 1998), on its part, takes in a Latex document and reads it using various tones to denote different terms of the expression. VICKIE (Visually Impaired Children Kit for Inclusive Education) (Moço et Archambault, 2003a) and BraMaNet (Braille Mathématique sur le Net) (Schwebel et Goiffon, 2005) are transcription tools that convert a mathematical document (written in Latex, MathML, etc.) to its Braille representation. DotsPlus is a tactile method of printing documents that

incorporates both Braille representation and graphic symbols (e.g. \prod , \sum , etc.). For EasyMath, its objective is to produce a bi-dimensional output of a mathematical expression, similar to the representation for sighted people, using Braille characters and an overlay keyboard.

None of the tools cited above, however, is complete. Studies were conducted evaluating these tools based on users' needs (Garlini et Fogarolo, 2003; Podevin, 2002). The results indicate that users are neither independent nor able to do their homework (i.e. case of students) without the help of sighted people. Indeed, each of these tools has its own set of limitations. For example, Aster uses only a LaTeX document while AudioMath allows only MathML document for conversion into an audio output.

To address the weaknesses of the various existing solutions cited above, we do get the strength of each tool and integrate each one in our work to build a system that (1) extends the limitations of currently existing systems, (2) provides users with opportunities to access as many types of document as possible, and (3) presents data output in as many suitable formats as possible after considering user situation and the special symbols within the expression. This work is essential as it offers to provide all types of data presentation formats while requiring minimum explicit intervention from the user.

In this paper, we present the infrastructure of a multimodal multimedia computing system that presents mathematical expressions to visually-impaired users after taking into account the user's interaction context and preferences and the nature of the expression itself. A part from this introductory section, the rest of this paper is structured as follows. Section 3.2 lists down the technical challenges in this work and essays our approaches to address each one of them. Section 3.3 presents the design and principle that address our system requirements. System design is provided in Section 3.4. In Section 3.5, we provide examples that apply the concepts discussed in the previous section. Also, we present some formal specifications on the functionalities of our system. Finally, we present our future works and conclusion in section 3.6.

3.2 Technical Challenges

Our principal goal is to build a model of an adaptive multimodal system that correctly presents mathematical expressions to visually-impaired users. In our proposed solution, we first consider the user's interaction context as a requisite in determining the most appropriate (a.k.a. optimal) *modality*. Here, we define modality as the mode of human-computer interaction. Next, knowing the optimal input/output modalities and using the media devices' priority ranking, we determine the media devices that would support the chosen modalities. Then, we determine the nature of the mathematical expression – whether or not it contains special mathematical operation symbol (more details provided in Sections 3.3.4 and 3.3.6). Finally, we determine the most suitable presentation format for the given mathematical expression based on the available media devices, the nature of mathematical expression and the user's preferences on presentation formats. The design of such a system needs to address the key requirements cited below:

Requirement 1: Provide a relationship between an interaction context (i.e. combined user, environment and system context) and the suitable modalities for visually-impaired users. What constitutes a user context given that a user is already visually-impaired? What parameters constitute the user, the environment and the system context? What are the modalities that are available to a blind user? On what basis a specific modality is selected?

Requirement 2: Provide a relationship between modality and media devices, and a system mechanism for a seamless adaptation when a media device fails (i.e. absent or defective) or is newly added. What relationship exists between media devices and modalities? When a media device is shut down, does it follow that the modality associated to the device must also be shut down? When there are many available media devices to support a modality, do they have to be activated all together? If not, then how are they prioritized?

Requirement 3: Provide a mechanism for optimal selection of a mathematical expression's presentation format based on available media devices and their priority rankings relative to

the modalities they support and of user's format preferences (as per presence/ absence of special operation symbol within the expression). What are the available presentation formats? How are they ranked? Is it possible to have two or more presentation formats activated at the same time?

Requirement 4: Provide a learning mechanism that is applicable to the selection of modality given an interaction context. Derive a relationship that relates media devices selection with respect to the given modality. Derive a function that selects a most suitable presentation format given the available media devices, the nature of mathematical expression and the user's preferences on presentation formats. What are the condition, the decision and the corresponding action that an intelligent multimodal system must learn? Is the learning continuous or will it stops after learning certain amount of knowledge?

We address the technical challenges by providing specific solutions to the system requirements cited above.

Proposed Solution to Requirement 1: The possible modalities left for visually-impaired users are tactile and vocal modalities (i.e. visual modality is already not suitable), both for data input and output. Throughout this paper, we will use the following modality designations: \mathbf{V}_{in} = *vocal input*, \mathbf{V}_{out} = *vocal output*, \mathbf{T}_{in} = *tactile input*, and \mathbf{T}_{out} = *tactile output*. In a user context, further handicap (e.g. manual handicap or lack of knowledge in using Braille) affects modality selection (e.g. tactile input and output). The noise level and environment restriction also affects the possibility of using vocal input and/or output modality. "Noise", in acoustics and in this paper, refers to meaningless or unwanted sound that is louder than a desired volume. We classify two levels of noise – *quiet* and *noisy*. "Environment restriction" is a term we used to refer to whether the working place imposes mandatory silence or not. More details of the effect of noise level and environment restriction on modality selection are provided in Section 3.4.2. In the system context, for example, the use of PDA and cellular phone as computing device also affects the possibility of using tactile modality. The ensemble of all these context parameters (i.e. all together forms the

interaction context) determines whether a modality is possible or not. Once possible, the interaction context also determines the most suitable modalities.

Proposed Solution to Requirement 2: We have a *media devices priority table* (MDPT) containing records of all media devices known to the system. Our system assigns *a priori* the type of modality that some familiar media devices support. The end users will do the same when new media devices are introduced into the system. There is a possibility, however, that two or more media devices support a common modality (e.g. both overlay keyboard and Braille terminal support tactile output mode). In such a case, the end user provides priority rankings on these media devices. The default devices (and software such as speech recognition software) are always ranked top in the priority ranking. Two or more devices, however, may share a common priority ranking, if end user so desires.

Proposed Solution to Requirement 3: The presentation format selection is resolved using a formula based on priority rankings of presentation formats and their supporting media devices. The format with the highest score is selected as the final presentation format. In case of similar scores for two (or more) presentation formats, both are selected since both are possible. The presence of special operation symbol in a mathematical expression yields a presentation format's priority ranking that is different from one that has no special symbol. Such priority ranking is obtained from user's profile.

Proposed Solution to Requirement 4: We provided a generic representation of interaction context. By using the proposed supervised learning method, our system learns what modalities are suitable to a given instance of interaction context. A relationship is established between modalities and media devices; media devices themselves are ranked by priority, the ranking being provided by the end user. This ranking is also important during the search for a replacement to a failed media device. Also, a function is derived that helps determine all possible presentation formats. By using the tables of user's preferences and the nature of the mathematical expression, we can determine the most suitable presentation format for the user. Our proposed method for knowledge acquisition is incremental, thus continuous. In this

manner, the system becomes adaptive as the system evolves. This adaptation includes schemes such as an introduction of newly added media devices, a modification of interaction context parameters, etc. In general, we attempt to make our system adaptive to most possible revisions with minimum cost of modification.

3.3 Detailed Design and Principles for the Resolution of System Requirements

The details of the proposed solutions based on the requirements cited in the previous section are discussed in this section.

3.3.1 Using Machine Learning to Find Optimal Suitable Modalities to an Interaction Context

Machine learning (ML) is concerned with the design and development of algorithms and techniques that optimizes an entity's performance using sample data or past experience (Alpaydin, 2004). In our work, ML is adopted in order that our system learns from its previous experiences.

A *scenario* is an event that needs system's response. The set of inputs that triggers a scenario is called the *pre-condition scenario* while the state of the system after reacting to a scenario is called *the post-condition scenario*. In this work, the system acquired knowledge about scenarios is stored in a *scenario repository* (SR).

Let interaction context, $\mathbf{IC} = \{\mathbf{IC}_1, \mathbf{IC}_2, \dots, \mathbf{IC}_{\max}\}$, be a set of all possible interaction context. At any given time, a user has a specific interaction context i denoted \mathbf{IC}_i , $1 \leq i \leq \max$. Formally, an interaction context is a tuple composed of a specific user context (\mathbf{UC}), environment context (\mathbf{EC}) and system context (\mathbf{SC}). An instance of \mathbf{IC} may be written as:

$$\mathbf{IC}_i = \mathbf{JC}_k \otimes \mathbf{C}_1 \otimes \mathbf{C}_m \quad (3.1)$$

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

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

$$UC_k = \bigotimes_{x=1}^{max_k} ICParm_{kx} \quad (3.2)$$

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

$$EC_l = \bigotimes_{y=1}^{max_l} ICParm_{ly} \quad (3.3)$$

$$SC_m = \bigotimes_{z=1}^{max_m} ICParm_{mz} \quad (3.4)$$

The first knowledge that our system's ML component must learn is to relate the interaction context to an appropriate modality. In general, a modality is possible if there exists at least one modality for data input and at least one modality for data output. Given a modality set **M** = {**V_{in}**, **T_{in}**, **V_{out}**, **T_{out}**} then modality is possible under the following condition:

$$\text{Modality Possible} = (V_{in} \vee T_{in}) \wedge (V_{out} \vee T_{out}) \quad (3.5)$$

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

$$\text{Modality Failure} = (\text{V}_{\text{in}} = \text{failed}) \wedge (\text{T}_{\text{in}} = \text{failed}) \vee ((\text{V}_{\text{out}} = \text{failed}) \wedge (\text{T}_{\text{out}} = \text{failed})) \quad (3.6)$$

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

Let \mathbf{M}_j = element of the power set of \mathbf{M} , that is, $\mathbf{M}_j \in \mathcal{P}(\mathbf{M})$ where $1 \leq j \leq \text{mod_max}$ (maximum modality). Also, let $\widehat{\mathbf{M}}$ = the most suitable \mathbf{M}_j for a given interaction context \mathbf{IC}_i . Let \mathbf{X} be the set of pre-condition scenarios. Hence, the relationship between \mathbf{X} and \mathbf{IC} can be written as $\mathbf{X}_i = \mathbf{IC}_i$. For the purpose of simplicity, we let the pre-condition set \mathbf{X} be represented by interaction context \mathbf{IC} . Each interaction context i , \mathbf{IC}_i , is composed of attributes of n components, that is, $\mathbf{IC}_i = (A_1, A_2, \dots, A_n)$ where attribute is a parameter that may belong to **UC** or **EC** or **SC**. We also let the set of post-condition \mathbf{Y} be represented by a set of modality \mathbf{M} . Let the function f map the set of \mathbf{IC} to the set of \mathbf{M} , in which h calculates the suitability score of such a mapping, and is given by:

$$h(\mathbf{IC}_i) \rightarrow \mathbf{M}_j = \langle \text{suitability_score} \rangle \quad (3.7)$$

In Mathematics, the probability function h can also be written as:

$$h = P(\mathbf{M}_j | \mathbf{IC}_i) \quad (3.8)$$

which should be read as the probability of the occurrence of \mathbf{M}_j given an \mathbf{IC}_i . To simplify calculation, *Bayes Theorem* (Kallenberg, 2002), given below, can be adopted:

$$P(\mathbf{M}_j | \mathbf{IC}_i) = \frac{P(\mathbf{IC}_i | \mathbf{M}_j) \times P(\mathbf{M}_j)}{P(\mathbf{IC}_i)} \quad (3.9)$$

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 \mathbf{IC}_i attributes

A_1, \dots, A_n are all conditionally independent of one another given a post condition M_j . The representation of $P(IC_i|M_j)$ becomes:

$$\begin{aligned} P(IC_i | M_j) &= P(A_1, \dots, A_n | M_j) \\ &= P(A_1 | M_j) \times \dots \times P(A_n | M_j) \\ &= \prod_{i=1}^n P(A_i | M_j) \end{aligned} \quad (3.10)$$

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 \mathbf{M} (i.e. M_1, M_2, \dots, M_m). Given that $IC_i = (A_1, A_2, \dots, A_n)$, then (9) becomes:

$$P(M_j | A_1 \dots A_n) = \frac{P(M_j)P(A_1 \dots A_n | M_j)}{\sum_{k=1}^m P(M_k)P(A_1 \dots A_n | M_k)} \quad (3.11)$$

Equation (11) can also be written as:

$$P(M_j | A_1 \dots A_n) = \frac{P(M_j) \prod_{i=1}^n P(A_i | M_j)}{\sum_{k=1}^m P(M_k) \prod_{i=1}^n P(A_i | M_k)} \quad (3.12)$$

which is the fundamental equation for the Naive Bayes classifier. Given a *new* instance of interaction context $IC_{new} = (A_1 \dots A_n)$, the equation shows how to calculate the probability that M_j will take given the observed attribute values of IC_{new} and given that the distributions $P(M_j)$ and $P(A_i|M_j)$ 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:

$$h_{best} = \hat{M} = \arg \max_j \left(\frac{P(M_j) \prod_{i=1}^n P(A_i | M_j)}{\sum_{k=1}^m P(M_k) \prod_{i=1}^n P(A_i | M_k)} \right) \quad (3.13)$$

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

$$h_{\text{best}} = \hat{M} = \arg \max_j \left(P(M_j) \prod_{i=1}^n P(A_i | M_j) \right) \quad (3.14)$$

where $P(M_j)$ = the frequency of M_j in SR \div cardinality of (SR).

3.3.2 Algorithms for Finding Optimal Modalities for Interaction Context

In Appendix A, Figure 3.12 shows the algorithm that calculates the suitability score of an element of a modality power set based on a given interaction context. Figure 3.13, on the other hand, finds the optimal modality for the given instance of interaction context.

Given that $M = \{V_{\text{in}}, T_{\text{in}}, V_{\text{out}}, T_{\text{out}}\}$, then the power set (i.e. the set of all subsets) of M is given by $\mathcal{P}(M) = \{\{V_{\text{in}}\}, \{T_{\text{in}}\}, \{V_{\text{out}}\}, \{T_{\text{out}}\}, \{V_{\text{in}}, T_{\text{in}}\}, \{V_{\text{in}}, V_{\text{out}}\}, \{V_{\text{in}}, T_{\text{out}}\}, \{V_{\text{in}}, T_{\text{in}}, V_{\text{out}}\}, \{V_{\text{in}}, T_{\text{in}}, T_{\text{out}}\}, \{V_{\text{in}}, V_{\text{out}}, T_{\text{out}}\}, \{V_{\text{in}}, T_{\text{in}}, V_{\text{out}}, T_{\text{out}}\}, \{T_{\text{in}}, V_{\text{out}}\}, \{T_{\text{in}}, T_{\text{out}}\}, \{T_{\text{in}}, V_{\text{out}}, T_{\text{out}}\}, \{V_{\text{out}}, T_{\text{out}}\}, \{\}\}$. M_j , therefore, evaluates the suitability score of each element of $\mathcal{P}(M)$, see Figure 3.12. The optimal modality, $\hat{M} = h_{\text{best}}$, is then chosen from one of these elements (see Figure 3.13). The selected element is one that satisfies (3.14) and one having the highest suitability score.

3.3.3 Relationship between Modality and Media Devices

Once the optimal (input/output) modalities (i.e. \hat{M}) are chosen for a given interaction context, the next step is to determine if media devices are available to support the chosen modalities. If so, then these devices are activated. Otherwise, the chosen optimal modalities are re-evaluated, that is, the modality that cannot be supported by media devices is taken out from the selection. Afterwards, the result is again evaluated to determine if (3.14) still holds. If the answer is affirmative, then modality is possible and the result is implemented. Otherwise,

there is a failure of modality. Figure 3.14 (see Appendix A) shows the algorithm that implements the update of optimal modalities based on availability of media devices.

There are usually more than one media devices that support a specific modality. For example, a regular keyboard, an overlay keyboard and a Braille terminal all support tactile input modality. Activating them all is just plain redundancy; hence the system must select one from the given available devices. If these devices have equal chances of being selected, then device selection becomes random, and the chance that device D_i gets chosen out of n available devices is given by:

$$P(D_i/M_j) = 1/n \quad (3.15)$$

In real world, however, the user has some preferences for media devices. And such user's preferences can vary from one user to another. Hence, we invoke priority ranking such that when many media devices supporting a chosen modality are available, only the top-ranked media device is activated. Also, priority ranking is essential in finding replacement to a failed media device. Usually, when a media device is malfunctioning or absent (a.k.a. *failed*), the system searches the device (that supports the same modality) which is next in priority. If it is found, the replacement device is activated and the search is over. Otherwise, the system keeps searching for a replacement through *priority ranking order*.

Given a media device D_i , where $1 \leq i \leq n$ and $i =$ priority index and $n =$ number of media devices supporting modality M_j , then the probability that D_i being adopted as the media device that will support modality M_j is given by:

$$P(D_i/M_j) = 1 - \sum_1^{i-1} (1/n) \quad (3.16)$$

Let there be a *media devices priority table* (MDPT) (see Tableau 3.1) containing media devices grouped according to the modality they support and arranged by priority ranking. When our system implements a modality, it selects the media device(s) that is/are ranked top in priority. It is also through the MDPT that the system searches for a replacement to a failed media device.

Tableau 3.1 A sample Media Devices Priority Table (MDPT).

Modality	Media Devices by Priority			
	1	2	3	n
Vocal input (V_{in})	microphone, speech recognition			
Vocal output (V_{out})	speaker, speech synthesis	headset		
Tactile input (T_{in})	keyboard	overlay; Braille terminal	Vmouse	
Tactile output (T_{out})	Braille terminal	tactile printer		

When a *new media device* 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 happens to the rankings of other devices d_i , ($1 \leq i \leq n$, and $n =$ number of media devices) which are in the same modality as 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 $\mathbf{Priority}(d_{new})$ set to r then the priority of the other device i , ($1 \leq i \leq n$) denoted $\mathbf{Priority}(d_i)$, 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: $\forall i, \exists r: \mathbf{r}; \forall d_i, \exists d_{new}: \mathbf{Devices} | (\mathbf{Priority}(d_{new}) = r \wedge \mathbf{Priority}(d_i) \geq r) \Rightarrow \mathbf{Priority}(d_i)' = \mathbf{Priority}(d_i) + 1$.

Given the familiar media devices, the association between modalities and media devices can be denoted as:

$$V_{in} = \text{Microphone} \wedge (\text{speech recognition}) \quad (3.17)$$

$$V_{out} = (\text{speech synthesis} \wedge \text{Speaker} \vee \text{Headset}) \quad (3.18)$$

$$T_{in} = \text{Keyboard} \vee \text{Braille Terminal} \vee \text{Overlay Keyboard} \vee \text{mouse} \quad (3.19)$$

$$T_{out} = \text{Braille Terminal} \vee \text{actile Printer} \quad (3.20)$$

3.3.4 Mathematical Expression and Presentation Formats

To a visually-impaired user, a *simple* mathematical expression becomes complex due to the presence of elements such as the *subscript*, *exponent*, *mathematical symbols* (e.g. Π , Σ , etc.), and the expression's *dimension* (i.e. complex numerator, denominator). In informatics, a mathematical expression is generally represented in *MathML* format which is an application of XML for describing mathematical notations and capturing its structure and content and aimed at integrating them into World Wide Web documents. For example, in Figure 3.1, a simple fraction is shown with its equivalence in MathML, in Braille and its linear representations.

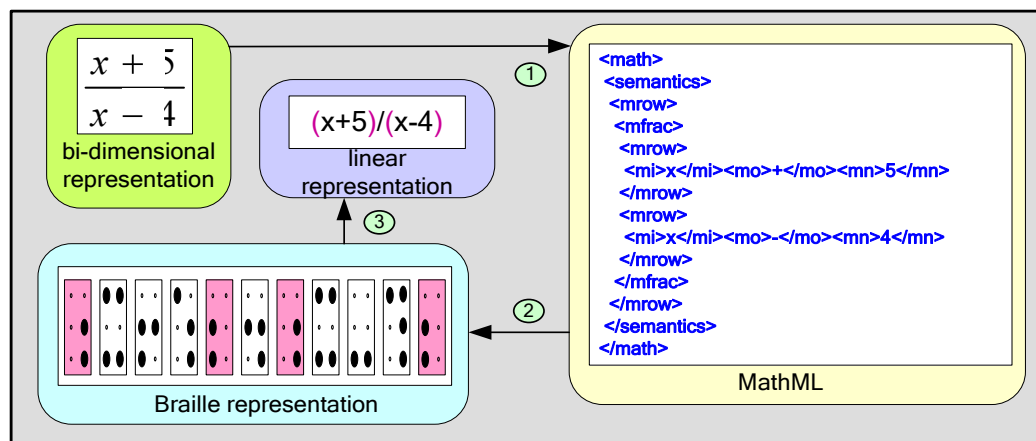


Figure 3.1 A fraction converted into its linear representation, and its corresponding MathML and Braille representations.

Different presentation formats use different methods to represent a mathematical operation. For example, in using *Braille*, there is a unique symbol for every operation. In using *speech*, an operation is uttered using a specific word (e.g. “+” is “plus”, “-” is “minus”, etc). Using *DotsPlus*, an operation is represented by a unique symbol in tactile form. Using *EasyMath*, every basic operation (e.g. +, -, ×, ÷, etc.) is represented by a unique symbol similar to its Braille representation. For *special operation* (e.g. Σ , Π , log, \int , \iint , \iiint , etc), however, its representation is in tactile form. Note that the representation of special operations in DotsPlus and EasyMath are not the same. For example, the + operation symbol is represented by a Braille symbol (see Figure 3.1) whereas in DotsPlus it is represented as “+” in embossed tactile form.

3.3.5 Analysis of a Mathematical Expression

A mathematical expression is analyzed as follows. Given a mathematical expression in MathML format, it is analyzed lexically using *grammar rules and dictionary*. The result yields a list of lexemes. A lexeme is a parameter within an expression which may be an operand or an operator. Given the lexemes, the *parser* analyzes the expression parameters (i.e. operands and operations) then sends the operation(s) to the *Expression Evaluator* to determine if special symbol is present while at the same time all the parameters are sent to the *Expression Encoder* to translate the expression into its encoded format. This process is shown in Figure 3.2.

As an example, the analysis of our specimen expression, the fraction defined in Figure 3.1, is shown in Figure 3.3. As shown, in step 1, the MathML expression is sent to the Lexer. In step 2, using the XML grammar, the expression is decomposed into a list of lexemes; the list is then sent to the parser. In step 3, the operations and operands in the expression are sent to expression evaluator and encoder. Together, in step 4, the evaluator deduces the presence of special symbol (e.g. fraction bar) while the encoder produces the encoded expression. Finally, in step 5, the presence of special operation symbol is detected and the encoded expression becomes an input to the presentation format selection process.

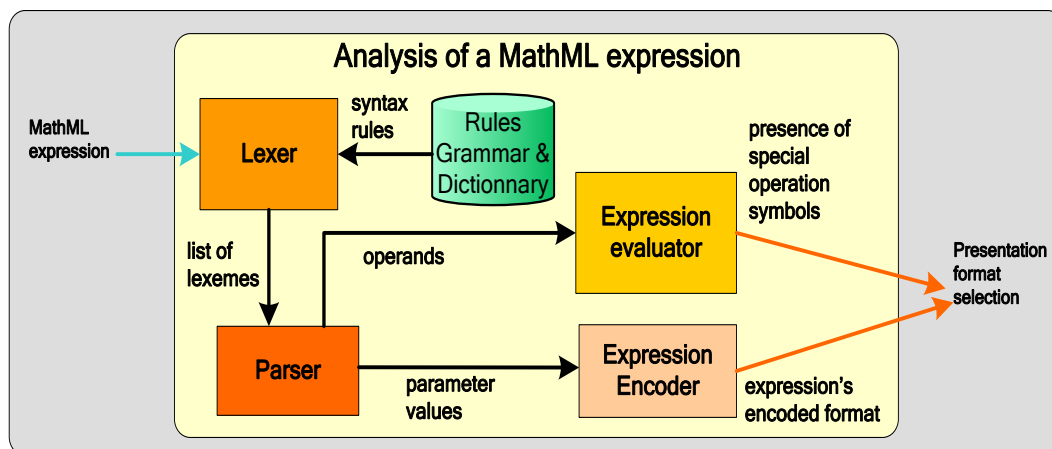


Figure 3.2 Structure of the analysis of a MathML expression.

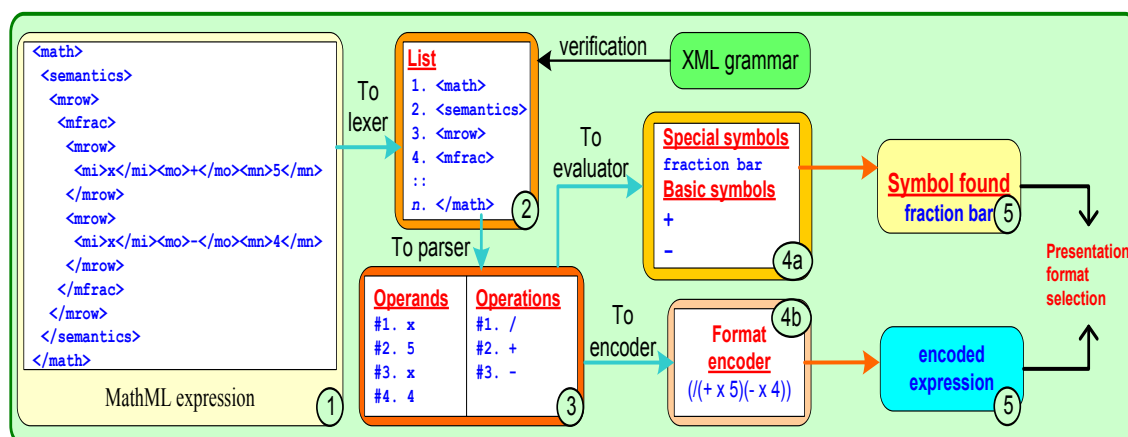


Figure 3.3 A sample analysis of a specimen fraction.

3.3.6 Selection of Presentation Format

In this work, we consider 4 presentation formats which are currently in existence, namely: the Braille, speech, DotsPlus and EasyMath.

The *Braille* format presents a mathematical expression in linear Braille representation. This format “linearizes” mathematics. Note that any bi-dimensional mathematical expression can be encoded as linear expression by using parentheses. A user must be familiar with Braille in

order to use this format. Some projects, such as VICKIE and BraMaNet, built translators that convert expression (in Latex or MathML format) to its Braille equivalent.

Speech, on the other hand, is the modality that is used daily by visually-impaired people. For mathematical application, speech converts an expression into its audio equivalent. Some tools that use speech as presentation format are MathTalk, Aster and AudioMath.

DotsPlus is an application that allows the printing of documents that contains mathematical symbols (e.g. \prod , \sum , etc.). These symbols are encoded as graphics shaped exactly as how they looked. When a blind user touches the embossed symbol, the user senses the bi-dimensionality of an expression. If the user intends to use DotsPlus, he must also be familiar with Braille since this specific format also uses Braille characters.

The objective of *EasyMath* is to produce a bi-dimensional output of a mathematical expression, similar to the representation for sighted people, using Braille characters and an overlay keyboard. EasyMath incorporates both audio and tactile forms. The use of EasyMath requires that the user uses an overlay keyboard and must be familiar with Braille.

The *media devices* are also factors in determining what presentation format should be used to present a mathematical expression. The four presentation formats, therefore, as functions of media devices, are given by:

$$\begin{aligned} \text{Speech presentation format} = & \text{Speech Synthesis)} \\ & \wedge \text{Speaker} \vee \text{Headset} \end{aligned} \quad (3.21)$$

$$\begin{aligned} \text{Braille presentation format} = & \text{Braille Terminal} \vee \\ & \text{TactilePrinter)} \end{aligned} \quad (3.22)$$

$$\begin{aligned} \text{EasyMath presentation format} = & \text{Overlay Keyboard} \\ & \wedge \text{actilePrinter) } \wedge \text{Speaker} \vee \text{Headset} \end{aligned} \quad (3.23)$$

$$\text{DotsPlus presentation format} = \text{actilePrinter} \quad (3.24)$$

As shown in Figure 3.4, the selection of the *suitable presentation format* begins with the given interaction context (step 1). The ML component, with reference to the modality SR (step 2), determines the optimal modalities (step 3). The selected optimal modalities as well as the available media devices (step 4) serve as input to the media selection process. With reference to the equations shown in Section 3.3.3, the selected media devices are determined (step 5). These devices along with the encoded expression (step 6-a), the presence/absence of special mathematical operation symbol (step 6-b), and the presentation formats available in the user’s computing system (step 6-c) are all inputs to the presentation format selection process. The output is the selected optimal presentation format. With reference to the specimen fraction expression in Figure 3.1, the selected presentation format is Braille. The diagram also illustrates the media devices that support Braille presentation format and the equation’s corresponding translated expression in various formats.

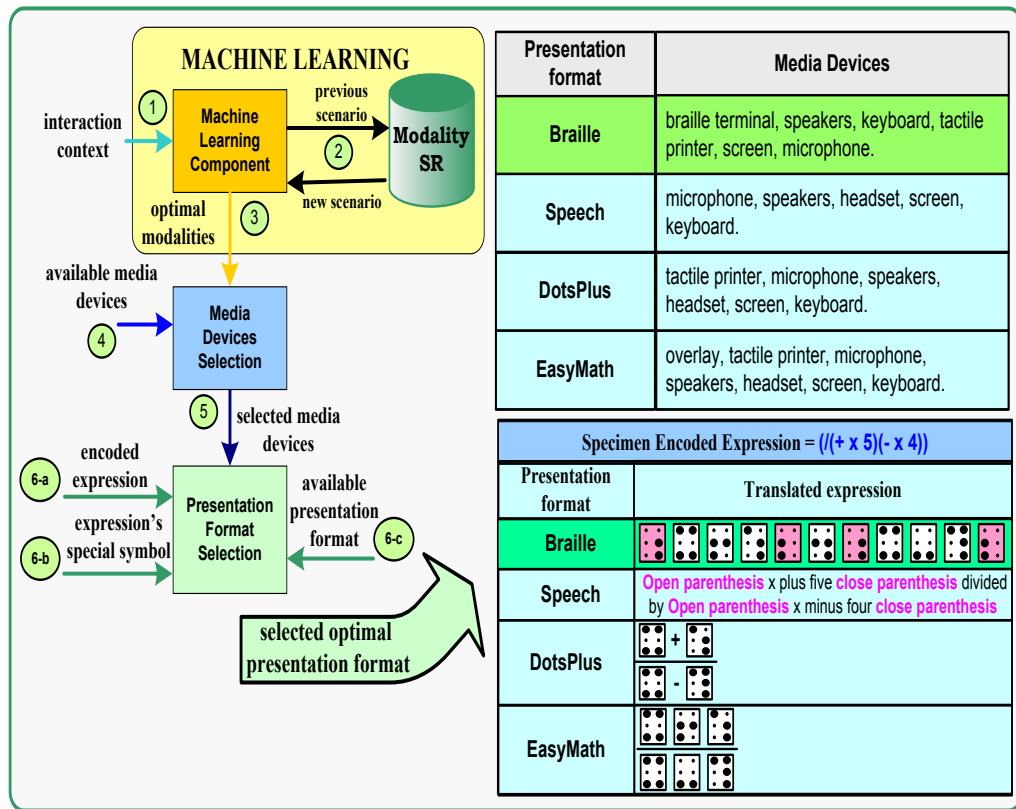


Figure 3.4 The presentation format selection process based on the given interaction context and MathML expression.

In the same figure, we also demonstrate the various *media devices that support other presentation formats* (namely, speech, DotsPlus and EasyMath).

Given selected modalities and available media devices, it is possible that two or more presentation format(s) may be suitable to a given MathML mathematical expression. How do we determine then the optimal presentation format? To resolve this issue, we adopt that the selection of optimal presentation format shall be based on user's preferences, on priority ranking that he/she assigned to different presentation formats and to media devices.

The MDPT is used in determining the priority of media devices as per modality that they support. Tableau 3.2 illustrates a sample of a *presentation format priority table (PFPT)*. As the table shows, there is a separate priority ranking for presentation format when the mathematical expression contains a special operation symbol. It also has another priority ranking for presentation format if the expression contains no special operation symbol. Examples of basic operation symbols include +, -, \times , \div , etc. Examples of *special operation symbols* include Σ , Π , log, \int , \iint , \iiint , etc.

Tableau 3.2 Presentation Format Priority Table.

Presentation Format by Priority	
Expression with special symbol	Expression without special symbol
1. EasyMath 2. DotsPlus 3. Braille 4. Speech	1. Braille 2. Speech 3. EasyMath 4. DotsPlus

The optimal presentation format selection process uses the following functions to determine the score of each presentation format. The one that obtains the highest score is considered as the optimal presentation format:

$$\text{Presentation Format Score} = \text{Presentation Format Priority} \times \text{Media Devices Priority} \quad (3.25)$$

$$\text{Presentation Format Priority} = (m + 1 - p) / m \quad (3.26)$$

$$\text{Media Devices Priority} = \frac{\sum_{i=1}^n (n_i + 1 - d_i) / n_i}{n} \quad (3.27)$$

such that for presentation format priority, the variable m = number of suitable formats and p = priority ranking as obtained from PFPT, $1 \leq p \leq m$. For priority of media devices, n = available media devices supporting a presentation format (see (21) through (24)). Also given i^{th} device, where $1 \leq i \leq n$, then d_i = priority ranking obtained from MDPT and n_i = number of media devices supporting the same modality as the i^{th} device. Figure 3.5 shows the algorithm that implements the selection of suitable presentation format.

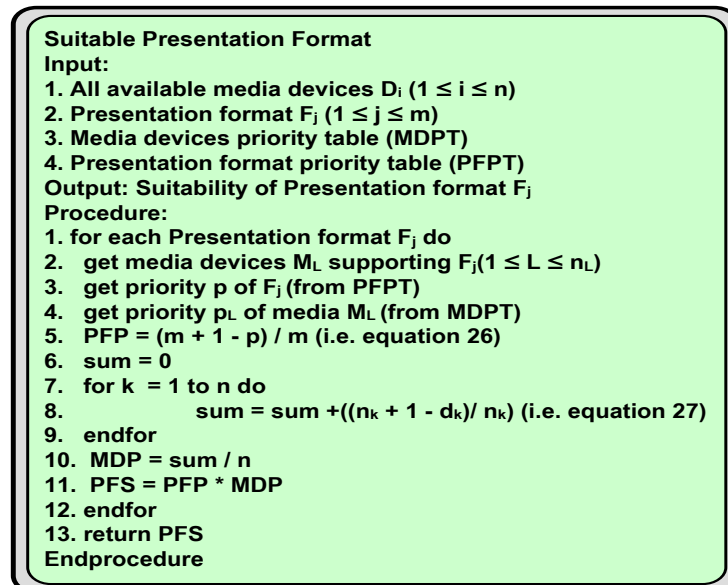


Figure 3.5 Algorithm for finding the suitable presentation format for a mathematical expression.

3.4 Detailed Design and Principles for the Resolution of System Requirements

The details of the infrastructure satisfying the design specifications cited earlier are explained in this section.

3.4.1 Architectural Framework

Figure 3.6 shows the layered view of our adaptive multimodal computing system for visually-impaired users. Adopting a layered architecture approach, one in which data moves from one defined level of processing to another, helps confine any error made during debugging to a specific layer and prevents the ripple effect of error propagation to other system components. The functionalities of each layer are given below:

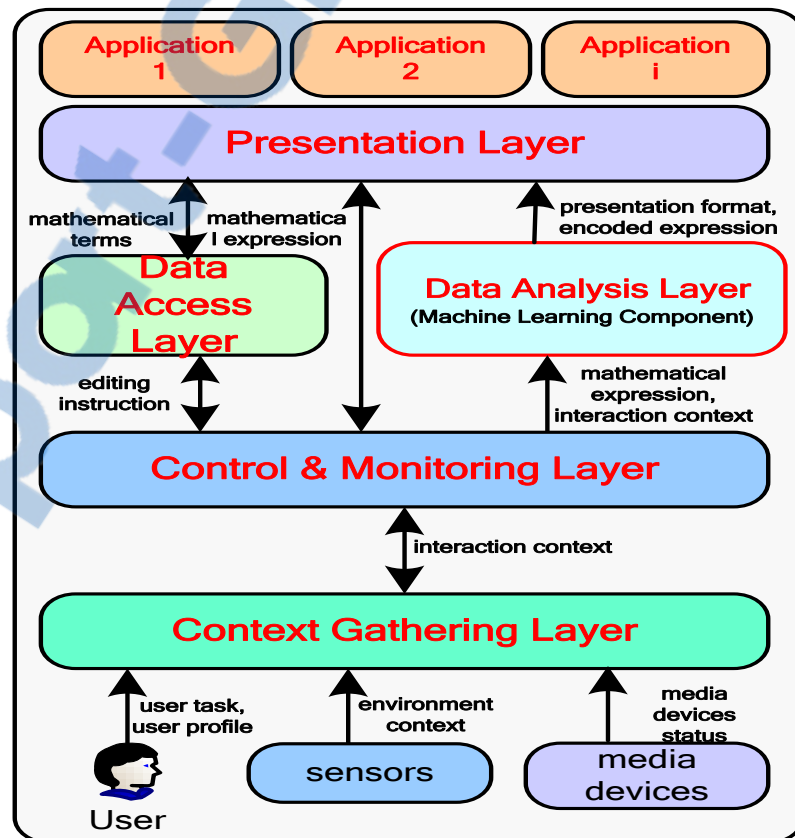


Figure 3.6 Architectural layer view of our multimodal computing system for visually-impaired users.

Context Gathering Layer : detects current interaction context;

Control and Monitoring Layer : controls the system, coordinates the detection of interaction context, the mathematical expression, its presentation and/or manipulation;

Data Analysis Layer : here, the presentation format of the mathematical expression is selected based on available resources and user's context;

Data Access Layer : allows search/edit of mathematical expression; and

Presentation Layer : presents mathematical expression via optimal presentation format.

An *agent* is a preferred programming technique because the traditional techniques (i.e. functional or object-oriented programming) are inadequate in developing tools that react to environment events. Recently, multi-agent systems (MAS) (Weiss, 1999; Wooldridge, 2002) have been widely used, from relatively small systems such as email filters up to large, open, complex, mission-critical systems such as air traffic control (Jennings et Wooldridge, 1998). Some works on MAS for visually-impaired users include (Awde et al., 2006a; Awde et al., 2006b). In contrast, in our work, our main goal is the correct representation of a mathematical expression while providing users autonomy. A multi-agent system is used to implement the architectural framework of Figure 3.6. The functionalities of various agents in our multimodal system are shown in a tabular format in Tableau 3.3.

3.4.2 Interaction Context Specification

Interaction context is formed by combining the context of the *user*, his *environment*, and his computing *system*. The *user context*, in this work, is a function of user profile (including any handicap) and preferences. A sample user profile, in generic format, is shown in Figure 3.7. The user's special needs determine other affected modalities (i.e. the user is already disqualified from using visual input/output modalities). For example, being mute prevents the user from using vocal input modality.

Tableau 3.3 Functionalities of Various Agents in Our Adaptive Multimodal Multimedia Computing System.

Layer	Agent	Functionality
Context Gathering Layer	User Agent	Detects user preferences, such as media and presentation format
	Environment Agent	Detects the user workplace's noise level and its noise restriction
	Device Manager Agent	Detects available and functional media devices
Control and Monitoring Layer	System Management Agent	Detects user's interaction context and mathematical expression, and determines how it will be treated
Data Analysis Layer	Analysis and Conversion Agent	Obtains MathML expression and converts it to its encoded format
	Machine Learning Agent	Determines the optimal presentation format for the given user situation
Data Access Layer	Searching and Editing Agent	Obtains editing instructions from the user
Presentation Layer	Translation Agent	Converts the encoded format of an expression into its final presentation format

Identity	Special Needs	Computing Device
username: <username>	Manually-Disabled : <Yes/No>	computing device1 : <MAC address 1>
	Mute : <Yes/No>	
password: <password>	Deaf : <Yes/No>	..
	Unfamiliarity with Braille: <Yes/No>	computing device n : <MAC address n>

Figure 3.7 A sample user profile.

As a function of modality, the user context **UC** can be represented by a single parameter, that of the user's *special needs*. This parameter is a 4-tuple representing additional handicaps,

namely the *manual disability*, *muteness*, *deafness*, and *unfamiliarity with Braille*. Each handicap affects user's suitability to adopt certain modalities. The convention for this special need is shown in Tableau 3.4. Hence, the failure of modality, adopted from (6), with respect to UC parameters is:

$$\begin{aligned} \text{Modality Failure} = & (\text{Manually-Disabled} \wedge \text{Mute}) \\ & \vee (\text{Manually-Disabled} \vee \text{Unfamiliar with Braille}) \\ & \wedge \text{Deaf}) \end{aligned} \quad (3.28)$$

Tableau 3.4 The Convention Table of User's Additional Handicap and Its Effect on User's Modality Disposition.

Convention No.	Manually Disabled	Mute	Deaf	Unfamiliarity with Braille	Inappropriate Modality
1	0	0	0	0	-
2	0	0	0	1	T _{out}
3	0	0	1	0	V _{out}
4	0	0	1	1	T _{out} , V _{out}
5	0	1	0	0	V _{in}
6	0	1	0	1	V _{in} , T _{out}
7	0	1	1	0	V _{in} , V _{out}
8	0	1	1	1	T _{out} , V _{in} , V _{out}
9	1	0	0	0	T _{out} , T _{in}
10	1	0	0	1	T _{out} , T _{in}
11	1	0	1	0	T _{in} , T _{out} , V _{out}
12	1	0	1	1	T _{in} , T _{out} , V _{out}
13	1	1	0	0	T _{in} , T _{out} , V _{in}
14	1	1	0	1	T _{in} , T _{out} , V _{in}
15	1	1	1	0	T _{in} , T _{out} , V _{out} , V _{out}
16	1	1	1	1	T _{in} , T _{out} , V _{out} , V _{out}

The *environment context EC* is the assessment of a user's workplace condition. To a blind user, a parameter such as light's brightness has no significance, while others, such as noise level, are significant.

In this work, the environment context is based on the following parameters: (1) the workplace's *noise level* – identifies if it is quiet/acceptable or noisy, and (2) the *environment restriction* – identifies whether a workplace imposes mandatory silence or not. Based on the

specified parameters, the environment context, therefore, is formally given by the relationship:

$$EC = (\text{Noise Level}) \wedge (\text{Environment Restriction}) \quad (3.29)$$

Tableau 3.5 shows the affected modality based on environment's context. It also shows the convention table we have adopted for **EC**.

Tableau 3.5 The Convention Table of Affected Modalities By Combined Noise Level and Environment Restriction.

Environment Context		Inappropriate Modality
Value	Convention	
1	(Environment Restriction = Silence Optional) \wedge (Noise Level = Noisy)	Vocal input
2	(Environment Restriction = Silence Optional) \wedge (Noise Level = Acceptable)	
3	(Environment Restriction = Silence Required) \wedge (Noise Level = Acceptable)	Vocal input, Vocal output (speaker)
4	(Environment Restriction = Silence Required) \wedge (Noise Level = Noisy)	Vocal input, Vocal output (speaker)

The unit of noise is *decibel* (dB). In our work, *50 dB or less* is considered “*acceptable*” or “*quiet*” while *51 dB or more* is considered “*noisy*”. In our system, this range can be modified, using user interface, by the end user based on his perception. The noise level is the result of an interpretation from sampled raw data taken from a sensor. An example of a noise-detection sensor is *PASPORT PS2100*³¹. In general, when the user's workplace is noisy, the effectiveness of vocal input modality is doubtful; hence an alternative modality is necessary.

In an environment where silence is required, sound-producing media (e.g. speaker) needs to be muted or deactivated. For an environment's noise restriction, we have defined a database of pre-defined places (e.g. library, park) and their associated noise restrictions (e.g. library:

³¹ PASPORT PS2100 Noise Detector <http://store.pasco.com/pascostore/>

silence required; park: silence optional). User can modify some database records. Also, new ones can be added through the user interface.

In our work, the *system context (SC)* implies the user's computing device and the available media devices. The system context is managed by the *Device Manager Agent (DMA)*. See Figure 3.8. The computing device (e.g. PC, laptop, PDA, cellular phone) also affects the modality selection. For example, using a PDA or cell phone prevents the user from using tactile input or output modality. Some of the most commonly-used media devices suitable for blind users are: (i) *Keyboard*; (ii) *Microphone*; (iii) *Speech Recognition*; (iv) *Speech Synthesis*; (v) *Speaker*; (vi) *Headset*; (vii) *Braille Terminal*; (viii) *Overlay or Concept Keyboard*; and (ix) *Tactile Printer or Embosser*.

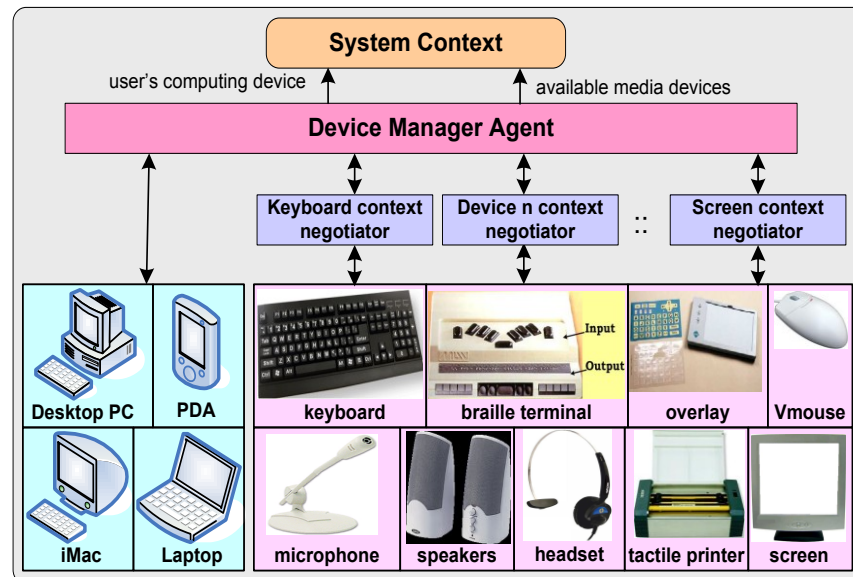


Figure 3.8 The DMA, responsible for detecting user's computing device and available media devices.

Let **SC**, for the purpose of modality selection, be represented by a single parameter, the user's computing device. Let **Computing Device** = {PC, MAC, Laptop, PDA, Cellular phone}. The computing device convention is shown in Tableau 4.6. For example, **SC** = 1 means that the user's computer is either a PC, a laptop or a MAC. Note that when **SC** = 2 (i.e. PDA), $T_{in} =$

Failed because the computing device has no tactile input device; its $T_{out} = \text{Failed}$ because, in a regular set-up, it is not possible to attach a tactile device (e.g. Braille terminal) onto it.

Tableau 3.6 The Convention Table of User's Computing Device and Its Effect on Modality Selection.

Convention No.	Computing device	Inappropriate Modality
1	PC/MAC/Laptop	-
2	PDA	T_{in}, T_{out}
3	Cellular Phone	T_{out}

Using ML, the following are the derived formulas on modality failures given the parameters of interaction context:

$$V_{in} \text{ Failure} = \text{User} = \text{Mute}) \vee \text{Noise Level} = \text{Noisy}) \vee \text{Environment Restriction} = \text{Silence required}) \quad (3.30)$$

$$V_{out} \text{ Failure} = \text{User} = \text{Deaf}) \quad (3.31)$$

$$T_{in} \text{ Failure} = \text{User} = \text{Manually - Disabled}) \vee \text{Computing Device} = \text{PDA}) \quad (3.32)$$

$$T_{out} \text{ Failure} = \text{User} = \text{Manually - Disabled}) \vee (\text{User} = \text{Unfamiliar with Braille}) \vee (\text{Computing Device} = \text{PDA}) \vee (\text{Computing Device} = \text{Cellphone}) \quad (3.33)$$

3.5 Formal Specification, Simulation and Examples

Here, we cited examples to illustrate how equations in previous sections are used. Formal specifications using Petri Net are also demonstrated.

3.5.1 Formal Specification and Petri Net

A *formal specification* is a mathematical description of a system that may be used to develop its implementation. Through specification, formal verification techniques can be used to demonstrate that the system design is correct with respect to the given specification.

*Petri Net*³² is a formal, graphical, executable technique for the specification and analysis of a concurrent, discrete-event dynamic system. Petri nets are used in deterministic and in probabilistic variants; they are good means to model concurrent or collaborating systems. They also allow for different qualitative or quantitative analysis that can be useful in safety validation.

A Petri-Net diagram is represented by an ellipse called *place* (i.e. a state), a rectangle called *transition* (i.e. a process) and an arc representing *input* for a transition to take place (either from a place to a transition, or from a transition to a place). Places can contain *tokens*; the current state of the modeled system (the *marking*) is given by the number of tokens (and type, if they are distinguishable) in each place. Transitions are active components. They model activities which can occur (i.e. the transition fires). When the transition fires, it removes tokens from its input places and adds some to all its output places. The number of tokens removed/added depends on the cardinality of each arc.

A Petri Net specification can be defined as a quadruple (P, T, F, B) , where P = a non-empty set of places, T = a non-empty set of transitions, $F: P \times T \rightarrow \mathcal{E}$ is the forward incidence function, $B: P \times T \rightarrow \mathcal{E}$ is the backward incidence function and \mathcal{E} = the set of integers ≥ 0 . F and B can be represented by forward and backward incidence matrices $C(F)$ and $C(B)$. A marking of a Petri Net is a mapping $M: P \rightarrow \mathcal{E}$. The “*firing*” of any transition changes the marking of the Petri Net.

³² <http://www.winpesim.de/petrinet>. Consulté le 12/12/2008.

In specification using Petri Net, we capture only the simulations' final results. In general, however, when a dynamic event is simulated, all possible outcomes can be obtained but due to space constraints, only a snapshot of one of many outcomes is shown. We use HPSim³³ to create, edit and simulate Petri Nets. The Petri Net diagram illustrates all possible cases that could arise, given the places and transitions of a scenario. A snapshot of one of these possible cases is shown in each of the following simulations. The diagrams are mechanisms to validate the results obtained from the simulation.

3.5.2 Sample Case 1

Consider, for example, an interaction context that is composed of the following parameters:

$\mathbf{IC} = (A_1, A_2, A_3, A_4, A_5, A_6, A_7)$ wherein

$A_1 = \{\text{true} \mid \text{false}\}$ = if user is manually disabled,

$A_2 = \{\text{true} \mid \text{false}\}$ = if user is mute,

$A_3 = \{\text{true} \mid \text{false}\}$ = if user is deaf,

$A_4 = \{\text{true} \mid \text{false}\}$ = if user is familiar with Braille,

$A_5 = \{\text{quiet} \mid \text{noisy}\}$ = environment's noise level,

$A_6 = \{\text{silence required} \mid \text{silence optional}\}$

= environment's noise level restriction, and

$A_7 = \{\text{PC or Laptop or MAC} \mid \text{PDA} \mid \text{Cell phone}\}$

= user's computing device.

The set of possible modalities (i.e. refer to (5)) is given by $M = \{M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9\}$ wherein $M_1 = \{T_{in}, T_{out}\}$; $M_2 = \{T_{in}, V_{out}\}$; $M_3 = \{V_{in}, V_{out}\}$; $M_4 = \{V_{in}, T_{out}\}$; $M_5 = \{V_{in}, T_{out}, V_{out}\}$; $M_6 = \{T_{in}, T_{out}, V_{out}\}$; $M_7 = \{T_{in}, V_{in}, V_{out}\}$; $M_8 = \{T_{in}, V_{in}, T_{out}\}$; $M_9 = \{T_{in}, V_{in}, T_{out}, V_{out}\}$. In this example, let us assume the following interaction context: (i) user context: blind with no further handicaps, familiar with Braille; hence $A_1 = \text{False}$, $A_2 = \text{False}$, $A_3 = \text{False}$, $A_4 = \text{True}$ (ii) environment context: the user is in a classroom, then $A_5 = \text{noisy}$, A_6

³³ HPSim. <http://www.winpesim.de>

= silence required (iii) system context: the user works on a laptop; A_7 = Laptop. Also, the user wishes to read the following expression:

$$\int_2^3 \frac{x+1}{x-1} dx \quad (3.34)$$

The system now finds the modality that suits the given interaction context. The system does so using the principles discussed in Section 3.3.2. Let us assume that a certain multimodal computing system's SR contains recorded scenarios as shown in Figure 3.9. The given figure is generated by using WEKA (*Waikato Environment for Knowledge Analysis*) (Witten et 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.

No.	A1	A2	A3	A4	A5	A6	A7	class
	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	Nominal	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 3.9 A sample snapshot of a scenario repository (SR).

As shown in the diagram, there are already 15 scenarios representing the system's acquired knowledge. The 16th scenario represents a new case. Using (14), and with reference to the given interaction context and SR, the suitability score of M_j (where $j = 1$ to 9) can be calculated. Let us consider, for instance, the calculations involved with modality M_1 :

$$\begin{aligned} \text{Suitability_Score}(M_1) &= P(A_1 = \text{False} \mid M_1) \times P(A_2 = \text{False} \mid M_1) \times \dots \times P(A_7 = \text{Laptop} \mid M_1) \times P(M_1) \\ &= 1 \times 0.5 \times 0 \times \dots \times 2/15 = 0. \end{aligned}$$

wherein $P(A_1 = \text{False} \mid M_1) = 2/2$, $P(A_2 = \text{False} \mid M_1) = 1/2$, $P(A_3 = \text{False} \mid M_1) = 0/2$, $P(A_4 = \text{True} \mid M_1) = 2/2$, $P(A_5 = \text{Noisy} \mid M_1) = 1/2$, $P(A_6 = \text{silence required} \mid M_1) = 1/2$, and $P(A_7 = \text{Laptop} \mid M_1) = 1/2$. Also, $P(M_1) = 2/15$.

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 :

$$\begin{aligned} \text{Suitability_Score}(M_6) &= P(A_1 = \text{False} \mid M_6) \times P(A_2 = \text{False} \mid M_6) \times \dots \times P(A_7 = \text{Laptop} \mid M_6) \times P(M_6) \\ &= 1 \times 2/3 \times 1 \times 1 \times 2/3 \times 1/3 \times 1/3 \times 3/15 = 0.00976. \end{aligned}$$

By applying the ML algorithm (see Figure 3.13), M_6 appears to respect the conditions imposed in (5), 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).

3.5.3 Simulation 1

As shown in Figure 3.10, the combination of interaction context's parameters yields the implementation of some modalities (i.e. M_1, \dots, M_9). The Net in the diagram illustrates the snapshot simulation of the case cited in sub-section 3.5.2. As shown, the simulation begins with a token in "Modality" place and "Interaction Context" place. The firing of the token in Interaction Context yields a specific value for "User Context", "Environment Context" and "System Context based on Computing Device" places, which is exactly similar to the values of A_1, \dots, A_7 . The traversal of tokens in different places is noted by green coloured places. As shown, the result yields modality M_6 being selected as the optimal modality. The Petri Net

simulation confirms the result obtained in the previous section. Also, the same case yields a V_{in} failure result (i.e. due to noisy environment).

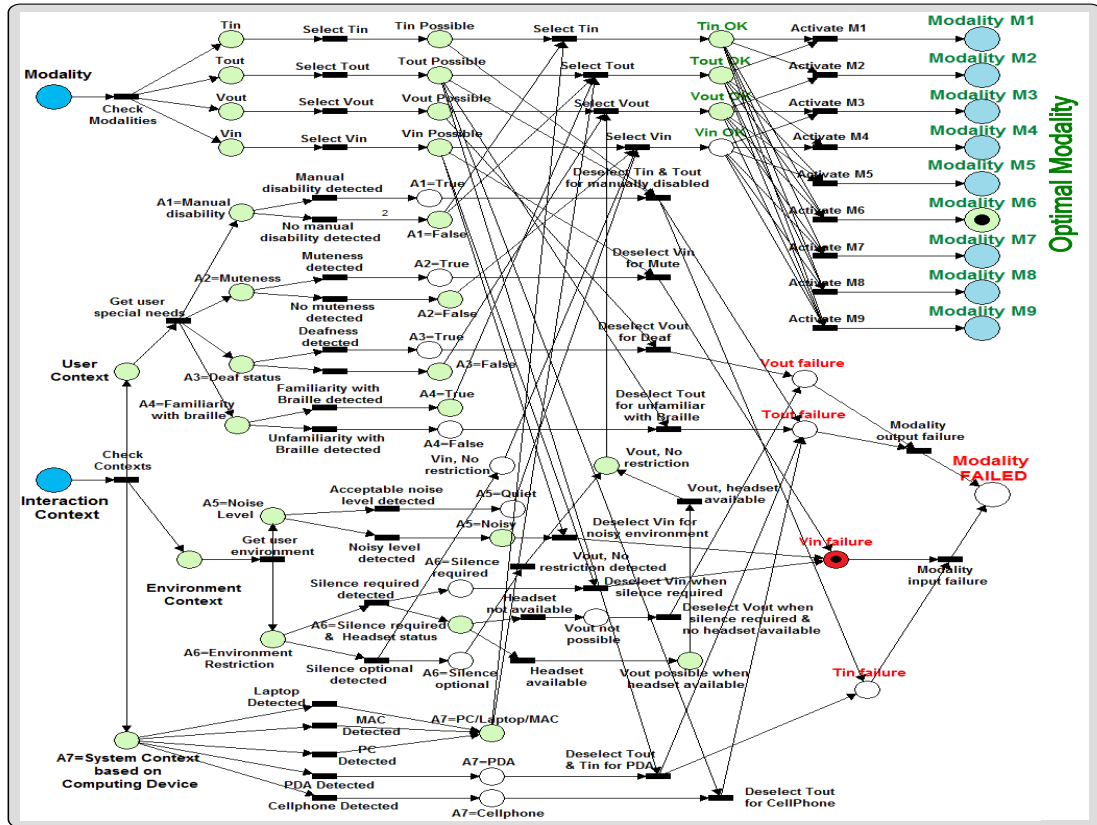


Figure 3.10 Snapshot of the simulated interaction context case in Section 3.3.

3.5.4 Sample Case 2

Now, consider further that the currently available media devices and their corresponding priority rankings are as shown in Tableau 3.1. Also, assume that the priority rankings of user’s preferred presentation formats are as shown in Tableau 3.2. Based on (21) through (24), the results indicate that all four presentation formats are possible. To determine the optimal presentation format, (25) and Tableaux 3.1 and 3.2 are used to calculate the scores of each format. The calculation results are as follows: (i) $EasyMath = (4/4) \times (1 + (2/3) + 1) / 3 = 8/9$, (ii) $Braille = (3/4) \times 1 = 3/4$, (iii) $DotsPlus = (2/4) \times 1 = 1/2$ and (iv) $Speech = (1/4) \times 1 = 1/4$. Hence, the optimal selection is *EasyMath*.

3.5.5 Simulation 2

After determining the optimal modality (i.e. M_6) in the previous simulation, the system next determines the availability of the presentation formats and the media devices that will support the selected optimal modality. In the Petri Net simulation diagram of Figure 3.11, the status of some media devices (e.g. overlay keyboard, Braille terminal, etc.) are simulated (i.e. either OK or failed). The presentation formats are either existing or not, but for consistency purposes with the cited case, the presentation formats are simulated as all available. The diagram indicates under which conditions a specific presentation format may be availed. In the simulated diagram, the Braille, speech, DotsPlus and EasyMath presentation formats are possible. As stated, the final selection depends on user's preferences and the available media devices. In this simulated case, EasyMath obtained the highest score hence it is selected as the optimal presentation format for the given mathematical expression (see marked place).

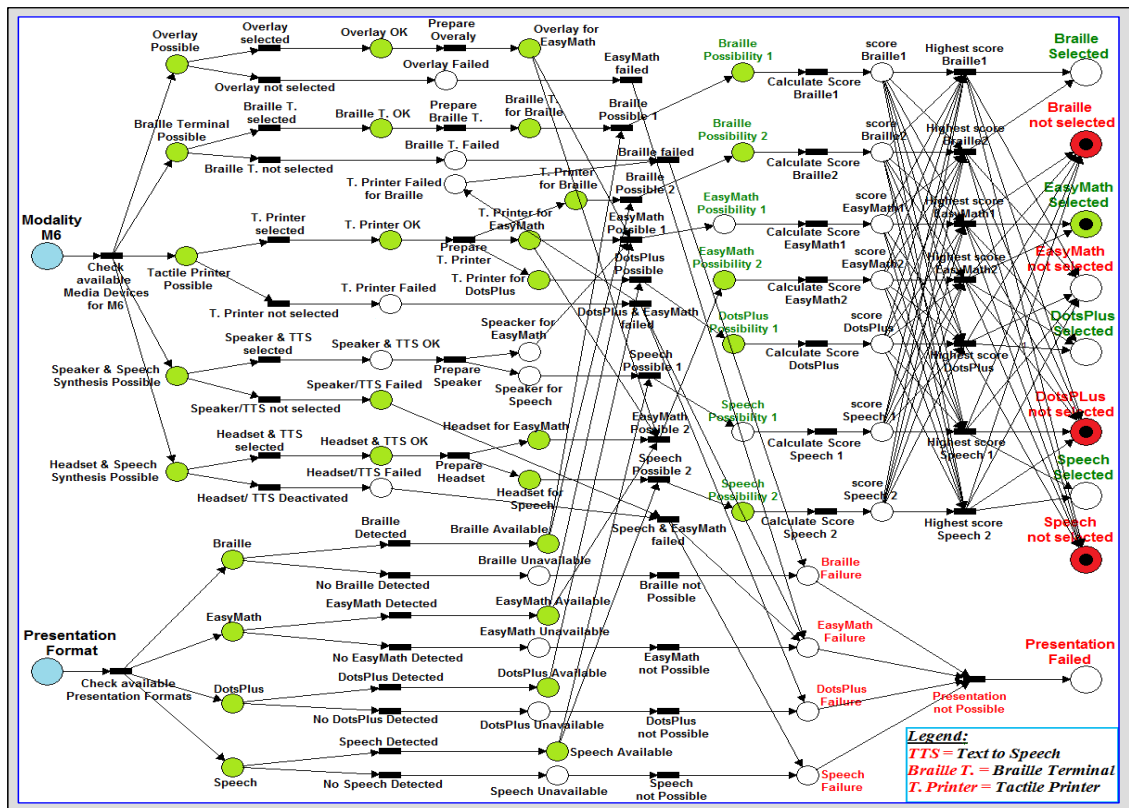


Figure 3.11 Selection of the optimal presentation format based on available media devices and presentation formats.

3.6 Conclusion

Our ongoing research focuses on providing computing infrastructure to visually-impaired users through multimodality. One area of such domain is the infrastructure for mathematical presentation to blind users which this paper addresses. In this paper, we have noted the weaknesses of the current state-of-the art solutions. We note that the currently available solutions present mathematical expressions in one presentation format alone with no consideration on the context of the user, his working environment and his computing system as well as the nature of the mathematical expression itself and of the user's preferences. Moreover, these solutions do not provide visually-impaired users with autonomy as the users still need the assistance of sighted people.

To address the weaknesses cited above, we propose an infrastructure of a multimodal system that allows the presentation of mathematical expressions in as many presentation formats as possible. The proposed solution considers the user's interaction context (i.e. combined user's, environment's and system's context) as well as the nature of the mathematical expression itself and of the user's preferences. The proposed solution is most efficient if the user is familiar with different presentation formats (e.g. Braille, speech, DotsPlus, EasyMath). Indeed, the proposed solution is apt for the given user's context and capacity.

Our future works involve the prototyping of this infrastructure and simulating its performance using several computing platforms. Such prototype will also be tested on visually-impaired users with varying interaction context.

3.7 Appendix A - figures on optimization algorithms

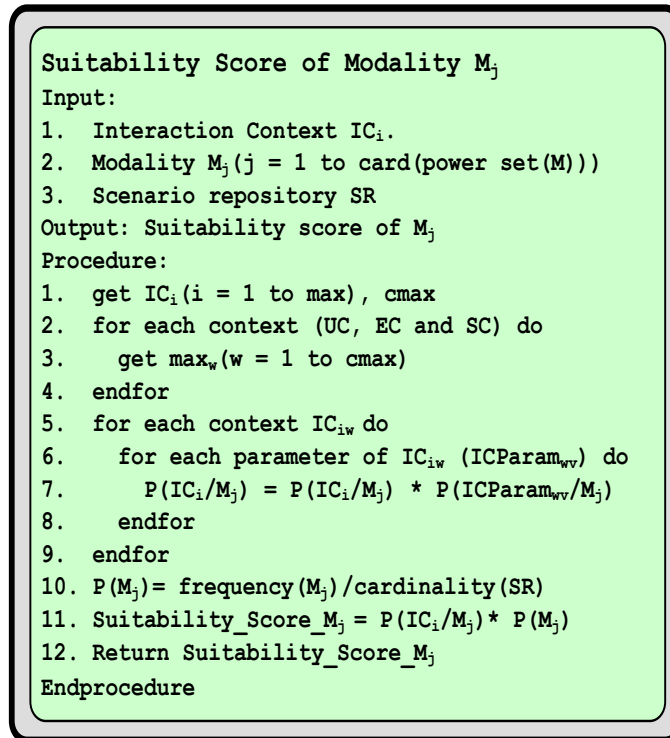


Figure 3.12 Algorithm that calculates for the suitability score of each element M_j of the modality power set, given an interaction context IC_i .

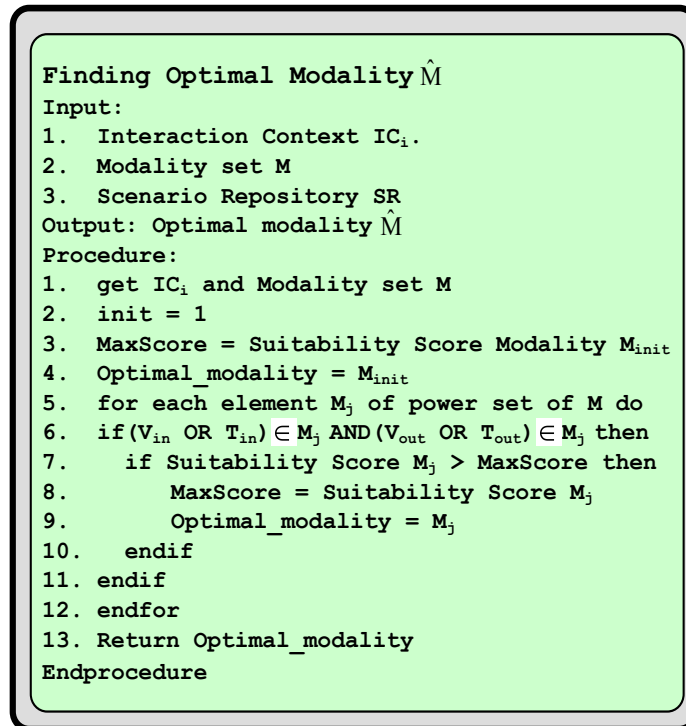


Figure 3.13 Algorithm for finding the optimal modality.

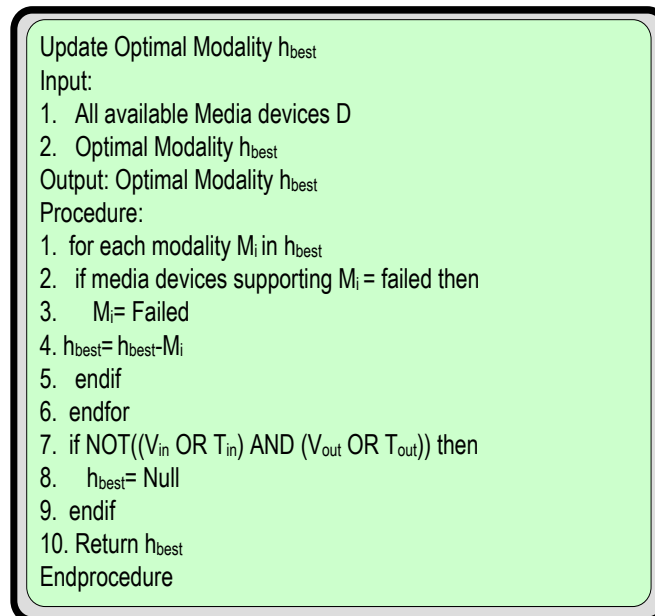


Figure 3.14 Algorithm for updating optimal modality based on the availability of media devices to support the chosen modalities.

3.8 Acknowledgment

This research work has been made possible through the funding provided by the *Natural Sciences and Engineering Research Council of Canada* (NSERC), and the scholarship funds from the *National Bank of Canada* and the *Décanat des études* of *École de technologie supérieure* and *La bourse de cotutelle* provided by the governments of Quebec and France.

3.9 References

- Alpaydin, E. 2004. *Introduction to Machine Learning*. Cambridge, Massachusetts: MIT Press.
- Awde, A., M. D. Hina, C. Tadj, Y. Bellik et A. Ramdane-Cherif. 2006a. « A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users ». In *The First International Conference on Grid and Pervasive Computing (GPC)* (Taichung, Taiwan, 3-5 May 2006). p. 620-633. Lecture Notes in Springer-Verlag.
- Awde, A., M. D. Hina, C. Tadj, A. Ramdane-Cherif et Y. Bellik. 2006b. « Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users ». In *ISIE 2006, IEEE International Symposium on Industrial Electronics* (Montreal, QC, Canada, July 9-13, 2006).
- Cahill, H., C. Lineham, J. McCarthy, G. Bormans et J. Engelen. 1995. « Ensuring Usability in MATHS ». In *The European Context for Assistive Technology*. p. 66-69. Amsterdam: IOS Press.
- Edwards, A. D. N., et R. D. Stevens. 1994. « A Multimodal Interface for Blind Mathematics Students ». In *INSERM'94*. p. 97 - 104. Paris, France.
- Ferreira, H., et D. Freitas. 2004. « Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project ». In *9th International Conference on Computer Helping People with Special Needs (ICHP)*. p. 678-685. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Ferreira, H., et D. Freitas. 2005. « AudioMath: Towards Automatic Readings of Mathematical Expressions ». In *Human-Computer Interaction International (HCII)*. Las Vegas, Nevada, USA.

- Garlini, P., et F. Fogarolo. 2003. *LAMBDA: Linear Access to Mathematics for Braille Device and Audio Synthesis - Analysis of the User's Needs*. Padova, Italy: University of Padova.
- Jennings, N. R., et M. J. Wooldridge. 1998. « Applications of Intelligent Agents ». In *Agent Technology: Foundations, Applications, and Markets*, sous la dir. de Jennings, Nicholas R., et Michael J. Wooldridge. p. 3-28. Springer-Verlag: Heidelberg, Germany.
- Kallenberg, O. 2002. *Foundations of Modern Probability*. Springer, 650 p.
- Lightfoot, D. 2001. *Formal Specification Using Z*, 2nd. McMillan Press.
- Mitchell, T. 1997. « Machine Learning ».
- Moço, V., et D. Archambault. 2003. « VICKIE: A Transcription Tool for Mathematical Braille ». In *7th European Conference for the Advancement of Assistive Technology in Europe (AAATE)* (September 2003). Dublin, Ireland: IOS press.
- Podevin, A. 2002. « Accès aux formules mathématiques par des personnes non-voyantes : étude et définition d'une méthode adaptée ». DEA, Université de CAEN.
- Preddy, M., J. Gardner, S. Sahyun, et D. Skrivanek. 1997. « Dotsplus: How-to Make Tactile Figures and Tactile Formatted Math ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA
- Raman, T. V. 1998. *Audio System for Technical Readings*, 1410. Berlin, Germany: Springer-Verlag.
- Schwebel, F., et R. Goiffon. 2005. « BraMaNet : Quelques règles simples à connaître pour qu'un aveugle puisse lire vos documents mathématiques et vos pages web ». In *Journées nationales Caen* (Caen, France, October 2005).
- Stöger, B., K. Miesenberger et M. Batusic. 2004. « Mathematical Working Environment for the Blind Motivation and Basic Ideas ». In *9th International Conference on Computers Helping People with Special Needs (ICCHP)*. p. 656-663. Springer Verlag, Lecture Notes in Computer Science (LNCS).
- Weiss, G. 1999. « Multiagent systems ». MIT-Press.
- Witten, I. H., et E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed.* San Francisco: Morgan Kaufmann, 525 p.
- Wooldridge, M. 2002. *An Introduction to Multi-agent Systems*. Chichester, UK: Wiley.

CHAPITRE 4

COMPLEXITY OF MATHEMATICAL EXPRESSIONS IN ADAPTIVE MULTIMODAL MULTIMEDIA SYSTEM ENSURING ACCESS TO MATHEMATICS FOR VISUALLY IMPAIRED USERS

Ali Awde¹, Yacine Bellik², Chakib Tadj¹

¹Département de Génie Électrique, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

²Laboratoire LIMSI-CNRS, Université de Paris-Sud
B.P. 133, 91043 Orsay, France

Ce chapitre a fait l'objet d'une publication dans la revue International Journal of Computer and Information Science and Engineering (IJCISE), Vol. 2, No. 2, 2008.

Résumé

Notre système adaptatif permet de présenter correctement les expressions mathématiques aux utilisateurs non-voyants. Partant du contexte de l'interaction (utilisateur, environnement, système) ainsi que de la complexité de l'expression elle-même et des préférences de l'utilisateur, le système calcule le degré de convenance de chaque forme de présentation. La ou les forme(s) qui obtiennent le plus haut score seront choisies pour présenter l'expression à l'utilisateur. Notre nouvelle méthode est plus efficace que les solutions proposées dans la littérature car elle prend en considération la situation de l'utilisateur et elle n'impose pas une forme de présentation figée non appropriée au contexte de l'interaction.

Dans cet article, nous présentons notre méthodologie pour calculer la complexité d'une expression mathématique ainsi que les résultats de nos expérimentations. Finalement, cet article montre comment nous avons intégré la complexité de l'expression dans notre système et les concepts utilisés pour le valider.

Mots clés: Système adaptatif, système multi-agent, mathématiques pour les non-voyants, système multimodal, interaction homme-machine.

Abstract

Our adaptive multimodal system aims at correctly presenting a mathematical expression to visually impaired users. Given an interaction context (i.e. combination of user, environment and system resources) as well as the complexity of the expression itself and the user's preferences, the suitability scores of different presentation format are calculated. Unlike the current state-of-the art solutions, our approach takes into account the user's situation and not imposes a solution that is not suitable to his context and capacity. In this work, we present our methodology for calculating the mathematical expression complexity and the results of our experiment. Finally, this paper discusses the concepts and principles applied on our system as well as their validation through cases studies. This work is our original contribution to an ongoing research to make informatics more accessible to handicapped users.

Keywords: Adaptive system, intelligent multi-agent system, mathematics for visually-impaired users.

4.1 Introduction

An adaptive computing system is a system that is capable to adapt its behavior according to changes in its environment or on user's situation or in parts of the system itself. Hence, the system can reason and react with no or very little human intervention.

A multimodal system allows user to interact with more than one mode of interaction. Indeed, incorporating *multimodality* into a computing system makes it more accessible to a wider range of users, including those with impairments. With multimodality, the strength or weakness of a modality or media device is decided based on its suitability to a given context. For example, visual modality (screen) is not appropriate to visually impaired users hence tactile modality (Braille terminal) or speech or both (using many media concurrently) can replace it.

Making mathematics accessible to visually impaired people is a significant challenge, due to the following reasons: First, the visual mathematical representation is bi-dimensional and the interpretation of a mathematical expression is related to one's knowledge of the expression's spatial components. Second, the conversion of a multi-dimensional structure to a non-visual representation (e.g. Braille) requires further information to explain a term (for example, an exponent) so the blind users can understand the expression correctly. Also, the conversion into an audio format is often ambiguous. Third, the vocabulary terms used by sighted people are quite large as compared to the amount of data that can be made accessible to blind people. For example, traditional Braille utilizes 6-dot character, which allows 64 possible characters. This quantity of characters is not enough to represent all frequently-used mathematical symbols (e.g. "X" is represented by a sequence of two characters: "x"). Also, large quantity of symbols is a challenge to a blind user.

The current state-of-the art proposes systems and solutions that translate mathematical expressions into other presentation formats. For instance, some systems convert a standard algebraic expression into its audio equivalent, while others convert a mathematical document into its Braille representation. Yet others produce mathematical expression's equivalent and present them as printed document in which characters are embossed and meant to be touched. Our work has been conceptualized after having reviewed the current state-of-the-art systems and solutions and found out that most, if not all, of available systems for presentation of mathematical expressions to visually-impaired users do not provide the user with the desired autonomy; that most of these systems do not take into consideration the current interaction context in their system's configuration and that the prevailing systems provide only one choice of presenting mathematical expressions, meaning that they are not multimodal. In other word, our work addresses the weaknesses that we found in the current state-of-the-art systems and solutions.

In this research, we take into account the current interaction context and the nature of expression because we have noted that the choice of presentation format varied also based on the nature of the expression. Various previous works (Awde et al., 2007) treated the interaction context. In this paper, we concentrate on studying the complexity of an

expression. We demonstrate here that the parameters that constitute expression context cannot be fixed or pre-defined. They have to be learned based on many parameters. In this regard, our approach is based on observations and a mathematical model to define the expression complexity. Once the complexity is calculated, it can be integrated to other contextual information in order to find the most appropriate presentation format for each specific visually impaired user. This last step is not addressed in this paper.

In this paper, we present the infrastructure of a multimodal computing system that presents mathematical expressions to visually-impaired users by taking into account his interaction context, preferences and the nature of the expression itself. The rest of this paper is structured as follows. Section 4.2 presents related researches and highlights the novelty of our work; Section 4.3 presents the purpose, objectives and questions of the work. Section 4.4 defines media, modality and presentation format. Section 4.5 presents the design of our system and the contextual information. In section 4.6, we present expression components and a mathematical taxonomy. Section 4.7 presents experiment protocol and running experiment. In section 4.8, we present and argue the results of the experiment. In section 4.9, we apply the concepts experimented in the previous section. Finally, we present our future works and conclusion in section 4.10.

4.2 Related Work

To a visually-impaired user, understanding mathematical expression requires repeated passages over the expression, sometimes skipping some secondary information, only to reverse back to it again and again until the user fully grasps the expression. A complicated task like this is detailed in (Stöger, Miesenberger et Batusic, 2004). Some tools, however, have been developed to lessen the complexity of performing a similar task, among them being Mathtalk (Edwards et Stevens, 1994), Maths (Cahill et al., 1995), DotsPlus (Preddy et al., 1997), EasyMath (Podevin, 2002) and AudioMath (Ferreira et Freitas, 2004; 2005). In Maths, and MathTalk the user can read, write and manipulate mathematics using a multimedia interface containing speech, Braille and audio. VICKIE (*Visually Impaired*

Children Kit for Inclusive Education) (Moço et Archambault, 2003a) and BraMaNet (*Braille Mathématique sur InterNet*) (Schwebel et Goiffon, 2005) are transcription tools that convert mathematical document (written in LaTeX, MathML, HTML, etc.) to its French Braille representation. Labradoor (*LaTeX-to-Braille-Door*) (Batusic, Miesenberger et Stöger, 1998) converts an expression in LaTeX into its Braille equivalent using the Marburg code. MAVIS (*Mathematics Accessible To Visually Impaired Students*) (Karshmer et al., 1998) supports LaTeX to Braille translation using Nemth code. DotsPlus is a tactile method of printing documents that incorporates both Braille and graphic symbols (e.g. \prod , \sum , etc.). For EasyMath (Podevin, 2002), the objective is to produce a bi-dimensional output of a mathematical expression, similar to the representation for sighted people, using Braille characters and an overlay keyboard.

As a background, some special Braille notations have been developed for mathematics. Also, different Braille code notations for mathematics are available depending on the country. Some of these codes are the Nemeth Math code (Nemeth, 1972) which is used in the USA, Canada, New Zealand, Greece and India; the Marburg code (Epheser, Pograniczna et Britz, 1992) used in Germany and Austria, the French Math code (*Commission Evolution du Braille Francais*, 2001).

None of the tools cited above, however, is complete. Studies were conducted evaluating these tools based on users' needs (Garlini et Fogarolo, 2003; Podevin, 2002). Results indicate that users are neither independent nor able to do their homework (i.e. case of students) without the help of sighted people. Indeed, each tool has its own set of limitations. For example, Aster (*Audio System for Technical Readings*) (Raman, 1998) uses only a LaTeX document, and AudioMath uses only a MathML document, in its conversion to its equivalent vocal output. Our approach, therefore, is to get the strength of each tool, integrate each one of them into our work in order to build a system that (1) broadens the limits of utilization, (2) provides the user with opportunities to access as many document types as possible, and (3) presents data output in as many suitable formats as possible after considering user situation and the expression complexity. This work is an essential contribution because we offer all

types of data presentation formats yet the system requires minimum explicit intervention from the user.

Also we noted that although functional, the system's effectiveness is limited as the modalities for user interaction are already pre-defined. In contrast, a computing system becomes more flexible if no pre-defined input-output modalities are set. In fact, the output presentation of information should be based on the user's application and interaction context (user, system, and environment) which could possibly be in constant evolution. The framework for intelligent multimodal presentation of information (Rousseau et al., 2006) is an example of such flexible system. The system's user interface also should be adaptive to these context variations while preserving its usability. Demeure's work (Demeure et Calvary, 2006) exhibits plasticity in context adaptation. Indeed, the forms of modality should be chosen only based on their merits to a user's interaction context. This is the approach adopted in our work.

In (Awde et al., 2006b), the use of multimodal interaction for non-visual application have been demonstrated. The strength of multimodality lies in the selection of a modality over another after having determined its suitability for data input or output given a specific situation. To visually-impaired users, multimodality is even more important as it provides them greater opportunities to use informatics. In determining the appropriate modality, the user situation plays an important role. In our work, we expanded the notion of user context as one that includes additional handicaps and the user preferences on the priority rankings of media devices and presentation formats.

4.3 Purpose of the Work

The main purpose of this paper is to provide computing infrastructure to visually-impaired users through multimodality. This is in-line with the need for improving the quality of existing applications and for providing visually-impaired users autonomy as they still need assistance of sighted people.

There are two main parts in selecting the suitable presentation format: the interaction context and the expression nature. The interaction context is not our focus here (see (Awde et al., 2007)) so we present it briefly. However, we present our methodology for calculating the expression complexity based on the experiments. Also, we present an architectural framework of an adaptive system that can reason and react accordingly with no or very little human intervention.

We have been working on adaptive multimodal systems ensuring access to mathematics for visually impaired users. The objectives of our works are: a) to improve the quality of applications that present mathematics to visually impaired users, b) to come up with an effective and intelligent way of presenting mathematics so that users can master the application without the assistance of sighted people and c) to provide various suitable presentations format to users.

In this paper, our immediate objectives are a) to provide an efficient method to determine the expression complexity and b) to provide a methodology to integrate the complexity in an adaptive multimodal system.

In order to accomplish these objectives, we prepared the following questions:

- What are the factors (i.e. modalities, media, context...) that applications need to consider in order to produce high quality application which present mathematics to blind users?
- Why are the currently available tools not so effective?
- Which factors and parameters determine the complexity of a mathematical expression?
- How the complexity approach introduced in this study can be integrated in the system to select the most suitable presentation format?

4.4 Modality, Media and Presentation Format

4.4.1 Modality and Media

In our work, we adopt the notions of media and modality that are defined by Bellik in (Bellik, 1995).

1. **Modality** is defined by the information structure as it is perceived by the user (e.g. text, speech, Braille, image, etc.).
2. **Media** is defined as a device used to acquire or deliver information or data (e.g. screen, terminal Braille, mouse, keyboard, etc.).

Here, *Vocal* and *Tactile* modalities are possible since we address visually impaired users. Also, in general, interaction is possible if there exists at least one modality for data input and at least one modality for data output. Given a modality set $M = \{V_{in}, T_{in}, V_{out}, T_{out}\}$ wherein $V_{in} = \text{vocal input}$, $V_{out} = \text{vocal output}$, $T_{in} = \text{tactile input}$ and $T_{out} = \text{tactile output}$ then interaction is possible under the following condition:

$$\text{Interaction Possible} = (V_{in} \vee T_{in}) \wedge (V_{out} \vee T_{out}) \quad (4.1)$$

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

There are usually more than one media that support a specific modality (see Figure 4.1). For example, a regular keyboard, an overlay keyboard and a Braille terminal are all devices supporting tactile input modality. Activating them all is plain redundancy; hence the system must select only the top-ranked device.








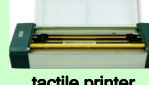
User	Media by Priority				Modalities
	1	2	3	4	
Uses	 microphone				Vocal input (V_{in})
	 speakers	 headset			Vocal output (V_{out})
	 keyboard	 Braille terminal	 overlay		Tactile input (T_{in})
	 Braille display	 tactile printer			Tactile output (T_{out})

Figure 4.1 A sample of media devices priority table and different modality's forms.

4.4.2 Presentation Format

MathML is an application of XML for describing mathematical notations and capturing its structure and content and aimed at integrating them into World Wide Web documents. *MathML* is used to edit mathematical expressions. Also, LaTeX is used to do the same task. These tools present mathematical expression in a bi-dimensional format which is not appropriate for visually impaired users. There are some alternative formats (i.e. Braille, Audio, DotsPlus, EasyMath, etc.) that are addressed to this kind of users. For example, Figure 4.2 shows two specimens expression (in bi-dimensional form) and their equivalents representation in DotsPlus and Braille.

The presentation formats available in the user's computing system are all inputs to the presentation format selection process. Based on interaction context, available media, modalities, expression complexity and user's preferences, the system selects optimal presentation format. For this paper, selection of appropriate presentation format is not discussed (see (Awde et al., 2009) for details).

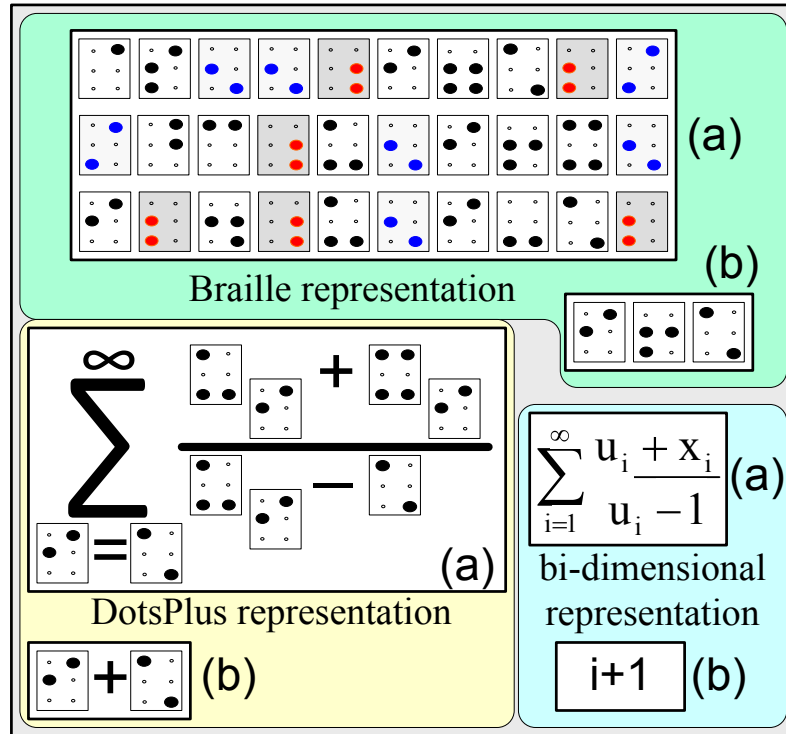


Figure 4.2 Two sample expressions in bi-dimensional form and its corresponding equivalent in Braille and DotsPlus format.

4.5 Adaptive Multimodal Computing Design

In this section, we provide the infrastructure that satisfies the design specifications of our adaptive multimodal system.

4.5.1 Architectural Framework

Figure 4.3 shows the layered view of our adaptive multimodal computing system for visually-impaired users. Adopting a layered architecture approach, one in which data moves from one defined level of processing to another, helps prevent the possibility of ripple effect from propagating to other system components.

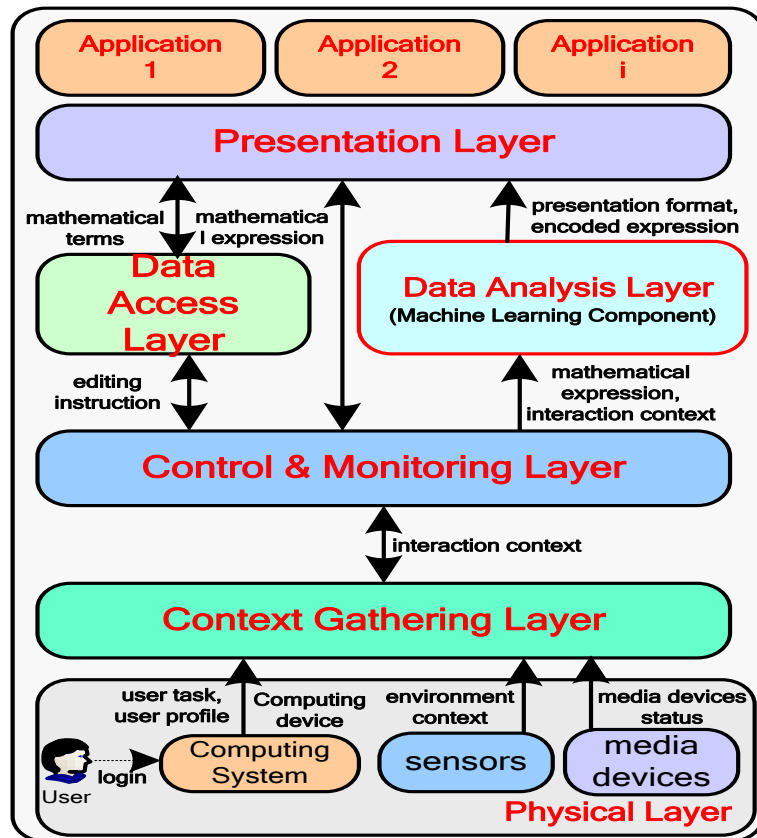


Figure 4.3 Architectural layer view of our multimodal computing system for visually-impaired users.

The various layers and their functionalities are as follows:

Physical Layer : contains all the physical entities of the system, including devices and sensors. The raw data from this layer are sampled and interpreted and forms the current instance of interaction context;

Context Gathering Layer : detects current interaction context;

Control and Monitoring Layer : controls the system, coordinates the detection of interaction context, the mathematical expression, its presentation and/or manipulation;

Data Analysis Layer : here, the presentation format of the mathematical expression is selected based on available resources and user's context;

Data Access Layer : allows search/edit of mathematical expression;

Presentation Layer : presents mathematical expression via optimal presentation format.

An agent programming technique was chosen because the traditional techniques (i.e. functional or object-oriented programming) are inadequate in developing tools that react to environment events. Multi-agent systems (MAS) (Weiss, 1999; Wooldridge, 2002) have been widely used, from relatively small systems such as email filters up to large, open, complex, mission-critical systems such as air traffic control (Jennings et Wooldridge, 1998). Some works on MAS for visually-impaired users are exposed in (Awde et al., 2006a; Awde et al., 2006b). In this work, however, our concern is on the correct representation of a mathematical expression while providing users autonomy. A multi-agent system is used to implement the architectural framework of Figure 4.3. The functionalities of the various agents in our multimodal system are shown in Tableau 4.1.

4.5.2 Contextual Information

Our approach is based on analysis of contextual information and learning techniques to design a system that is capable of adapting to the context (i.e. *interaction context (IC)* and *complexity of expression*). For this paper, interaction context is not discussed in detail since it is not related to this paper's content. IC is formed by combining the contexts of the *user*, his *environment*, and his computing *system*.

The User Context

In this work, the user context (UC) is a function of user profile (including any handicap) and preferences. A sample of user profile, in generic format, is shown in Figure 4.4. The user's special needs determine other affected modalities (i.e. the user is already disqualified from using visual input/output modalities). For example, being mute prevents the user from using vocal input modality.

Tableau 4.1 Functionalities of various agents in our adaptive multimodal multimedia computing system.

Layer	Agent	Functionality
Context Gathering Layer	User Agent	Detects user preferences, such as media and presentation format
	Environment Agent	Detects the user workplace's noise level and its noise restriction
	Device Manager Agent	Detects available and functional media devices
Control and Monitoring Layer	System Management Agent	Detects user's interaction context and mathematical expression, and determines how it will be treated
Data Analysis Layer	Analysis and Conversion Agent	Obtains MathML expression and converts it to its encoded format
	Machine Learning Agent	Determines the optimal presentation format for the given user situation
Data Access Layer	Searching and Editing Agent	Obtains editing instructions from the user
Presentation Layer	Translation Agent	Converts the encoded format of an expression into its final presentation format

Identity	Special Needs	Computing Device	Presentation format preference
username: <username>	Manually Disabled : <Yes/No> Mute : <Yes/No>	computing device1 : <MAC address 1> :: ::	Complexity1 of expression: <presentation format 1> :: <presentation format n>
password: <password>	Deaf : <Yes/No> Unfamiliarity with Braille : <Yes/No>	computing device n : <MAC address n>	Complexity p of expression: <presentation format 2> Etc.

Figure 4.4 A sample of user profile.

The Environment Context

The *environment context* EC is the assessment of a user's workplace condition. To a blind user, a parameter such as light's brightness has no significance, while others, such as noise

level, are significant. In this work, the environment context is based on the following parameters: (1) the workplace's noise level – identifies if it is quiet/acceptable or noisy, and (2) the environment restriction – identifies whether a workplace imposes mandatory silence or not. EC affects the selection of media hence the choice of suitable format. For example: in a library where silence is required, sound-producing media (e.g. speaker) needs to be muted or deactivated.

The System Context

In our work, the *system context* (SC) represents the user's computing device and the available media devices. The computing device (e.g. PC, laptop, PDA, cellular phone) also affects the modality selection. For example, using a PDA or cell phone prevents the user from using tactile output modality so Braille format is not possible.

The Expression Complexity

The complexity of the expression affects the choice of the format of presentation. In case of simple expressions (see Figure 4.2, expression (b)), the user will choose simple presentation format such as Braille or audio. Note that when the expression is complex, user has to choose more complex presentation format such as DotsPlus's presentation (e.g. expression (a) in Figure 4.2). Hence, the complexity of the expression is important for determining the suitable presentation format. A mathematical expression consists of one or more operands and operators. Operators are symbols that represent a function, whereas operands are its arguments. Therefore we developed a method based on the syntax tree of the mathematical expression to determine its complexity.

In our work, we retain the three following parameters: the depth of syntax tree, number of operands and operators. We also examined the impact of mathematical expression branch (e.g. algebra, analysis, etc.) on its complexity. In the next sections, we provide the details of our approach to calculate the expression complexity.

4.6 Mathematics

4.6.1 Mathematical Taxonomy

In the literature, mathematics are defined as the study of the measurement, properties, sets, shapes and motions of physical objects using numbers and symbols. Also we know it as a science of modeling, demonstration and calculation. In our work, we divide mathematics into 6 branches: algebra, arithmetic, calculus, geometry, logic and Statistics & Probability.

4.6.2 Principal Branches of Mathematics:

i. Arithmetic: This branch is also known as numbers theory. It deals with numbers (natural or relative) and their proprieties. Arithmetic does not consider the literal operands that are variables. Our system covers all the arithmetic expressions that contain symbols such as +, -, \times , \div , /, Σ , Π , exponent.

Example: $1+3\times 5=16$

ii. Algebra: It is the branch that substitutes letters for numbers. Algebra deals with symbols and variables. Thus we can consider it as an extension of arithmetic where letters, symbols represent variables. Also, this branch is about the study of quantity, relation and structure (e.g. set, group). In this work, we focus our experience on expressions made of operations and functions such as the four elementary arithmetic operators (+, -, *, /).

Example: $4x^3 - 2x^2 + x - 3 = 0$

iii. Analysis: It is a branch which deals with numbers (real, complex) and their relations. It studies continuously changing quantities and relations which assigns to a variable another corresponding variable. The functions can be presented graphically (curve). Our topics cover the functional expressions such as algebraic and differential equations, variables and functions. On the other hand, it does not manage curve and figures.

Example: $f(x) = \frac{3x+5}{\sqrt{2x}}$

iv. Geometry: It is the science of the figures in the plan and the forms (or volumes) in space. Trigonometry is part of the geometry. It makes it possible to study angles and their properties. In our research, we do not treat the figures as well as the vectors and their properties. However, we treat trigonometric expressions including functions such as sin, cos, tan.

Example: $\cos^2(x) + \sin^2(x) = 1$

v. Logic: it is the study for composing proofs, which gives us reliable confirmation of the truth of the proven proposition. It is also called "science of the reasoning". It is used to approve a reasoning, to distinguish the truth of the forgery. This branch is widely used in computer science field. The logic expressions we are considering here are propositional logic and first order logic.

Example: $a \wedge b$

vi. Statistics & Probability: Probability is the likelihood or chance that something is the case or will happen. Probability is essential to many human activities that involve quantitative analysis of large sets of data. Statistics is a mathematical science about the collection, description, analysis, interpretation of data and prediction. In this branch, we consider expressions written to describe a phenomenon but we do not treat graphs or histograms.

Example: $P(x) = \frac{1}{2}$

4.6.3 Mathematical Expression

A mathematical expression consists of several operands connected by operators. In a mathematical expression, brackets and parentheses group terms aim to simplify reading of expression and specify priority of calculations. Each set of terms ranging between two brackets (open and close) is called block.

Operands

An operand is defined as the element on which operations are carried. It can be constant or variable. In a mathematical expression, there is always at least one operand. The nature of the operand varies from a mathematical branch to another. For example, in arithmetic the operands are numbers. However, in algebra they are numbers and literal variables, and in logic, the operands take truth values to make predicates. Operand can be simple (single operand) or composed (a group of operand and operator enclosed in parentheses).

Operators

An operator performs a logical or mathematical operation on the operands in an expression. It settles the relation between terms of the mathematical expression. Each mathematical branch has some specific operators although there are common symbol operators between these branches. The complexity of an operator depends on the number of its operands. Indeed, there are several types of operators:

Unary operator : it needs just only one operand such as the square root, sin, logical predicates (NOT).

Binary operato : it involves exactly two operands such as the addition (+), the multiplication (*), logical predicates (e.g. OR, AND), etc.

N-ary operators : when the operator has three operands or more, it is known as n-ary operator. For example, definite integral has 3 arguments (i.e. $f(x).dx$, *upper and lower* limits).

In an expression that contains multiple operators, there are rules to decide the order in which operators are evaluated. These rules are called operator priority. Operators of higher priority within an expression are executed before operators of a lower priority. For example multiplication has a higher priority than addition. To increase the priority of an operator, we place it and its operands inside parentheses. Also, these parentheses or brackets contribute to read and understand the expression easily in dividing the expression to subsets.

The Syntax Tree

To determine the complexity of a mathematical expression, it is necessary to consider all expression components. Operators and operands are easily detected while reading the expression. However, presence of many brackets and especially overlapping brackets increase the challenge and complicates analysis of the expression. Therefore, we elaborated a solution based on the principle of syntax tree of the expression where internal nodes are labeled by operators (nonterminal), and the leaf nodes represent the operands of the operators (terminal). Thus, the leaf nodes only represent variables or constants (Cormen et al., 2001). Finally we consider *the depth* of tree being the number of connections that we encounter starting from root node (i.e. topmost node) to the farthest leaf. This parameter points to the number of groups (i.e. parentheses or brackets) in the expression. In this work, the tree of an expression has a unique root and at least one leaf.

To build a syntax tree of the expression, we proceed in a similar way as the conversion of the expression into postfix form. We build subtrees for the subexpressions (i.e group of terms between 2 brackets) by creating a node for each operator and operands. The children of an operator node are the parents of the nodes representing the subexpressions constituting the operands of that operator. The subexpressions are determined based on the nature of operators and their priorities. When operators have identical priority, the construction of subexpressions is done by grouping operators and operands from left to right (i.e. left-associative or the order of evaluation). Consider, for example, this expression:

$$\frac{(1 + x)(y + 5 \times x)}{x + 2 + y}$$

In this expression, parentheses in numerator identify well the two operands of the multiplication (2 subexpressions). However for the term $(y+5\times x)$, the multiplication has a higher priority than the addition. In this case, multiplication operator and its operands (i.e. 5 and x) become a subexpression and addition operator with its operands (i.e. y and $5\times x$) are

another one. On the other hand, in the denominator, there are 2 identical operators. Therefore we divide denominator in 2 subexpressions according to the order of left associative. So the expression can also be written as follows:

$$\frac{(1 + x)(y + (5 \times x))}{((x + 2) + y)}$$

Figure 4.5 shows the syntax tree of that expression. Note that left side of the figure presents the numerator and right side illustrates the denominator.

The root node represents the fraction bar. Internal nodes represent operators in the expression (*i.e.* $\times, +, +, \times, /, +, +$). While leaf nodes correspond to variables and constants (*i.e.* $1, x, y, 5, x, x, 2, y$). For the depth, we count 4 connections from the root (*i.e.* $/$) to the farthest leaf (*i.e.* 5 or x).

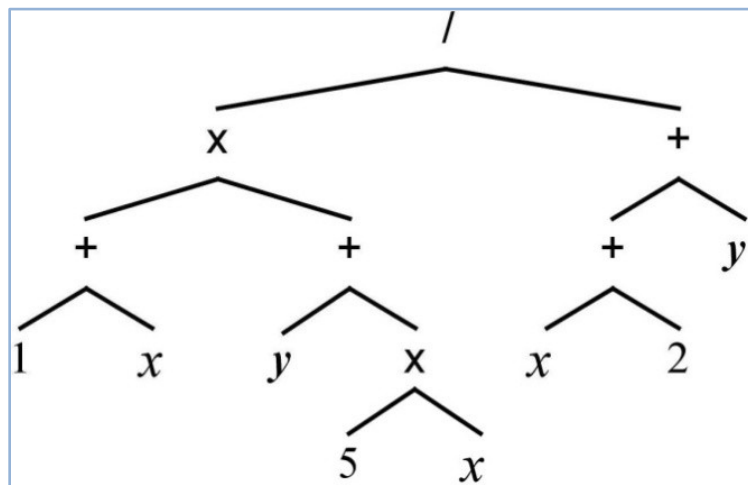


Figure 4.5 A syntax tree of the sample expression.

4.7 The Experiment

4.7.1 Hypothesis

In our experiment, we have some assumptions:

- There is a relation between the three parameters of the expression (operators, operands, depth) and its complexity. Then our tentative is to determine the function which allows calculating the complexity.
- The parameters of the expression have not the same influence on the complexity so we have to determine the weight of each one of them.
- The branch of the expression may affect its complexity. In probability branch, special symbols such as (Σ , Π , etc.) are used often and they are more complex than arithmetic symbol such as $+$, $-$, \times , etc.

4.7.2 Evaluation Protocol

Our objective of the evaluation is to establish a relation between the three parameters (operators, operands, depth) and the complexity of the expression. According to the mathematical taxonomy that we presented above, mathematics comprise 6 branches. We composed ten expressions of each branch, with various complexities to cover all possible degrees of complexity (varying the number of operators, operands and parentheses from one expression to another).

We want to check if the complexity of the mathematical expression varies from a branch to another when both have the same number of parameters as well as the relation between the type of mathematical operators and expression complexity. Also, we made another test where the collection contains expressions of various mathematics branches.

In our experience, the main hint to the complexity is the time allocated by participant for reading and memorizing the expression. Also, error rate during the rewriting of the expression gives another indication to the level of complexity.

To measure the error rate, we took a metric that is similar to a common metric of the performance of a speech recognition system and machine translation system. That metric is known as word error rate (WER) (McCowan et al., 2005) . It is derived from Levenstein distance. The distance between two sentences is the minimum number of insertions, deletions

and substitutions required to transform one sentence into the other. Here, we assign the same weight to these three operations. Also, we use expressions instead of sentences and terms (operator and operand) instead of words. The WER is the distance between an expression reference and its corresponding expression written by the participant. Formally, the WER is defined as:

$$WER = \frac{I + D + S}{N}$$

Where I is the number of the insertions, D is the number of the deletions, S is the number of substitutions and N is the number of operators and operands in the expression reference.

4.7.3 Experiment Running

The participant must take his time to read and memorize the expression shown on-screen. When participant think that he knows well the expression, he hide it and try to write the expression on a sheet. At the end, the participant assigns a digit (from 1 to 5) to the expression to indicate its complexity according to his experience and values of Tableau 4.2. In this experiment, participant considers the expression that takes a long time to be memorized as very complex.

Tableau 4.2 The complexities and its suiting values.

Complexity	Value
Very Simple	1
Simple	2
Average	3
Complex	4
Very Complex	5

The purpose of the experiment is to verify if the participants have the same tendency to classify expressions or the order varies from one person to another. In this way, we can

determine the relationship between the profile of the reader (i.e. participant) and the complexity of the mathematical expression. The expressions are presented to the participants in a random order of complexity.

The expressions are presented on a screen of a computer in order to measure the time for reading, memorizing and rewriting the expression. However, the rewriting of expressions is made by hand. The participants in this experiment are university graduates in different studies, such as computer science, mathematics, chemistry, literature, etc. We choose this degree of studies because the types of the operators used in this experiment are not recognized by college students. The participants are skilful to use computer therefore we did not encounter problems of this kind during tests. There were 50 participants. A test of ten expressions takes in average 15 minutes. After each test passed by the participant, we collected information by noting times taken for each expression, and by checking the syntax of the expression written by this participant. The error rate for each expression is then calculated.

4.8 Results of Experiments and Discussions

In this section, we present for each branch of mathematics the data collected during our experiments and the corresponding results. Results are shown in the following tables and graphs. The first column in these tables presents the mathematical expression number, the second column shows the number of operands, the third is the number of operators and the fourth is the depth of the syntax tree of the expression. These data are easily extracted from the syntax tree of the expression. The last four columns of the tables are results of our experiments. We calculated the average time taken by all participants for each expression. The rate column is the average of the error rates of all the participants. Complexity row shows in which complexity level the participant has placed the expression (there are 5 levels).

On graphs, R. Time and W. Time mean average time taken by all participants to read and write an expression respectively.

4.8.1 Arithmetic

Tableau 4.3 presents the results for arithmetic expressions. The number of operands varies between 1 and 8. The number of operators varies between 2 and 9. Here, the depth of the syntax tree is not more than 4 as the depth is always less than the number of operators.

Let us take expressions 2 and 4 (see appendix). They have the same number of operands "5" and operators "4" but their tree depths are different ("3" and "4"). The average time for reading and memorizing expression 2 was 13.06 seconds. On the other hand, expression 4 took almost 24.87 seconds. In the same way, the error rate passed from 1.48% to 8.15%, and the majority of the participants gave complexity 4 and 3 to expression 4 and 2 respectively. We can say that it is due to the fact that expression 4 includes several brackets. Thus, the depth has an influence on the complexity of the expression and should be taken into account in the other branches of mathematics.

Tableau 4.3 Experiment's results of arithmetic branch.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	3	2	2	5.01	6.67	0.00	1.64
2	5	4	3	13.06	11.47	1.48	2.67
3	4	3	3	7.27	4.40	0.00	1.81
4	5	4	4	24.87	16.73	8.15	4.33
5	4	4	4	10.27	8.20	1.90	2.53
6	8	7	3	23.47	10.67	11.11	3.73
7	4	3	3	9.13	5.87	0.00	1.87
8	6	9	4	20.00	13.80	1.33	4.04
9	7	7	4	28.60	12.20	19.05	4.67
10	1	2	2	2.20	3.73	0.00	1.01

Figure 4.6 illustrates reading time that reaches its high with expressions 4, 6, 8 and 9. These expressions have the most operators, operands and depth (i.e. 4).

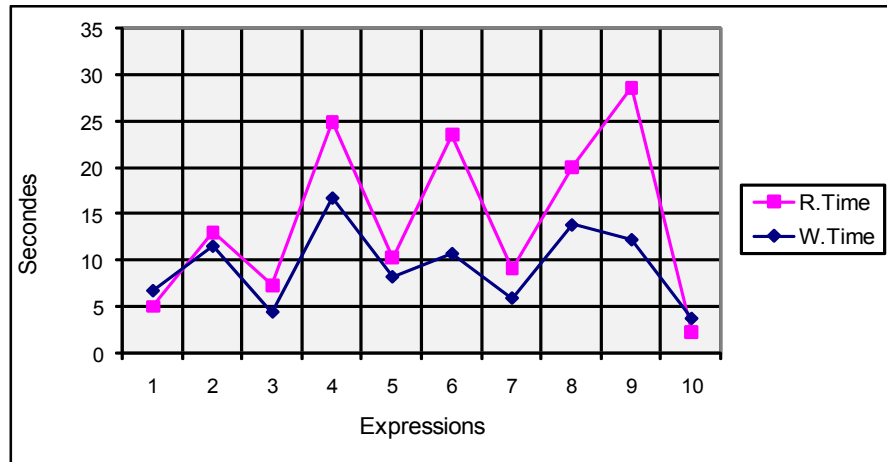


Figure 4.6 Curves representing reading and writing times taken by participants during experiment of arithmetic branch.

4.8.2 Algebra

Tableau 4.4 and Figure 4.7 represent test results for algebra branch. We notice that the tendencies of participants continue in the same direction to confirm our assumption that the complexity of mathematical expressions depends on these three parameters (i.e. operators, operands, depth). For example, expression 7 has higher parameters than other expressions. Experiments classify this expression as the most complex expression among all expressions in this branch. In Figure 4.7, we notice that expression 7 takes the highest reading time (26.82 seconds).

Tableau 4.4 Experiment's results of algebra branch.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	4	4	4	4.80	7.13	0.00	1.99
2	6	7	4	17.62	12.07	1.54	3.69
3	7	6	4	8.73	7.62	9.23	3.62
4	5	6	4	12.93	10.02	6.06	3.40
5	3	2	2	2.20	2.18	0.00	1.52
6	9	8	4	12.61	11.13	5.49	3.87
7	8	11	5	26.82	16.20	15.19	4.93
8	4	3	2	4.92	4.62	0.00	1.29
9	2	1	1	2.13	4.73	0.00	1.07
10	5	5	3	6.64	7.87	12.67	2.87

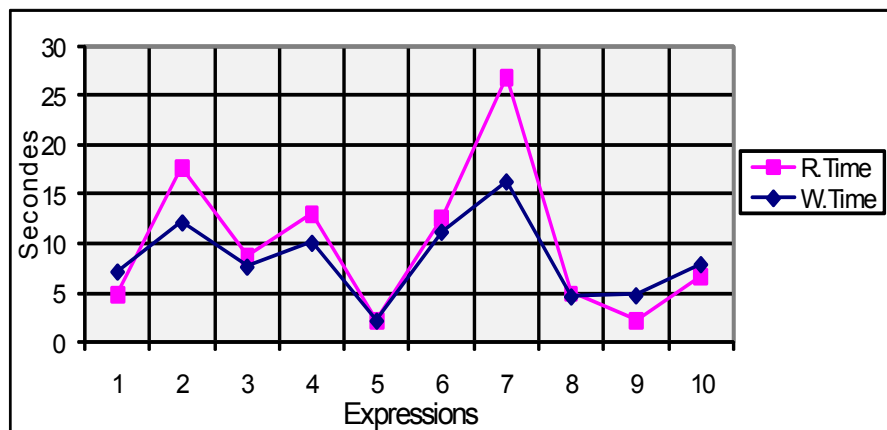


Figure 4.7 Curve representing reading and writing times taken by participants during experiment of algebra branch.

4.8.3 Analysis

Let us take the two expressions, 6 and 9, which are considered complex by participants. As shown in Tableau 4.5, parameters of expression 6 are higher than the ones of expression 9.

Then, expression 6 must take more time and should be considered the most complex among all expressions in this experiment. However, the experimental results illustrated in Figure 4.8 shows the contrary (20.53 seconds for expression 6 and 22.94 for expression 9). In fact, there are some participants not familiar with the mathematical function "log", and then reading time increased for this expression.

Tableau 4.5 Experiment's results of analysis branch.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	5	5	3	8.45	7.53	0.00	2.56
2	2	2	2	2.80	3.31	0.00	1.09
3	3	1	1	2.82	5.04	0.00	1.01
4	4	3	3	4.87	4.35	0.00	1.72
5	6	7	4	14.61	11.42	18.97	3.97
6	7	10	5	20.53	16.41	5.88	4.53
7	4	4	3	14.07	5.82	4.17	2.80
8	8	7	3	12.73	11.67	4.44	3.93
9	5	8	5	22.94	13.13	9.23	4.67
10	7	6	4	17.02	11.92	7.18	3.59

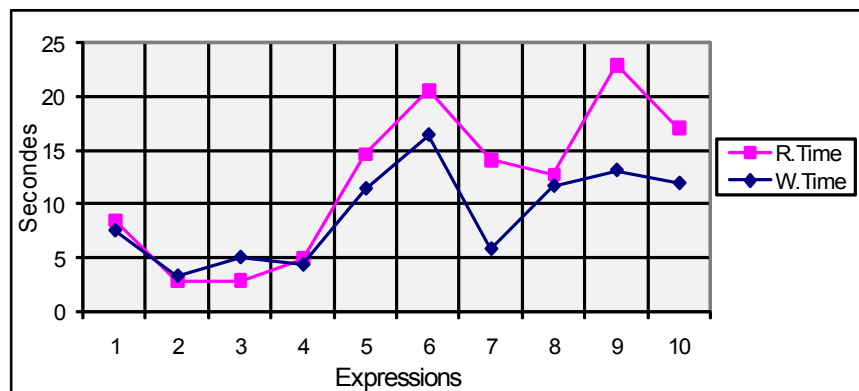


Figure 4.8 Curves representing reading and writing times taken by participants during experiment of analysis branch.

4.8.4 Geometry / Trigonometry

In this branch, expressions cover all trigonometry functions. Test results are presented in Tableau 4.6 and Figure 4.9.

Expressions 2 and 9 have the same number of operands, but expression 9 has one more operator. However, the complexity of the expression 9 is higher than the expression 2. We asked participants about that difference. They think that it is because of the presence of numbers in the expression. They also found that expression 2 is similar to arithmetic expression. In addition, for scientific participants, these 2 expressions had the same complexity, but for other participants the expression 9 was more complicated.

Tableau 4.6 Experiment's results of trigonometry branch.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	3	4	3	6.47	6.21	0.00	1.51
2	5	5	4	15.36	7.87	0.00	2.53
3	6	7	4	19.53	9.32	1.54	3.64
4	4	6	4	17.51	8.24	1.34	3.13
5	2	5	3	8.34	9.72	4.76	2.04
6	1	3	3	6.72	4.28	0.00	1.13
7	7	12	5	37.73	16.42	1.41	4.61
8	8	9	5	24.92	12.26	9.42	4.58
9	5	6	5	22.27	9.04	6.06	4.24
10	7	8	5	30.40	12.82	12.02	4.47

Graph in Figure 4.9 shows that expressions 7, 8, 9 and 10 are the most complex among all expressions, and this is due to the largest number of parameters of these expressions.

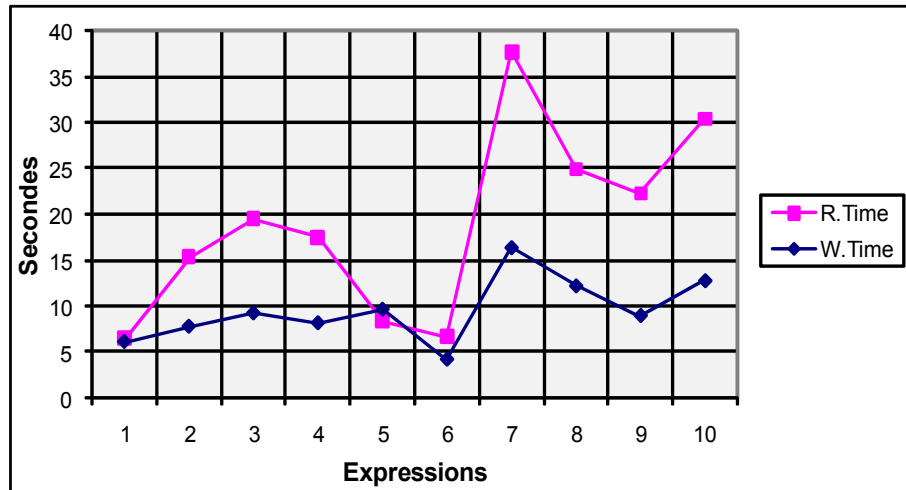


Figure 4.9 Curves representing reading and writing times taken by participants during experiment of trigonometry branch.

4.8.5 Logic

During our tests, we prepared a summary of this branch to explain it to participants who had not enough knowledge in this field.

Even with this assistance, we can notice, in Tableau 4.7 and Figure 4.10, that participants took longer time to read logic expressions comparing to other branches like arithmetic even with the same number of operands and operators.

In the same way, expressions with a higher number of operators and operands need a longer reading time. For example, expression 8 took more than one minute for reading. This is due to the high number of terms.

Tableau 4.7 Experiment's results of logic branch.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	4	2	2	9.40	6.46	0.00	1.92
2	8	6	4	21.73	17.81	1.91	2.87
3	2	3	3	10.52	5.88	0.00	1.59
4	7	6	3	26.61	20.32	0.00	4.15
5	6	7	3	22.24	18.53	1.03	3.94
6	3	4	3	19.07	11.34	13.31	2.48
7	5	5	3	22.62	16.41	3.34	3.46
8	9	10	5	61.53	28.12	27.72	4.92
9	5	8	6	37.12	32.47	22.56	4.71
10	6	7	4	26.34	15.21	6.15	3.52

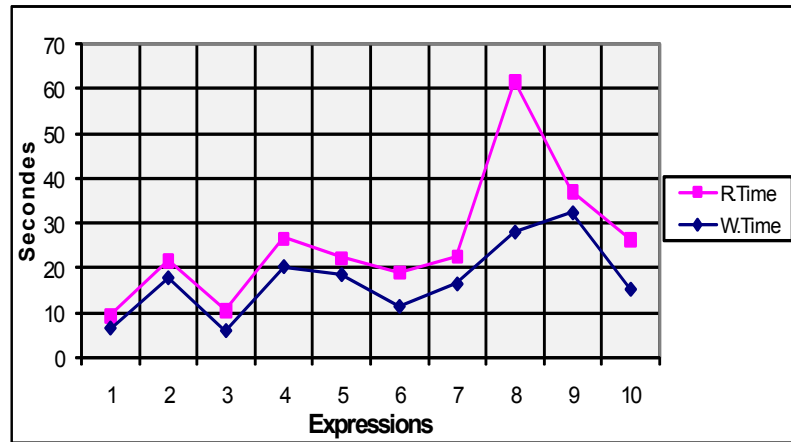


Figure 4.10 Curves representing reading and writing times taken by participants during experiment of logic branch.

4.8.6 Statistics & Probability

In this mathematical branch, we use often operators which are "tri-nary" such as the summation. This will increase the quantity of information to be memorized and therefore the reading time. This hypothesis was confirmed during evaluations, as shown in Tableau 4.8 and Figure 4.11.

For example, participants took on average of 56.72 seconds to memorize expression 10. This time is justified by the appearance, in the expression, of a rarely used function (i.e. VAR) and the large number of parameters compared to other expressions.

Tableau 4.8 Experiment's results of statistics & probability branch.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	7	9	5	40.42	18.52	18.33	4.28
2	5	4	3	22.23	15.47	11.11	2.64
3	3	1	1	5.87	5.41	0.00	1.04
4	9	7	5	33.87	23.02	7.08	3.85
5	2	1	1	2.03	3.52	0.00	1.02
6	4	6	4	18.81	11.46	4.04	2.53
7	6	5	5	35.04	19.81	5.15	3.87
8	6	3	3	24.93	12.72	5.19	3.33
9	4	2	2	9.41	7.03	0.00	1.53
10	8	12	6	56.72	21.53	14.68	4.83

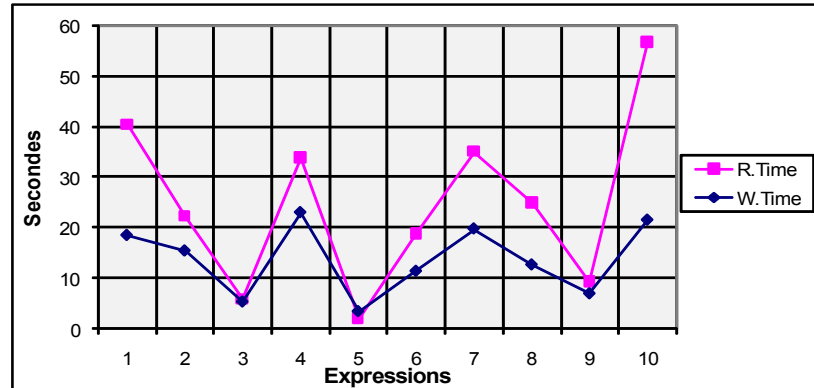


Figure 4.11 Curves representing reading and writing times taken by participants during experiment of probability & statistics branch.

4.8.7 Combination

In this section, the set of expressions is a collection of 12 expressions of all mathematical branches. Here, our objective is to verify experiment results when expressions are a combination of various mathematical branches. To do so, the same participants of previous experiments realized this test. Tableau 4.9 and Figure 4.12 show the test results.

The results of this test confirmed that the tendencies of participants remained the same as the previous experiments. We noticed that expressions with higher parameters were classified among the most complex expressions such as expressions 5, 11 and 12.

According to the parameters, these expressions were really the most complicated in this test. In fact, the depth (10) of expression 5 is more significant than the one of expression 12 (6). Also, expression 5 has a very high number of overlapping blocks that affects significantly the complexity of expression 5.

Tableau 4.9 Experiment's results of a collection of 12 expressions.

No. Expression	Operands	Operators	Depth	Memorizing			
				Average reading time (sec)	Average writing time (sec)	Error rate	Subjective complexity
1	7	6	4	20.56	7.19	2.41	2.75
2	4	3	3	19.88	7.75	3.57	1.66
3	1	4	4	15.25	11.02	0.00	1.37
4	7	9	4	31.75	21.51	5.08	3.63
5	6	11	10	66.38	36.31	16.18	4.57
6	3	1	1	8.75	5.38	0.00	1.05
7	4	2	2	18.44	12.63	4.17	1.58
8	5	7	5	27.04	14.75	5.21	3.24
9	9	5	4	47.69	23.63	9.38	4.71
10	2	10	6	45.88	30.52	6.25	4.19
11	8	5	3	38.15	22.77	6.25	4.06
12	10	16	6	46.13	32.31	13.22	4.89

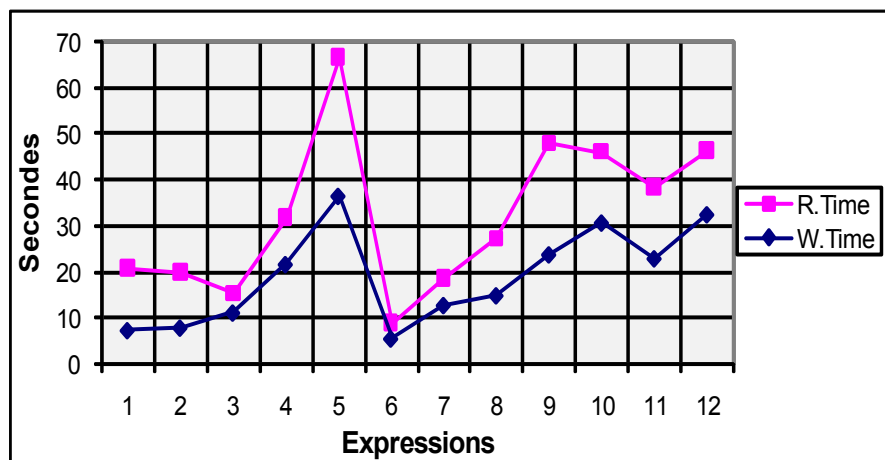


Figure 4.12 Curves representing reading and writing times taken by participants during experiment of the 12 expressions.

4.9 Results and Application

4.9.1 Results Analysis

According to the results obtained above, our assumption is confirmed that expression complexity depends on the three parameters as well as the mathematical knowledge of participants. We know that the three parameters have a significant impact on the complexity of the expression, but we do not know the weight of each one. Then we try to find these weights by examining graphs and the collected data.

We noted that the three variants follow the same curve. In fact, when the complexity of an expression increased, the reading and writing time increased. However error rate is null when expression is categorized very simple or simple and becomes more significant when expression is more complex. So in this experiment we work on the variant named “subjective complexity”.

The principle of this evaluation is to observe complexity of the similar expressions (same number of operands, operators and depth). Tableau 4.10 presents 4 similar expressions that have the same parameters (operands=6, operators=7, depth=4). In this work, we say that these 4 expressions are of model $M(6,7,4)$.

Tableau 4.10 4 similar expressions and its average complexity.

Branch	Operands	Operators	Depth	complexity
Algebra	6	7	4	3.69
Analysis	6	7	4	3.97
Probability	6	7	4	3.52
Combination	6	7	4	3.64
Average				3.71

Expressions shown on Tableau 4.10 come from 4 different mathematical branches (i.e. algebra, analysis, statistics & probability and combination). The data of the last column show the average of all complexities estimated by the participants during the evaluations and the data in the last cell is the average of all averages of these complexities.

Since our objective is to find this complexity in a systematic way, it is necessary to find the relation between these parameters (operands, operators, depth).

We note here that the subjective complexity follows a linear curve then we opt to propose a linear function. Specifically, the idea is to associate a weight for each parameter. The formula that we proposed is:

$$C = x \times P + y \times F + z \times D \quad (4.2)$$

where C is the complexity of the expression, P the number of operands, F the number of operators, and D is depth of the tree. x, y and z are the weights of the operands, operators and depth respectively.

Let us consider a mathematical expression which is of model M(6,7,4) then according to Tableau 4.10, its average complexity is 3.71.

The equation becomes: $3.71=6x+7y+4z$

Then, we take a set of similar expressions in order to have more data for determining the values of these unknown variables.

To find the values for these variables, we follow the same method used to collect the data of Tableau 4.10. The experiments count 72 different expressions which belong to 51 different models. We keep the model that has at least two samples of expressions in our experiments. We found only 13 models which presented in Tableau 4.11. Considering these data, we build a problem of 13 equations (see Figure 4.13) that illustrates a linear programming problem. It consists of the following three parts:

- 1) The linear function to be optimized is: $C = x \times P + y \times F + z \times D$;
- 2) The problem constraints presented in Figure 4.13;
- 3) the non-negative variables $x \geq 0, y \geq 0, z \geq 0$.

$$\left\{ \begin{array}{l} 2x + y + z = 1.05 \\ 3x + y + z = 1.03 \\ 3x + 2y + 2z = 1.58 \\ 3x + 4y + 3z = 1.99 \\ 4x + 2y + 2z = 1.68 \\ 4x + 3y + 3z = 1.77 \\ 4x + 4y + 4z = 2.26 \\ 4x + 6y + 4z = 2.83 \\ 5x + 4y + 3z = 2.66 \\ 5x + 5y + 3z = 2.96 \\ 6x + 7y + 4z = 3.71 \\ 7x + 6y + 4z = 3.32 \\ 8x + 7y + 3z = 3.83 \end{array} \right.$$

Figure 4.13 Our linear programming problem.

Solving this problem, we get: $X=0.39; y=0.10; Z=0.16$. Then, the formula becomes:

$$C = 0.39 \times P + 0.10 \times F + 0.16 \times D \quad (4.3)$$

In Tableau 4.11, we use this formula to validate our approach. Column 5 (average Complexity) presents the results of experiments for several models of expressions. However, column 6 (estimated complexity) shows results of calculation using our formula. The last column shows if two complexities (average and estimated) are equivalent.

Tableau 4.11 Comparison table of estimated complexity using our formula and average complexity according to experiments.

Model	Operands	Operators	Depth	Complexity		Valid hypothesis
				Average	Estimated	
M(2,1,1)	2	1	1	1.05	1.04	YES
M(3,1,1)	3	1	1	1.03	1.43	YES
M(3,2,2)	3	2	2	1.58	1.69	YES
M(3,4,3)	3	4	3	1.99	2.05	YES
M(4,2,2)	4	2	2	1.68	2.08	YES
M(4,3,3)	4	3	3	1.77	2.34	YES
M(4,4,4)	4	4	4	2.26	2.60	NOT
M(4,6,4)	4	6	4	2.83	2.80	YES
M(5,4,3)	5	4	3	2.66	2.83	YES
M(5,5,3)	5	5	3	2.96	2.93	YES
M(6,7,4)	6	7	4	3.71	3.68	YES
M(7,6,4)	7	6	4	3.32	3.97	NOT
M(8,7,3)	8	7	3	3.83	4.30	YES

Tableau 4.12 shows the categories of complexity (very simple, simple, average, complex, very complex) as we mentioned in the protocol of evaluation and its corresponding numeric value.

The exhaustive success classification rate (for the 72 available mathematical expressions) is 81.94%. The success classification rate (for the 13 models) is about 84.62%. In fact, expressions of model M(4,4,4) and M(7,6,4) in Tableau 4.11 are considered as simple and average respectively according to the experiment, but average and complex according to our formula. Then the results of our approach, even if they are not always equivalent to the results of the evaluations, are coherent and acceptable.

Tableau 4.12 The five categories of complexity.

Complexity	Category
$C < 1.5$	Very Simple
$1.5 \leq C < 2.5$	Simple
$2.5 \leq C < 3.5$	Average
$3.5 \leq C < 4.5$	Complex
$4.5 \leq C$	Very complex

4.9.2 Integration of Expression Complexity in Our System

The analysis of a mathematical expression is explained as follows. Given a mathematical expression in MathML format for example, it is analyzed lexically using *grammar rules and dictionary*. The result yields a list of lexemes. A lexeme is a parameter within an expression which may be an operand or an operator. Given the lexemes, the *parser* analyzes the expression parameters (i.e. operands and operators and syntax tree) then sends parameters to the *Expression Evaluator* to determine its complexity based on equation (3). The parser sends then the expression to the *Expression Encoder* to be translated into its encoded format. This process is shown in Figure 4.14.

As an example, Figure 4.15 shows a sample of fraction (4.4).

$$\frac{x + 1}{x - 1} \quad (4.4)$$

As shown, in step 1, the MathML expression is sent to the Lexer. In step 2, using the XML grammar, the expression is decomposed into a list of lexemes; the list is then sent to the parser. In step 3, the operations and operands in the expression are sent to expression evaluator and encoder. Together, in step 4, the evaluator deduces the complexity of the expression (e.g. simple) while the encoder produces the encoded expression. Finally, in step

5, the complexity of the expression is determined and the encoded expression becomes an input to the presentation format selection process.

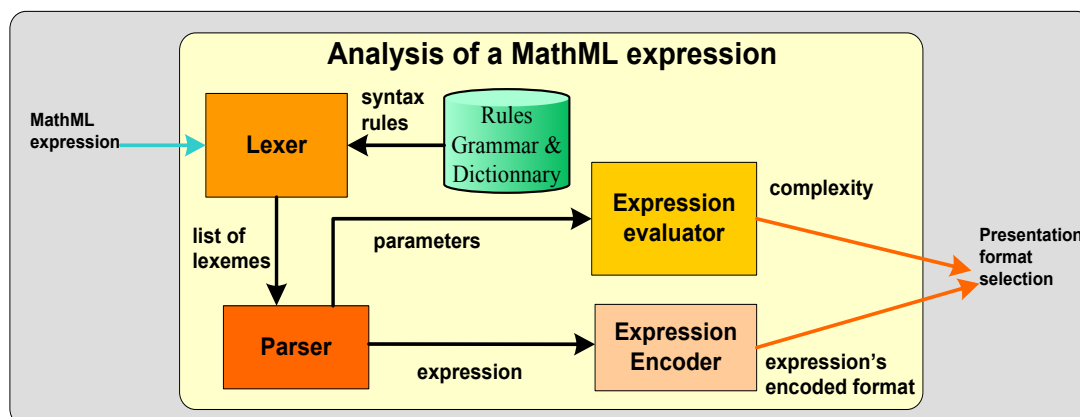


Figure 4.14 Structure of the analysis of a MathML expression.

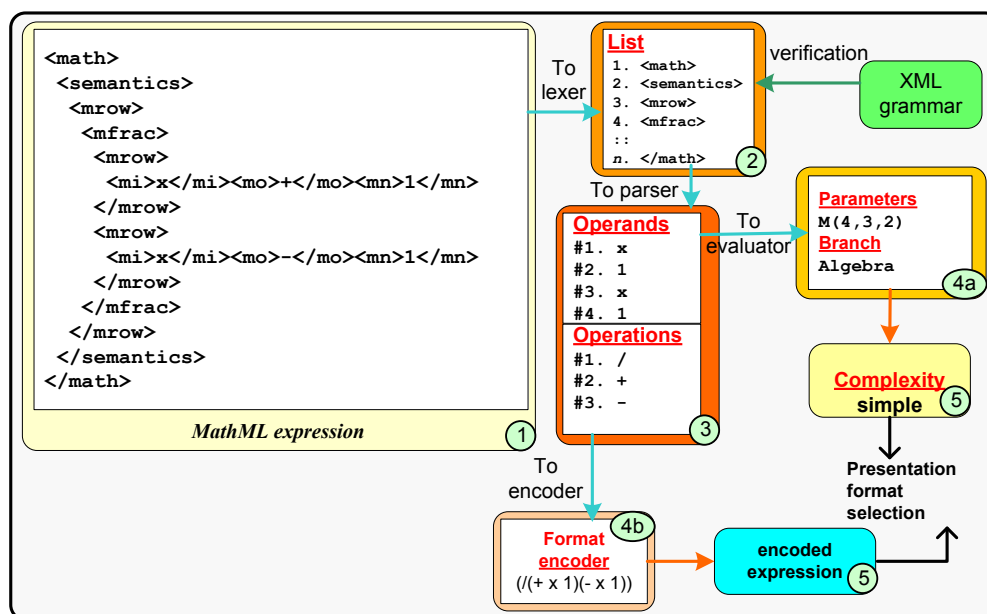


Figure 4.15 A sample analysis of a specimen fraction.

4.10 Conclusion

Our ongoing research focuses on providing computing infrastructure to visually-impaired users through multimodality. One area of such domain is the infrastructure for mathematical presentation to blind users which this paper addresses. In this paper, we have noted the weaknesses of the current state-of-the art solutions. We note that the current available solutions present mathematical expressions in one presentation format. Also there is no consideration to the contexts of the user, his working environment and his computing system as well as the nature of the mathematical expression itself and of the user's preferences. Moreover, these solutions do not provide visually-impaired users with autonomy as the users still need the assistance of sighted people.

In this work, we have presented the architectural framework of our adaptive system. We provided taxonomy of mathematics and our methodology for determining expression complexity. Also, we have presented the experiment results and the way to integrate complexity in our system.

In our future works, the solution will consider various presentation formats to present the expression in suitable presentation format based on user's situation and expression complexity. Also, future works involve the prototyping of this infrastructure and simulating its performance using several computing systems. Such prototype will also be tested on blind users with various interaction contexts.

4.11 Appendix

a) Arithmetic

$$3 \times (2 + 4) \tag{1}$$

$$((7 \times 6) \times 8) \times (3 + 2) \tag{2}$$

$$5 \times (9^2 - 3) \tag{3}$$

$$3 \times (((5 \times) - 2) \div 4) \tag{4}$$

$$(\sqrt{4^3} + 2)^4 \tag{5}$$

$$\frac{2^2}{3^2} \div \frac{4}{5} \times 2 \tag{6}$$

$$\frac{1}{2 + \frac{1}{3}} \quad (7)$$

$$\frac{\sqrt{2} \times \sqrt{5}}{\sqrt{3} \times \sqrt{6}} \times \frac{2}{3} \quad (8)$$

$$\sqrt{\frac{8^3}{11}} - (2^2 - 3^5) \quad (9)$$

$$\sqrt{\sqrt{6}} \quad (10)$$

b) Algebra

$$\sqrt{ab^2c} \quad (1)$$

$$\frac{\sqrt{a \times b + c \times d}}{\sqrt{ab}} \quad (2)$$

$$x^2 + 5mx - m^2 \quad (3)$$

$$(x\sqrt{2} + 1)\sqrt{x+1} \quad (4)$$

$$x^2 + x \quad (5)$$

$$\frac{1+x}{1-x^2} + \frac{1-x}{1+x} \quad (6)$$

$$\frac{\sqrt{x} \times \sqrt{y}}{2\sqrt{x+y}} \times \frac{\sqrt{x^2}}{y} \quad (7)$$

$$\frac{\frac{1}{x}}{x+1} \quad (8)$$

$$\sqrt[3]{x} \quad (9)$$

$$\frac{\sqrt{x}}{x+1} \times \frac{1}{x} \quad (10)$$

c) Analysis

$$f(x) = \frac{x+1}{x-1} \quad (1)$$

$$\frac{2}{\log x} \quad (2)$$

$$\int_0^4 x dx \quad (3)$$

$$y = 3x + 1 \quad (4)$$

$$y = \frac{\sin x}{\sqrt{x}} + \frac{1}{x+2} \quad (5)$$

$$y = \frac{\sqrt{\sin x}}{2+x} + \frac{1+x}{\sqrt{\cos x}} \quad (6)$$

$$y = \frac{x}{2} \times \sqrt{\frac{3}{x}} \quad (7)$$

$$\frac{\log_{10} x + \log_{10} y}{x^2 - y^2} \quad (8)$$

$$\frac{\sqrt{\log(x+1)}}{\log \sqrt{x^2 - 1}} \quad (9)$$

$$y = (3x + 2)(2x - 1) \quad (10)$$

d) Geometry / Trigonometry

$$\sin x = \frac{1}{\cos y} \quad (1)$$

$$\cos x = \frac{3\sqrt{7}}{7} \quad (2)$$

$$(1 + \cos 2x)(1 - \sin 3x) \quad (3)$$

$$(\sin 2x)^3 + \sqrt{\tan x} \quad (4)$$

$$|\tan x| = \sqrt{\sin x} \quad (5)$$

$$\sqrt{|\sin x|} \quad (6)$$

$$\cos(x + y) - \cos x \cos y + \sin x \cos y = 0 \quad (7)$$

$$(\cos 3x)^2 + 2a(\sin 2x)^2 \quad (8)$$

$$\cos y = a \cos(bx + c) \quad (9)$$

$$4 \cos(x - \pi) = \cos(3y + \pi) \quad (10)$$

e) Logic

$$\bigwedge_{i=m}^n P_i \quad (1)$$

$$\bigwedge_{i=0}^n P(a_i) \vee \bigwedge_{j=1}^m P(b_j) \quad (2)$$

$$\neg(\neg p \wedge q) \quad (3)$$

$$(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)) \quad (4)$$

$$(\neg B \Rightarrow \neg A) \Rightarrow ((B \Rightarrow A) \Rightarrow (A \Rightarrow B)) \quad (5)$$

$$\neg q \wedge (\neg p \vee q) \quad (6)$$

$$\neg p \vee p = \bigwedge_{x=0}^{\infty} P(x) \quad (7)$$

$$(p \wedge r \vee q) \wedge \neg(p \vee (r \wedge s)) \vee (p \wedge \neg \wedge s) \quad (8)$$

$$\neg(\neg p \wedge q \vee \neg(r \wedge \neg s) \vee p) \quad (9)$$

$$\neg(p_1 \vee t) = \neg p_2 \vee r \quad (10)$$

f) Statistics & Probability

$$\prod_{i=0}^m \sqrt{i} + \frac{k!}{n!(k-n)!} \quad (1)$$

$$\frac{1}{n} \times \sum_{k=1}^n P(k) \quad (2)$$

$$\sum_{i=1}^n i \quad (3)$$

$$\sum_{i=0}^k \left(\frac{n_i^2}{np_i} - n_i \right) \quad (4)$$

$$C_n^k \quad (5)$$

$$P(A \cup B) - (P(c) + P(d)) \quad (6)$$

$$\sum_{k=1}^n \sum_{j=0}^m P(k)P(j) \quad (7)$$

$$\sum_{i=0}^n \prod_{j=1}^m i \times j \quad (8)$$

$$\sum_{i=0}^n x_i \quad (9)$$

$$|\sqrt{n+1}| = \frac{1}{\frac{\text{var}(x) - 2}{2}} \frac{1}{n! \times k} \quad (10)$$

g) Combination

$$2x^3 - 3x^2 + 1 \quad (1)$$

$$x + \log_{10}|x^3| \quad (2)$$

$$|\ln(\sqrt{|x|})| \quad (3)$$

$$\frac{(1+x)(\sin x + \cos x)}{x + 3 + \tan x} \quad (4)$$

$$\sqrt{|x[x + 3(\sqrt{|\ln|x||} - 2x)]|} \quad (5)$$

$$\int_0^2 x dx \quad (6)$$

$$\sum_{i=1}^k n^i \quad (7)$$

$$|x| - \frac{1}{\frac{5}{x}} \quad (8)$$

$$\prod_{i=1}^l i + \sum_{j=1}^m \sum_{k=0}^n (j \times k) \quad (9)$$

$$\cos(\sqrt{|\sin^{-1}(x)|}) + \cos(\sqrt{|\sin^{-1}(\cos y)|}) \quad (10)$$

$$\sum_{i=0}^l i + \prod_{j=1}^m j + \log_n(x) \quad (11)$$

$$\frac{\ln \left| \frac{1}{x} \right| + \sqrt{\frac{x-1}{x+1}}}{\sqrt{|x|}} \times x! \quad (12)$$

4.12 References

Awde, A., M D. Hina, C Tadj, Y Bellik et A Ramdane-Cherif. 2006a. « A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users ». In *The First International Conference on Grid and Pervasive Computing*

- (GPC) (Taichung, Taiwan, 3-5 May 2006). p. 620-633. Lecture Notes in Springer-Verlag.
- Awde, A., M. D. Hina, C. Tadj, Y. Bellik et A. Ramdane-Cherif. 2009. « An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually- Impaired Users ». *Journal of Multimedia (JMM)*. Vol. 4, No. 4, 2009.
- Awde, A., M. D. Hina, Y. Bellik, A. Ramdane-Cherif et C. Tadj. 2007. « Task Migration in a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users ». In *2nd International Conference on Grid and Pervasive Computing (GPC)* (Paris, France, 2-4 May 2007). p. 459-471. Lecture Notes in Springer-Verlag.
- Awde, A., M. D. Hina, C. Tadj, A. Ramdane-Cherif et Y. Bellik. 2006b. « Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users ». In *ISIE 2006, IEEE International Symposium on Industrial Electronics* (Montreal, QC, Canada, July 9-13, 2006).
- Batusic, M., K. Miesenberger et B. Stöger. 1998. « Labrador: A Contribution to Making Mathematics Accessible for the Blind ». In *6th International Conference on Computers Helping People with Special Needs (ICCHP)* (Vienna, Austria & Budapest, Hungary, 31 August - 4 September 1998). p. 307-315. Springer Verlag.
- Bellik, Y. 1995. « Interfaces multimodales : concepts, modèles et architectures. ». Thèse de doctorat en informatique, Paris, Université de Paris-Sud XI Orsay.
- Cahill, H., C. Lineham, J. McCarthy, G. Bormans et J. Engelen. 1995. « Ensuring Usability in MATHS ». In *The European Context for Assistive Technology, Assistive Technology Research Series*. p. 66-69. Amsterdam: IOS Press.
- Commission Evolution du Braille Francais*. 2001. Notation Mathématique Braille, mise à jour de la notation mathématique de 1971.
- Cormen, T. H., C. E. Leiserson, R. L. Rivest et C. Stein. 2001. *Introduction to Algorithms*, second. MIT Press and McGraw-Hill.
- Demeure, A., et G. Calvary. 2006. « Le Modèle d'Evolution en Plasticité des Interfaces: Apport des Graphes Conceptuels ». In *15ème Conférence francophone sur l'Interaction Homme-Machine (IHM 2003)*. Caen, France: ACM Press.
- Edwards, A. D. N., et R. D. Stevens. 1994. « A Multimodal Interface for Blind Mathematics Students ». In *INSERM* (1994). p. 97 - 104. Paris, France.
- Epheser, H., D. Pograniczna et K. Britz. 1992. « Internationale Mathematikschrift für Blinde, Deutsche Blindenstudienanstalt ». Marburg (Lahn).

- Ferreira, H., et D. Freitas. 2004. « Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project ». In *9th International Conference on Computer Helping People with Special Needs (ICCHP)*. p. 678-685. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Ferreira, H., et D. Freitas. 2005. « AudioMath: Towards Automatic Readings of Mathematical Expressions ». In *Human-Computer Interaction International (HCII)*. Las Vegas, Nevada, USA.
- Garlini, P., et F. Fogarolo. 2003. *LAMBDA: Linear Access to Mathematics for Braille Device and Audio Synthesis - Analysis of the User's Needs*. Padova, Italy: University of Padova.
- Jennings, N. R., et M. J. Wooldridge. 1998. « Applications of Intelligent Agents ». In *Agent Technology: Foundations, Applications, and Markets*, sous la dir. de Jennings, Nicholas R., et Michael J. Wooldridge. p. 3-28. Springer-Verlag: Heidelberg, Germany.
- Karshmer, A. I., G. Gupta, S. Geiiger et C. Weaver. 1998. « Reading and Writing Mathematics: the MAVIS Project ». In *Proceedings of the third international ACM conference on Assistive technologies (ASSETS)*. p. 136-143. Marina del Rey, California, USA: ACM SIGCAPH Computers and the Physically Handicapped
- McCowan, I., D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner et H. Bourlard. 2005. *On the Use of Information Retrieval Measures for Speech Recognition Evaluation*. Coll. « IDIAP Research Report ».
- Moço, V., et D. Archambault. 2003. « VICKIE: A Transcription Tool for Mathematical Braille ». In *7th European Conference for the Advancement of Assistive Technology in Europe (AAATE)* (September 2003). Dublin, Ireland: IOS press.
- Nemeth, A. 1972. « The Nemeth Braille Code for Mathematics and Science Notation ». American Printing House for the Blind.
- Podevin, A. 2002. « Accès aux formules mathématiques par des personnes non-voyantes : étude et définition d'une méthode adaptée ». DEA, Université de CAEN.
- Preddy, M., J. Gardner, S. Sahyun, et D. Skrivanek. 1997. « Dotsplus: How-to Make Tactile Figures and Tactile Formatted Math ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA.
- Raman, T. V. 1998. *Audio System for Technical Readings*, 1410, 1st edition. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 137 p.

- Rousseau, C., Y. Bellik, F. Vernier et D. Bazalgette. 2006. « A Framework for the Intelligent Multimodal Presentation of Information ». *Signal Processing Journal*, vol. 86, n° 12, Special Issue on Multimodal Interface (12 April 2006), p. 3696 - 3713.
- Schwebel, F., et R. Goiffon. 2005. « BraMaNet: Quelques règles simples à connaître pour qu'un aveugle puisse lire vos documents mathématiques et vos pages web ». In *Journées nationales Caen* (Caen, France, October 2005).
- Stöger, B., K. Miesenberger et M. Batusic. 2004. « Mathematical Working Environment for the Blind Motivation and Basic Ideas ». In *9th International Conference on Computers Helping People with Special Needs (ICCHP)*. p. 656-663. Springer Verlag, Lecture Notes in Computer Science (LNCS).
- Weiss, G. 1999. « Multiagent systems ». MIT-Press.
- Wooldridge, M. 2002. *An Introduction to Multi-agent Systems*. Chichester, UK: Wiley.

CHAPITRE 5

MIDDLEWARE FOR UBIQUITOUS ACCESS TO MATHEMATICAL EXPRESSIONS FOR VISUALLY-IMPAIRED USERS

Ali Awde¹, Manolo Dulva Hina^{1,2}, Chakib Tadj¹

¹Département de Génie Électrique, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

²PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines
45, avenue des États-Unis, 78035 Versailles Cedex, France

Yacine Bellik³

³Laboratoire LIMSI-CNRS, Université de Paris-Sud
B.P. 133, 91043 Orsay, France

Amar Ramdane-Cherif^{2,4}

⁴LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines,
Bâtiment Descartes, 45 avenue des États-Unis - 78000 Versailles, France

Ce chapitre a fait l'objet d'une publication dans la revue *Ubiquitous Computing and Communication Journal (UBICC)*, Vol. 3, No. 5, 2008.

Résumé

Fournir aux utilisateurs déficients visuels un accès omniprésent aux informations est un problème difficile. En effet, un système ubiquitaire engage l'utilisation de plusieurs médias et modalités, et doit prendre en considération le contexte de l'interaction dans leur configuration. Dans la littérature, il existe plusieurs systèmes offrant l'accès aux expressions mathématiques pour les non-voyants dont la majorité limite la mobilité de l'utilisateur et ils ne permettent l'accès qu'à une seule forme de présentation. Nous remédions à ces lacunes en fournissant un intergiciel qui fournit un accès omniprésent aux expressions mathématiques. Notre système a la capacité de présenter l'expression sous plusieurs formes. Il doit sélectionner les formes de présentations optimales selon le contexte de l'interaction et les préférences de l'utilisateur. La configuration de la forme de présentation sélectionnée, y compris ses attributs et la qualité de ses services (QoS), s'adapte également aux préférences de l'utilisateur. Cet article discute des défis rencontrés durant la conception de notre système adaptatif et présente les solutions à chacun de ces défis. Nous présentons les concepts et les

principes appliqués ainsi que leurs validations par des études de cas et des spécifications formelles. Ce travail fait partie de notre contribution pour rendre l'information accessible aux utilisateurs ayant des déficiences visuelles et plus spécifiquement la rendre omniprésente.

Mots clés: Système pervasif, mathématiques pour les non-voyants, système multimodal, interaction homme-machine, système adaptatif.

Abstract

Providing visually-impaired users with ubiquitous access to mathematical expressions is a challenge. Our investigation indicates that most, if not all, of the current state-of-the-art systems and solutions for presentation of mathematical expressions to visually-impaired users are generally not pervasive, that they do not take into account the user's interaction context (i.e. combined contexts of the user, his environment and his computing system) into their system's configuration and that they present mathematical expressions in only one format. We address these weaknesses by providing a middleware that provides a ubiquitous access to mathematical expressions. Our middleware gathers various suppliers in which one supplier is selected based on its suitability to the given instance of interaction context and of the user's preferences. The configuration of the chosen supplier, including its quality of service (QoS) dimensions, is also adaptive to the user's preferences. This paper discusses the challenges in designing this middleware and presents a proposed solution to each of these challenges. This paper discusses the concepts and principles applied on the middleware design as well as their validation through case studies and formal specification. This work is intended to contribute on the ongoing research to make informatics accessible to handicapped users, specifically providing them ubiquitous access to mathematical expressions.

Keywords: pervasive computing, visually-impaired users, mathematical expressions presentation, multimodal computing, adaptive system.

5.1 Introduction

In pervasive computing (Satyanarayanan, 2001), the user can continue working on his computing task whenever and wherever he wishes. The *task* is usually realized through the use of one or more software applications. If the task is the ubiquitous access to mathematical expressions for visually-impaired users, then the applications are those that will correctly present mathematical expressions to these users with special needs. There are available applications that can do the task. Hence, instead of inventing another one, it is more practical to collect all these applications and put them in one middleware. The middleware then selects an *application*, henceforth called a *supplier*, based on its suitability to the given context. The *quality-of-service* (QoS) dimensions of the chosen supplier are also configured based on user's preferences.

To realize the ubiquity of documents containing mathematical expressions, the document is stored in a server and its copy is replicated on every member of the server group, making it accessible to the user whenever and wherever he wishes. The mathematical document serves as input file to the middleware.

This research work is a result of an investigation of the current state-of-the-art systems and solutions in which we found out that the available systems for presentation of mathematical expressions to visually-impaired users are not pervasive; that most of them do not take into account the current interaction context in the system's configuration and that the prevailing systems provide only one type of format for the presentation of mathematical expressions. As such, they are not multimodal. Our work addresses these weaknesses.

This paper presents the concepts and principles used in the design of this middleware. Data validation is done through case studies and formal specifications. This proposed middleware is functional under these assumed conditions: (i) that wired and wireless communication facilities are available to support our system, (ii) that the mathematical expression as input to the middleware is stored in a MathML or LaTeX input file, and (iii) that there are available

application suppliers and media devices to support the optimal modalities selected by the middleware.

Apart from this introductory section, the rest of this paper is structured as follows. Section 5.2 provides a brief review of the state of the art related to our work; Section 5.3 lists down the challenges, proposed solutions and contributions of our work. Section 5.4 presents the infrastructure and system architecture of our middleware. In Section 5.5, we explain the concept of machine learning in relation to the system design so it adapts seamlessly to the given interaction context. Section 5.6 presents our methodology to configure an application supplier and optimize settings, taking into account user's satisfaction. Section 5.7 discusses interaction contexts, modalities and media devices that characterize a typical scenario of a blind user trying to access mathematical expressions. Through scenario simulations in Section 5.8, we show the design specification of our infrastructure. Future works and conclusion are presented in Section 5.9.

5.2 Review of the State of the Art

To a visually-impaired user, the mere understanding of a mathematical expression is already a challenge. It requires repeated passage on an expression where user often skips secondary information only to revert back to it again and again until he fully grasps the meaning of the expression. A complicated task like this is explained in (Stöger, Miesenberger et Batusic, 2004). Fortunately, some applications were developed to lessen the complexity of performing similar task, some of them are MathTalk (Edwards et Stevens, 1994), Maths (Cahill et al., 1995) and AudioMath (Ferreira et Freitas, 2004; 2005). MathTalk and Maths convert a standard algebraic expression into audio information. In Maths, the user can read, write and manipulate mathematics using a multimedia interface containing speech, Braille and audio. VICKIE (*Visually Impaired Children Kit for Inclusive Education*) (Moço et Archambault, 2003a) and BraMaNet (*Braille Mathématique sur InterNet*) (Schwebel et Goiffon, 2005) are transcription tools that convert mathematical document (in LaTeX, MathML, HTML, etc.) to its French Braille representation. Labradoor (*LaTeX-to-Braille-Door*) (Batusic, Miesenberger

et Stöger, 1998) converts an expression in LaTeX into its Braille equivalent using the Marburg code. MAVIS (*Mathematics Accessible To Visually Impaired Students*) (Karshmer et al., 1998) supports LaTeX to Braille translation via Nemeth code.

As a background, some special Braille notations have been developed for mathematics, different countries adopting different Braille code notations. Some of these are the *Nemeth Math code* (Nemeth, 1972) which is used in the USA, Canada, New Zealand, Greece and India; the *Marburg code* (Epheser, Pogranczna et Britz, 1992) used in Germany and Austria, the *French Math code* (*Commission Evolution du Braille Francais*, 2001).

Lambda (Edwards, McCartney et Fogarolo, 2006) translates expression in MathML format into multiple Braille Math codes that are used in the European Union. ASTER (*Audio System for Technical Readings*) (Raman, 1994) reformats electronic documents written in LaTeX into their corresponding audio equivalent. AudioMath provides vocal presentation of mathematical content encoded in MathML format.

Of all the tools cited above, none is pervasive. None takes into account the user's interaction context into its configuration. Our approach, therefore, is to get the strength of each tool, integrate each one of them into our work in order to build a middleware that (1) broadens the limits of utilization, (2) provides the user with opportunities to access mathematical expressions written in either MathML or LaTeX format, and (3) selects appropriate application supplier and its configuration depending on the given instance of interaction context.

In (Awde et al., 2006b), the use of multimodal interaction for non-visual application was demonstrated. The multimodal system selects the modality over another after determining its suitability to a given situation. Multimodality is important to visually-impaired users because it provides them equal opportunities to use informatics like everybody else.

Recently, agents or multi-agent systems (MAS) (Ferber, 1995; Weiss, 1999) have been widely used in many applications, such as on a large, open, complex, mission-critical systems like air traffic control (Jennings et Wooldridge, 1998). Generally, agency is preferred over traditional techniques (i.e. functional or object-oriented programming) because the latter is inadequate in developing tools that react on environment events. Some works on MAS for visually-impaired users include (Awde et al., 2006a; Awde et al., 2006b). In contrast to those works, ours focuses on ubiquitous access to mathematical expressions.

Incremental machine learning (IML) is a progressive acquisition of knowledge. In the literature, various IML algorithms exist but in this work, *supervised learning* is adopted because limited scenarios have been considered. *Supervised learning* is a ML method by which the learning process produces a function that maps inputs to certain desired outputs. More details of ML design are in Section 5.5.

Our focus has always been pervasive multimodality for the blind. This work was initially inspired by (Sousa et al., 2006). As our work evolves, however, the domain of application and its corresponding optimization model becomes completely different as this paper is reflective of our intended user. The methodology is different; this work adopts *machine learning* (ML) to acquire knowledge. Such knowledge is stored onto the *knowledge database* (KD) so that it can be made omnipresent, accessible anytime and anywhere via wired or wireless networks.

A major challenge in designing systems for blind users is how to render them autonomy. To this end, several tools and gadgets have emerged, among them are the GPS (global positioning system), walking stick that detects user context (Jacquet et al., 2004) and a talking Braille (Ross et Lightman, 2005). Our work aims the same. Ours is adaptive to user's condition and environment. Through pervasive computing networks, the ML acquired knowledge, and user's task and profile all become omnipresent, and the system's knowledge on the feasible configuration for the user's task is generated without any human intervention.

5.3 Challenges, Proposed Solutions and Contribution

In this section, we define the requirements in designing a pervasive system that will present mathematical expressions to visually impaired users by posing specific technical challenges to solve and then describe the approach to address them.

Contribution 1: Design of a scheme in which a mathematical document can be made accessible anytime and anywhere to visually-impaired users.

Related Questions: Given that the computing task is to provide ubiquitous access to a mathematical document, how can this document be made accessible whenever and wherever the user wishes to given that the user can be either stationary or mobile? Also, given that any computing machine or server may fail, what configuration must be established in order to ensure the ubiquity of such document?

Proposed Solutions: A mathematical expression becomes accessible if it is stored as a MathML or LaTeX document. This document, along with the user's data and profile, becomes available for ubiquitous computing if it is stored in a server. The content of this server becomes omnipresent if it is replicated to the other members of the server group. Hence, the server group renders the user's document ubiquitous, available anytime and anywhere. The user connects to a member of the server group upon login. This server, one that is closest to the user's location, becomes a point of access for the user's profile, data and mathematical document.

Contribution 2: Conceptualization of an adaptive system that selects an optimal application supplier, one that provides visually-impaired users access to mathematical expressions, based on the given instance of interaction context.

Related Questions: How do we associate a mathematical document to an optimal application supplier? What would be the basis of such association? If this optimal supplier happens to be not available, how will the system find a relevant replacement?

Proposed Solutions: We incorporate machine learning mechanism wherein the system remembers scenarios – the input conditions and the resulting output conditions of each scenario. This learning mechanism assists the system in selecting the optimal supplier based on the given scenario. Different suppliers are apt for different scenarios. Part of this learning mechanism is the ranking of application suppliers; the ranking takes into account the user's preference. In general, it is possible that in a given scenario, two or more suppliers may be found suitable for invocation, with the top-ranked supplier being activated by default. When the chosen supplier is not available, then the next-ranked supplier is taken as its replacement.

Contribution 3: Design of a system that is tolerant to faults due to failure of servers and media devices.

Related Questions: In case of server failure, what must be done to guarantee that the current user's data, profile and machine knowledge are not lost? If selected media devices are malfunctioning, what must be done to keep the system remain operational and persistent?

Proposed Solutions: Like any pervasive system, ours assume that there are many members of the server group by which the user can connect. During the time that the server to which the user is connected is down, the user may continue working and that his data, profile and machine knowledge are all stored in the local cache. As soon as the server is up, the server starts communicating with the user's terminal, and the server's copy of ubiquitous information is updated. Afterwards, the server's ubiquitous information is sent to all members of the server group. Our system also adopts a ranking scheme for media devices' suitability to a given modality. When a highly-ranked media device is not available or has failed, a lower-ranked media device is taken as its replacement. If no replacement is found, then the system re-evaluates the current instance of interaction context and accordingly chooses the optimal supplier and modalities. Afterwards, the system activates the available media device(s) that support the chosen modalities.

Contribution 4: Design of a system that provides configuration suitable to the user's satisfaction.

Related Questions: How do we represent and quantify user's satisfaction? What parameters are used to measure the user's satisfaction with regards to system configuration?

Proposed Solutions: We use the utility measure $[0,1]$ to denote user satisfaction in which 0 = condition is inappropriate while 1 = user is happy with the condition. In general, the closer the configuration setting is to the user's pre-defined preferences, the higher is the user satisfaction. In this work, we consider the application supplier and its QoS dimensions as parameters for system configuration.

5.4 Infrastructure and System Architecture

Here, we present the architectural framework of our pervasive multimodal multimedia computing system that provides ubiquitous access to mathematical expression for blind users.

5.4.1 System Architecture

Figure 5.1 shows the layered view of our pervasive multimodal multimedia computing system. It is a multi-agent system organized into layers. Layering is a method by which communication takes place only between adjacent layers. It limits the undesirable ripple-effect propagation of errors within the boundary of the layer involved. The various system layers and their functionalities are:

The Presentation Layer : Here, the mathematical expression is presented to the user via optimal presentation format. This layer is responsible for the conversion of the mathematical expression's encoded format (from Data Analysis Layer) into its final presentation format.

The Data Access Layer : Here, users may edit or search for a term in a mathematical expression.

The Data Analysis Layer : Here, the mathematical expression presentation format is selected based on available media devices and supplier, and interaction context. Also, an agent takes in a MathML or LaTeX expression and converts it to its encoded format. Furthermore, its machine learning component selects the optimal presentation format of the expression based on parameters just cited.

The Control and Monitoring Layer : This layer controls the entire system, coordinating the detection of user's interaction context, and the manipulation and presentation of the mathematical expression. In this layer, there is an agent that retrieves MathML or LaTeX document and the user's profile.

The Context Gathering Layer : Here, the current interaction context is detected. Also, the user's media devices and application supplier preferences are also detected. The layer's agents detect the available media devices and the contexts of the user's environment and of the computing system.

The Physical Layer : It contains all the physical entities of the system, including media devices and sensors. The raw data from this layer are sampled and interpreted and forms the current instance of interaction context.

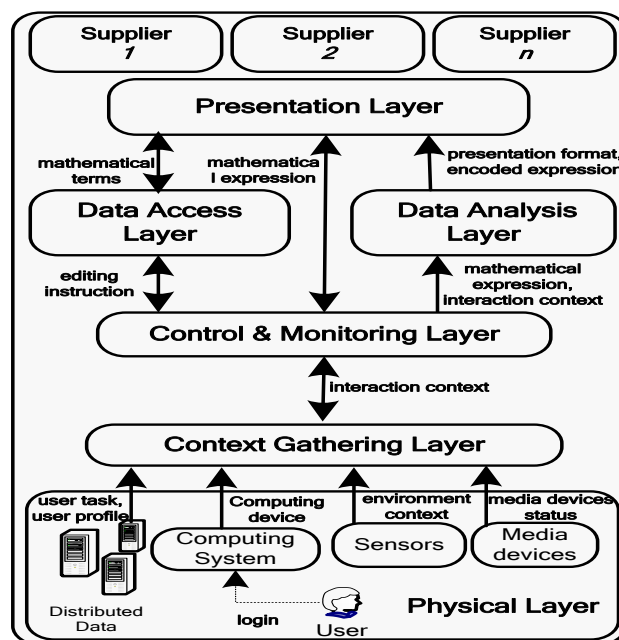


Figure 5.1 The architectural abstraction of a generic MM computing system for visually-impaired users.

5.4.2 The Ubiquitous Mathematical Document

We define *task* as user's work to do and its accomplishment requires the use of a computing system. Hence, in this paper, the user's task is to access (i.e. read, write, edit) a mathematical expression in a document in a ubiquitous fashion.

Creating a Mathematical Document

The processes of creating a ubiquitous mathematical document are as follows: (i) Using Braille or speech as input media, the user writes a mathematical equation using the syntax of MathML or LaTeX; (ii) There is a supplier (selected by the middleware as the most appropriate to the current interaction context) that takes in this input equation and yields its corresponding output – via speaker or Braille (note that in a Braille terminal, there is a separate section where the user can touch/sense the output); (iii) The supplier saves the document in the local cache; and (iv) Our middleware saves the document onto the server which is also propagated to other members of the server group. Figure 5.2 shows a specimen fraction (in bi-dimensional form) and its equivalent representation in MathML, LaTeX and Braille.

Reading and Presenting the Mathematical Expression in a Document

A ubiquitous mathematical document becomes readily available to the user once a connection is made between the user's terminal and a member of the server group. In general, the processes for reading or presenting a mathematical document to the user are as follows: (i) Our middleware retrieves the mathematical document from the server and download it to the user's cache/local terminal; (ii) The middleware selects a supplier based on the given interaction context; (iii) The middleware reads or presents the mathematical equation to the visually-impaired user via speech or Braille.

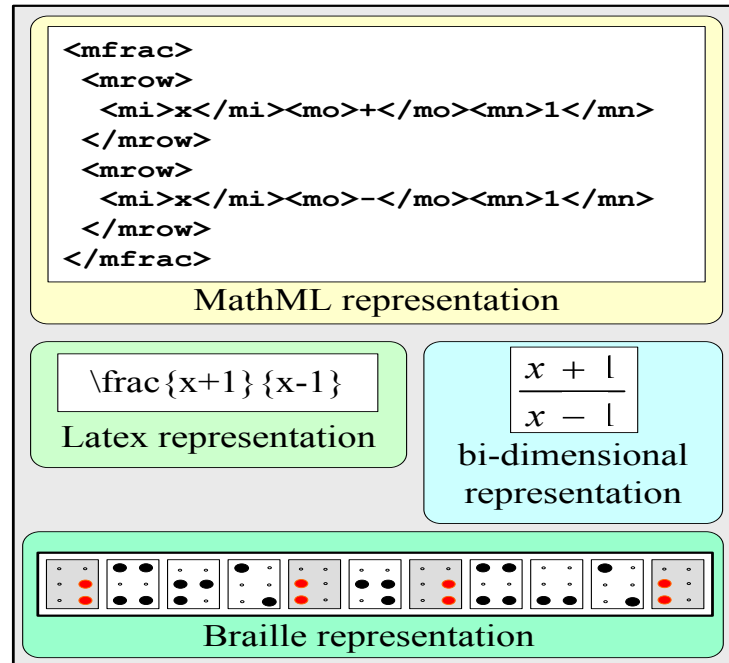


Figure 5.2 A fraction in bi-dimensional form and its corresponding equivalent in LaTeX, MathML and Braille.

Modifying a Mathematical Document

At any time, the user may delete or modify a mathematical equation within a document. Once a mathematical document is identified, our middleware opens up the file and our *Data Access Layer* (see Section 5.4.1) allows users to search for a term within the equation and edit same. Once editing is done, the document is saved accordingly.

5.4.3 Anytime, Anywhere Mathematical Computing

As shown in Figure 5.3, computing with mathematical equations for visually-impaired users is possible anytime, anywhere the user wishes to. In the diagram, the user is assumed to be working at home using our middleware.

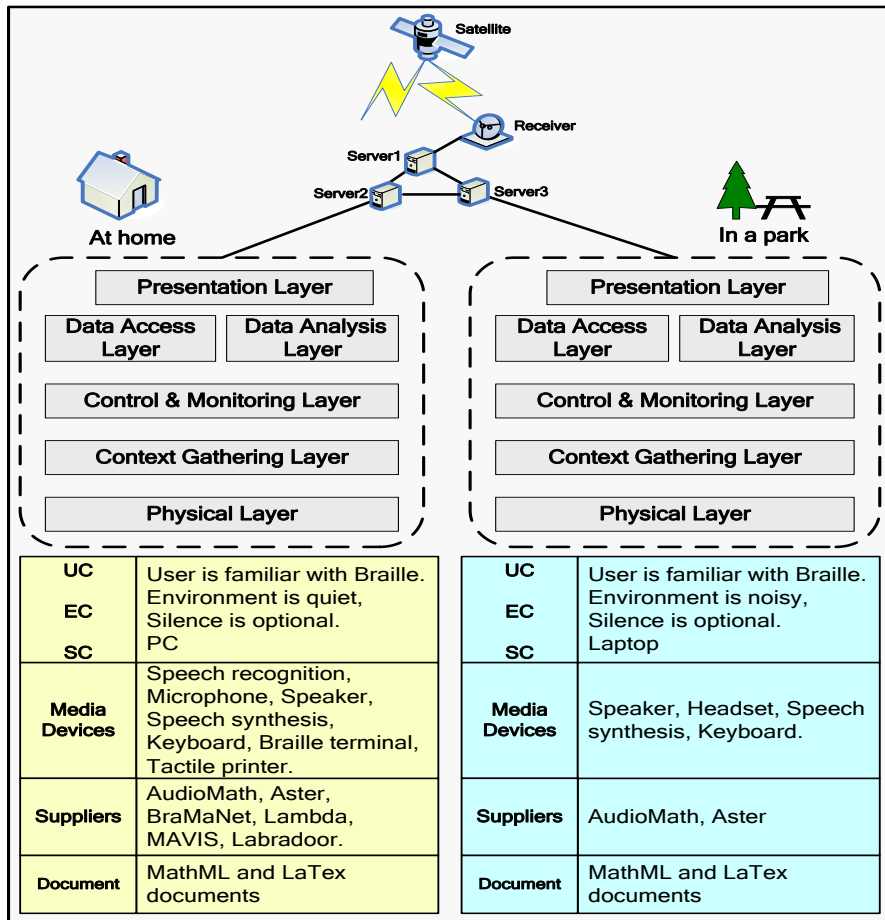


Figure 5.3 An anytime, anywhere adaptive computing with mathematical document for visually-impaired users.

The system detects the user’s interaction context (i.e. getting the values of parameters that make up the contexts of the user, his environment and his computing system – more details in Section 5.7), detects the currently available media devices and selects the most suitable application supplier, and eventually accessing the user’s mathematical document. The user then reads or modifies the mathematical equations in this document or writes a new one. At the end of the user’s session, his mathematical document is saved onto the server and on all other members of the server group.

The user may continue working on the same mathematical document in, say, a park. In this new setting, the middleware detects the user’s interaction context and the available media devices. The system then provides an application supplier that is likely different from what

the user used when he was at home. Although different in setting, the point is that the user is still able to continue working on an interrupted task.

5.5 Designing an Interaction Context Adaptive System

Here, we elaborate on the system's knowledge acquisition that makes it adaptive to the given interaction context.

5.5.1 Theoretical Machine Learning

In *machine learning* (ML), a program is said to have *learned* from experience E with respect to a class of task T and a performance measure P if its performance P at task T improves with experience E (Mitchell, 1997). In this work, the learning problem is defined as follows: (i) *task T*: selecting the modalities (and later the media devices) that are appropriate to the given instance of interaction context, (ii) *performance P*: measure of the selected modalities' suitability to the given interaction context, as given by their "suitability score" (iii) *training experience E*: various combinations of possible modalities for visually-impaired users.

Figure 5.4 shows the functionality of a generic ML-based system. In general, a *scenario* is an event that needs to be acted upon appropriately. An input to the ML component is the *pre-condition* of a scenario. Here, the pre-condition scenario is a specific instance of an interaction context. The ML component analyzes the input, performs calculations and decisions and yields an output called the *post-condition* of a scenario. For knowledge acquisition purposes, the pre- and post-conditions of every scenario are stored in a databank called the *scenario repository* (SR). Each entry in SR forms a distinct scenario.

When a ML component is given a situation (i.e. pre-condition of a scenario), it consults SR for a similar entry. If a match is found then it simply retrieves the post-condition scenario and implements it. If no match is found (i.e. scenario is new), then the ML component, using its acquired knowledge, performs calculations producing the post-condition of the scenario. If

accepted by the expert (i.e. the user), then the complete scenario is stored onto SR as a new entry. This process is performed whenever a new scenario is encountered. Over time, the ML component accumulates enough knowledge that it knows the corresponding reaction (i.e. post-condition) given a certain situation (i.e. pre-condition).

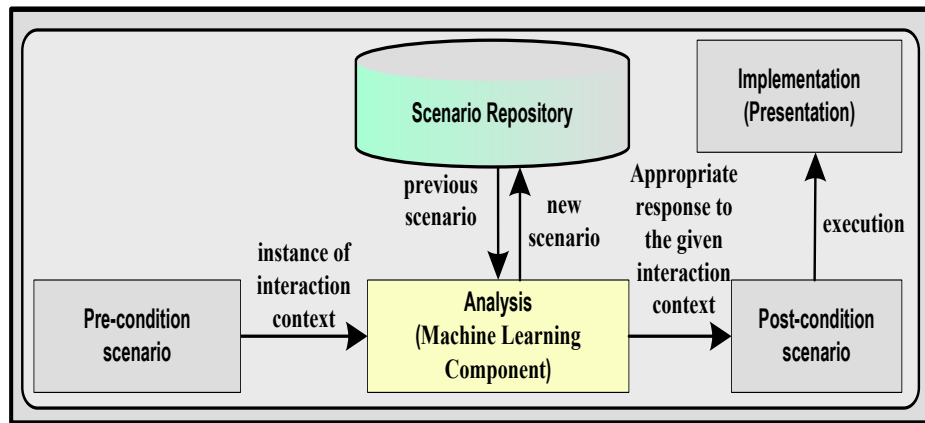


Figure 5.4 A generic ML system: an instance of interaction context serves as input to a ML component yielding a corresponding post-condition result.

Incremental machine learning (IML) is a progressive acquisition of knowledge. In the literature, various IML algorithms exist, such as the candidate elimination (Mitchell, 1997), and ID5 (i.e. the *iterative dichotomizer version 5*, an incremental implementation of ID3 which itself is a decision-tree induction algorithm developed by Quinlan) (Utgoff, 1988). In this work, *supervised learning* is adopted because limited scenarios have been considered. *Supervised learning* is a ML method by which the learning process produces a function that maps inputs to certain desired outputs. Let there be a set of inputs \mathbf{X} with n components and a set of outputs \mathbf{Y} with m components. Let f be the function to be learned which will map some elements in \mathbf{X} to the elements in \mathbf{Y} , and h be the hypothesis about this function. Furthermore, let \mathbf{X} be the set of interaction context instances, i.e. set of pre-condition scenarios. Hence, \mathbf{Y} would be a set of post-condition scenarios, and the mapping between the pre- and post-condition scenarios is denoted by $\mathbf{f}: \mathbf{X} \rightarrow \mathbf{Y}$. See Figure 5.5.

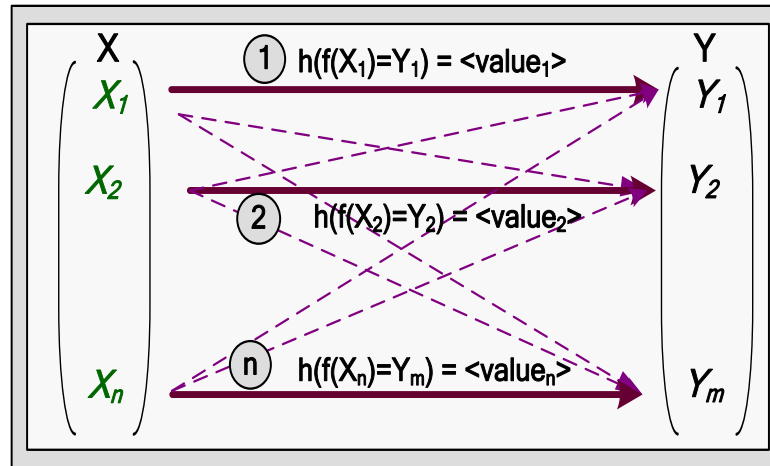


Figure 5.5 The relation between pre-condition scenarios and the post-condition scenarios using supervised learning.

As shown in the diagram, X_i = an element of \mathbf{X} , and $i \in 1 \dots n$, and Y_j = a element of \mathbf{Y} , and $j \in 1 \dots m$. When a new pre-condition X_{new} occurs, the learning system finds the best possible output value Y_{best} using hypothesis. In general, the learner compares the results of hypothesis functions $h(f(X_{new}) = Y_1)$, $h(f(X_{new}) = Y_2)$, \dots , $h(f(X_{new}) = Y_m)$ and selects one that yields the best score. The new case is the newly-acquired knowledge which is later on added to the knowledge repository. This learning process is called *incremental machine learning*.

5.5.2 Basic Principles of Interaction Context

In this paper, the following logic symbols are used: \otimes = Cartesian product yielding a set composed of tuples, ε_1 = set of positive integers excluding zero, \forall = universal quantifier, \exists = existential quantifier, the basic logical connectives \wedge (AND) and \vee (OR), and the intervals $[x, y]$ which denotes that a valid data is in the range of x and y , and $(a, b]$ which denotes that a valid data is higher than a and up to a maximum of b .

The interaction context, \mathbf{IC} =, is a set of all possible interaction contexts. At any given time, a user has a specific interaction context i denoted \mathbf{IC}_i , $1 \leq i \leq \max$. Formally, an interaction context is a tuple composed of a specific user context (\mathbf{UC}), environment context (\mathbf{EC}) and system context (\mathbf{SC}). An instance of \mathbf{IC} may be written as:

$$\mathbf{IC}_i = \mathcal{J}\mathbf{C}_k \otimes \mathcal{I}\mathbf{C}_l \otimes \mathbf{C}_m \quad (5.1)$$

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

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

$$\mathbf{UC}_k = \bigotimes_{x=1}^{\max_k} \mathbf{ICParam}_{kx} \quad (5.2)$$

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

$$\mathbf{EC}_l = \bigotimes_{y=1}^{\max_l} \mathbf{ICParam}_{ly} \quad (5.3)$$

$$\mathbf{SC}_m = \bigotimes_{z=1}^{\max_m} \mathbf{ICParam}_{mz} \quad (5.4)$$

The first knowledge the ML component must learn is to relate the interaction context to an appropriate modality. In general, a modality is possible if there exists at least one modality for data input and at least one modality for data output. Given a modality set $\mathbf{M} = \{\mathbf{V}_{in}, \mathbf{T}_{in}, \mathbf{V}_{out}, \mathbf{T}_{out}\}$ wherein \mathbf{V}_{in} = *vocal input*, \mathbf{V}_{out} = *vocal output*, \mathbf{T}_{in} = *tactile input* and \mathbf{T}_{out} = *tactile output* then modality is possible under the following condition:

$$\text{Modality Possible} = (V_{in} \vee \Gamma_{in}) \wedge (V_{out} \vee \Gamma_{out}) \quad (5.5)$$

The failure of modality, therefore, can be specified by the relationship:

$$\begin{aligned} \text{Modality Failure} = & (V_{in} = \text{failed}) \wedge (T_{in} = \text{failed}) \vee \\ & ((V_{out} = \text{failed}) \wedge (T_{out} = \text{failed})) \end{aligned} \quad (5.6)$$

Let M_j = element of the power set of M , that is, $M_j \in \mathcal{P}(M)$ where $1 \leq j \leq mod_max$ (maximum modality). Also, let \hat{M} = the most suitable M_j for a given interaction context IC_i . As stated, X is a set of pre-condition scenarios. Hence, the relationship between X and IC may be written as $X_i = IC_i$. For simplicity purposes, we let the pre-condition set X be represented by interaction context IC . Each interaction context i , denoted as IC_i , is composed of various attributes of n components, that is, $IC_i = (A_1, A_2, \dots, A_n)$ where an attribute may be a parameter belonging to UC or EC or SC . We also let the set of post-condition Y be represented by a set of modality M . Let the function f map the set of IC to the set of M , in which h calculates the suitability score of such mapping, that is,

$$h(IC_i) \rightarrow M_j = \langle \text{suitability_score} \rangle \quad (5.7)$$

In Mathematics, function h can be written as:

$$h = P(M_j/IC_i) \quad (5.8)$$

which should be read as the probability of the occurrence of M_j given IC_i . To simplify calculation, *Bayes Theorem* (Kallenberg, 2002), given below, can be adopted:

$$P(M_j/IC_i) = \frac{P(IC_i/M_j) \times P(M_j)}{P(IC_i)} \quad (5.9)$$

The implementation of Bayes Theorem leads to the *Naive Bayes algorithm* (Mitchell, 1997). The Naive Bayes algorithm is a classification algorithm that assumes that IC_i attributes

A_1, \dots, A_n are all conditionally independent of one another given a post condition \mathbf{M}_j . The representation of $P(\mathbf{IC}_i | \mathbf{M}_j)$ becomes:

$$P(\mathbf{IC}_i | \mathbf{M}_j) = \prod_{i=1}^n P(A_i | \mathbf{M}_j) \quad (5.10)$$

Here, our goal is to train a classifier that, given a new \mathbf{IC}_i to classify, will provide the probability distribution over all possible values of \mathbf{M} (i.e. $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_m$). Given that $\mathbf{IC}_i = (A_1, A_2, \dots, A_n)$, then Eq. (5.9) becomes:

$$P(\mathbf{M}_j | A_1 \dots A_n) = \frac{P(\mathbf{M}_j) P(A_1 \dots A_n | \mathbf{M}_j)}{\sum_{k=1}^m P(\mathbf{M}_k) P(A_1 \dots A_n | \mathbf{M}_k)} \quad (5.11)$$

Equation 5.11 can also be written as:

$$P(\mathbf{M}_j | A_1 \dots A_n) = \frac{P(\mathbf{M}_j) \prod_{i=1}^n P(A_i | \mathbf{M}_j)}{\sum_{k=1}^m P(\mathbf{M}_k) \prod_{i=1}^n P(A_i | \mathbf{M}_k)} \quad (5.12)$$

which is the fundamental equation for the Naive Bayes classifier. Given a new instance of interaction context $\mathbf{IC}_{\text{new}} = (A_1 \dots A_n)$, the equation shows how to calculate the probability that \mathbf{M}_j will take on any given value, given the observed attribute values of \mathbf{IC}_{new} and given the distributions $P(\mathbf{M}_j)$ and $P(\mathbf{A}_i | \mathbf{M}_j)$ estimated from the training data (SR). If we are interested only in the most suitable value of \mathbf{M}_j , then we have the Naive Bayes classification rule:

$$h_{\text{best}} = \hat{\mathbf{M}} = \arg \max_j \left(\frac{P(\mathbf{M}_j) \prod_{i=1}^n P(A_i | \mathbf{M}_j)}{\sum_{k=1}^m P(\mathbf{M}_k) \prod_{i=1}^n P(A_i | \mathbf{M}_k)} \right) \quad (5.13)$$

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

$$h_{\text{best}} = \hat{M} = \arg \max_j \left(P(M_j) \prod_{i=1}^n P(A_i | M_j) \right) \tag{5.14}$$

where $P(M_j)$ = the frequency of M_j in SR ÷ cardinality of (SR).

5.5.3 Finding Optimal Modalities for Interaction Context

Given that $M = \{V_{in}, T_{in}, V_{out}, T_{out}\}$, then the power set (i.e. the set of all subsets) of M is given by $\mathcal{P}(M) = \{\{V_{in}\}, \{T_{in}\}, \{V_{out}\}, \{T_{out}\}, \{V_{in}, T_{in}\}, \{V_{in}, V_{out}\}, \{V_{in}, T_{out}\}, \{V_{in}, T_{in}, V_{out}\}, \{V_{in}, T_{in}, T_{out}\}, \{V_{in}, V_{out}, T_{out}\}, \{V_{in}, T_{in}, V_{out}, T_{out}\}, \{T_{in}, V_{out}\}, \{T_{in}, T_{out}\}, \{T_{in}, V_{out}, T_{out}\}, \{V_{out}, T_{out}\}, \{\}\}$. M_j , therefore, evaluates the suitability score of each element of $\mathcal{P}(M)$. $\hat{M} = h_{\text{best}}$ is then chosen from one of these elements. The selected element is one that satisfies Eq. (5.14) and one with the highest suitability score.

5.5.4 Realizing User Task Using Optimal Modalities and Supporting Media Devices

Given that an optimal modality has just been selected based on the given instance of interaction context, the next step is to find the media devices that will support the chosen modality. To do so, let there be a function f_1 that maps a specific modality to its appropriate media device(s) as given by $f_1: \text{Modality} \rightarrow \text{Media, Priority}$. See Figure 5.6.

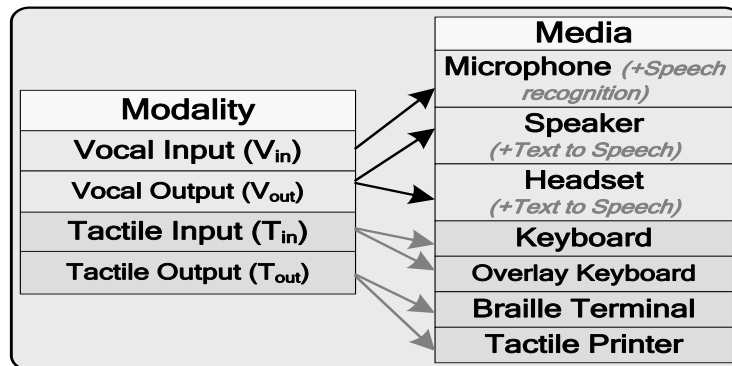


Figure 5.6 Media selections to support a modality.

Throughout this paper, the following acronyms will be used to denote names of media devices: **MIC** = *microphone*, **KB** = *keyboard*, **OKB** = *overlay keyboard*, **SPK** = *speaker*, **HST** = *headset*, **BRT** = *Braille terminal*, **TPR** = *tactile printer*. The availability of supporting media devices is important if a modality is to be implemented. The following is a sample set of elements of \mathbf{f}_1 for visually-impaired users:

$$f_1 = \{(V_{in}, (MIC,1)), (V_{in}, (Speech\ Recognition,1)), (V_{out}, (SPK,1)), (V_{out}, (HST,2)), (V_{out}, (Text-to-Speech,1)), (T_{in}, (KB,1)), (T_{in}, (OKB,2)), (T_{out}, (BRT,1)), (T_{out}, (TPR,2))\}$$

Note that although *media* technically refers to hardware components, a few software elements, however, are included in the list as vocal input modality would not be possible without speech recognition software and the vocal output modality cannot be realized without the presence of text-to-speech translation software. From \mathbf{f}_1 , we can obtain the relationship in implementing multimodality:

$$\begin{aligned} f_1(V_{in}) &= (MIC,1) \wedge (Speech\ Recognition,1) \\ f_1(V_{out}) &= ((SPK,1) \vee (HST,2)) \wedge (Text-to-Speech,1) \\ f_1(T_{in}) &= (KB,1) \vee (OKB,2) \\ f_1(T_{out}) &= (BRT,1) \vee (TPR,2) \end{aligned}$$

Therefore, the assertion of modality, as expressed in Eq. (6.5), with respect to the presence of media devices becomes:

$$\begin{aligned} \text{Modality Possible} &= \\ &((MIC \wedge \exists \text{Speech Recognition}) \vee \exists KB \vee \exists OKB) \wedge \\ &(((SPK \vee \exists HST) \wedge \exists \text{Text-to-Speech}) \vee \exists BRT \vee \exists TPR) \end{aligned} \quad (6.15)$$

5.5.5 Machine Learning Training for Selection of Application Supplier

In *supervised learning*, there exists a set of data, called *training set*, from which the learning knowledge is based. The function to be learned maps the training set to a preferred output. The user provides the result of the mapping. A successful mapping, called *positive example*,

is saved while an incorrect mapping, called *negative example*, is rejected. The positive examples are collected and form part of the system's knowledge. The acquired knowledge is then saved onto the *knowledge database* (KD). Figure 5.7 shows the ML knowledge acquisition.

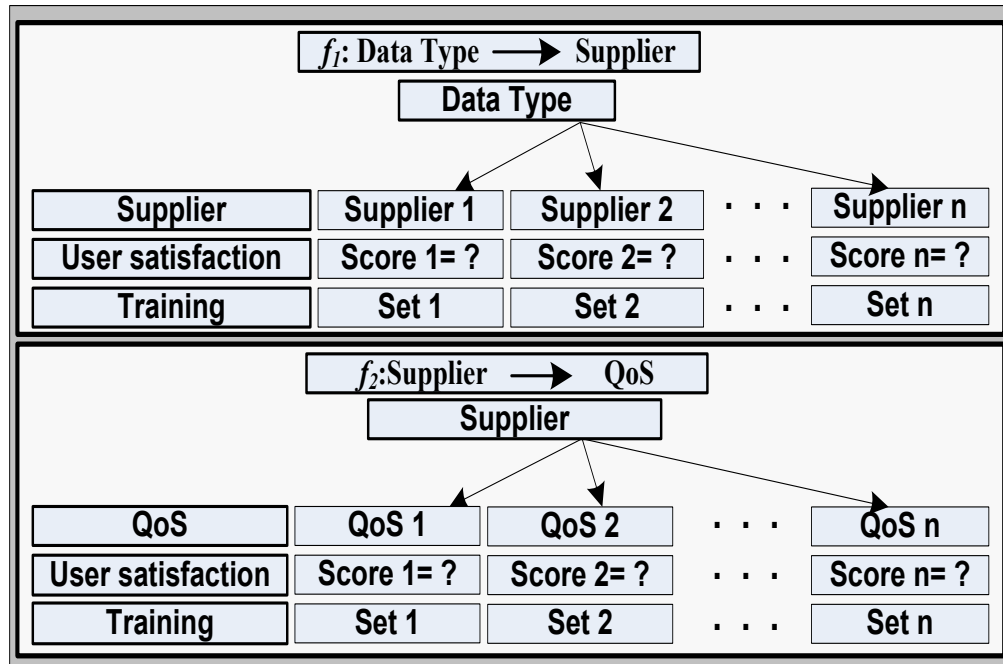


Figure 5.7 The training process for ML knowledge acquisition: (Up) the mapping of data type to a supplier and (Down) building a user's preferred QoS dimensions for a supplier.

A mathematical document of type LaTeX (with filename extension **.tex**) and MathML (with extension **.xml**) is supported by several suppliers. Let there be a function f_2 that maps a given mathematical document (of type MathML and LaTeX) and available media device(s) to the user's preferred supplier and its priority ranking as given by $f_2: \text{Document}, \{\text{media devices}\} \rightarrow \text{Preferred supplier}, \text{Priority}$. For demonstration purposes, let us assume that the user chooses his 3 preferred suppliers and ranks them by priority. The learned function is saved onto knowledge repository and is called user supplier preference.

Given that **Media Devices** = {MIC, KB, OKB, SPK, HST, BRT, TPR} and **Supplier** = {MAVIS, LaBraDoor, ASTER, AudioMath, BraMaNet, Lambda} and **Document** = {.tex, .xml} then the following are some possible mappings within f_2 :

$$f_2 = \{((.tex, BRT) (MAVIS, 1)), ((.tex, BRT) (LaBraDoor, 2)), ((.tex, \{BRT, SPK |HST\}), (ASTER, 1)), ((.tex, \{BRT, SPK|HST\}), (MAVIS, 2)), ((.tex, \{BRT, SPK|HST\}), (LaBraDoor, 3)), ((.tex, \{SPK|HST\}), (ASTER, 1)), ((.xml, \{BRT\}), (Lambda, 1)), (.xml, \{BRT\}), (BraMaNet, 2)), ((.xml, \{BRT, SPK|HST\}), (AudioMath, 1)), ((.xml, \{BRT, SPK|HST\}), (Lambda, 2)), ((.xml, \{BRT, SPK|HST\}), (BraMaNet, 3)), \text{etc.}\}$$

Every supplier has its set of *quality of service* (QoS) dimensions that consumes computing resources. Here, the only important QoS dimensions are those that are valuable to visually-impaired users. A function f_3 creates a mapping of a supplier and its QoS dimensions that the user prefer (f_3 : **Supplier** \rightarrow **QoS dimension j , Priority**) Also, $1 \leq j \leq \text{qos_max}$ (maximum number of QoS dimensions). Priority is of type \mathcal{E}_1 . Since there are many possible values for each QoS dimension, the user arranges these values by their priority ranking. A sample f_3 is given below:

$$f_3 = \{(Lambda, (40 \text{ characters per line}, 1)), (Lambda, (60 \text{ characters per line}, 2)), (Lambda, (80 \text{ characters per line}, 3)), (Lambda, (\text{French Math code}, 1)), (Lambda, (\text{Nemeth code}, 2)), (\text{AudioMath}, (\text{medium volume}, 1)), (\text{AudioMath}, (\text{high volume}, 2)), (\text{AudioMath}, (\text{low volume}, 3)), \text{etc.}\}$$

5.6 Configuration and Optimization of Application Supplier

Here, we discuss the method by which the system would self-configure taking into account the user's satisfaction.

5.6.1 Alternative Configuration Spaces

Given a user task, application suppliers are instantiated to realize the task. For each supplier, however, there are various QoS dimensions that can be invoked in its instantiation. Respecting the user's preferences is the way to instantiate a supplier, but if this is not

possible, the dynamic reconfiguration mechanism should look upon the various configuration spaces and determine the one that is suitable and optimal for the user's needs.

A *QoS dimension* is an application's parameter that consumes *computing resources* (battery, CPU, memory, bandwidth). As an application's QoS dimension improves, then the application's quality of presentation (e.g. clarity of sound, etc.) also improves but at the expense of larger resources' consumption. Given a task and its various applications, the task's QoS dimension space is given by:

$$\text{QoS Dimension space} = \exists \prod_{i=1}^{\text{qos_max}} \bar{D}_i \quad (5.16)$$

In this work, the QoS dimensions that matter are those that are valuable to blind users. For suppliers that convert LaTeX or MathML to Braille, the QoS dimensions of importance are the *number of characters per line*, and the *number of Math codes that it can support*. For suppliers that convert **.tex** and **.xml** document into its audio equivalent, the QoS dimensions are the *volume*, *age of the speaker*, *gender of the speaker*, *words uttered by the speaker per unit of time*, and the *speaker's language*.

5.6.2 Optimizing Configuration of User's Task

An *optimal configuration* is a set-up that tries to satisfy the user's preferences given the current interaction context and available media devices. When the configuration is optimal, 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 tries to achieve a result that is closer to 1.

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

$$\text{Supplier preferences} = h_s^{x_s} \cdot c_s \quad (5.17)$$

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

$$c_{\text{supplier } i} = 1 - \frac{i-1}{n} \quad (5.18)$$

The term $f_s: \text{dom}(s) \rightarrow [0, 1]$ is a function that 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, for a supplier that produces an audio output, the user might prefer one that provides *wave intonation*, or capable of *informing the user of the nature of the next mathematical expression*, etc. For example, if the user listed $n = 3$ preferred features for an application, and the selected supplier supports them all, 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 $f_s = 1 - m/(n + 1) = 1 - 1/4 = 0.75$. In general, the user satisfaction with respect to supplier features is given by:

$$f_{\text{supplier}} = \frac{n-m}{n+1} \quad (5.19)$$

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:

$$h_s = \arg \max \left(\frac{c_s + c_{rep}}{2 \times c_s} \right) \quad (5.20)$$

where c_{rep} is value obtained from Eq. (5.18) for the replacement supplier. The term x_s indicates if change penalty should be considered. $x_s = 1$ if the supplier exchange is done due to the dynamic change of the environment, while $x_s = 0$ if the exchange is instigated by the user.

The algorithm for finding the optimized supplier configuration of a task is given in Figure 5.8. In the algorithm, the default configuration is compared with other possible configurations until the one that yields the maximum value of user's satisfaction is found and is returned as result of the algorithm.

```

Optimize_Supplier
Input:
  1. Application a, current Supplier Scurrent
  2. All suppliers Si 1 ≤ i ≤ n for an application a.
Output: Best_Supplier for application a
Procedure:
  1. calculate fcurrent and ccurrent of supplier Scurrent
  2. Preference = fcurrent * ccurrent
  3. Max = Preference
  4. Best_Supplier = Scurrent
  5. for each supplier Si do
  6.   if Scurrent is replaced with Si by dynamic configuration
  7.     then X=1
  8.     else X=0
  9.   endif
  10.  get ci of replacement supplier Si
  11.  h=(ccurrent + ci)/(2*ccurrent)
  12.  User_Satisfaction = Preference * h ^ X.
  13.  if User_Satisfaction > Max then
  14.    Max= User_Satisfaction
  15.    Best_Supplier= Si
  16.  endif
  17.endifor
  18.return Best_Supplier
Endprocedure

```

Figure 5.8 Algorithm for optimized supplier and its QoS configuration.

5.7 Interaction Context, Modality and Media Devices

Interaction context is formed by combining the contexts of the *user*, his *environment*, and his computing *system*. The *user context*, in this work, is a function of user profile (including any handicap) and preferences. A sample user profile, in generic format, is shown in Figure 5.9. The user's special needs determine other affected modalities (i.e. the user is already disqualified from using visual input/output modalities). For example, being mute prevents the user from using vocal input modality.

Identity	Special Needs	Computing Device
username: <username>	Manually-Disabled : <Yes/No>	computing device1: <MAC address 1>
	Mute : <Yes/No>	
password: <password>	Deaf : <Yes/No>	∴
	Unfamiliarity with Braille: <Yes/No>	computing device n : <MAC address n>

Figure 5.9 A sample user profile.

As a function of modality, the user context **UC** can be represented by a single parameter, that of the user's *special needs*. This parameter is a 4-tuple representing additional handicaps, namely the *manual disability*, *muteness*, *deafness*, and *unfamiliarity with Braille*. Each handicap affects user's suitability to adopt certain modalities. The failure of modality, adopted from Eq. (5.6), with respect to **UC** parameters is:

$$\begin{aligned}
 \text{Modality Failure} = & (\text{Manually-Disabled}) \wedge (\text{Mute}) \\
 & \vee ((\text{Manually-Disabled} \vee \text{Unfamiliar with Braille}) \\
 & \wedge \text{Deaf})
 \end{aligned}
 \tag{5.21}$$

The *environment context* **EC** is the assessment of a user's workplace condition. To a blind user, a parameter such as light's brightness has no significance, while others, such as noise level, are significant. In this work, the environment context is based on the following

parameters: (1) the workplace's *noise level* – identifies if it is quiet/acceptable or noisy, and (2) the *environment restriction* – identifies whether a workplace imposes mandatory silence or not. Based on the specified parameters, the environment context, therefore, is formally given by the relationship:

$$EC = (\text{Noise Level}) \wedge (\text{Environment Restriction}) \quad (5.22)$$

Tableau 5.1 shows the affected modality based on environment's context. It also shows the convention table we have adopted for **EC**.

Tableau 5.1 The tabulation for affected modalities by combined noise level and environment restriction.

Environment Context		Inappropriate Modality
Value	Convention	
1	(Environment Restriction = Silence Optional) AND (Noise Level = Noisy)	Vocal input
2	(Environment Restriction = Silence Optional) AND (Noise Level = Acceptable)	
3	(Environment Restriction = Silence Required) AND (Noise Level = Acceptable)	Vocal input, Vocal output (speaker)
4	(Environment Restriction = Silence Required) AND (Noise Level = Noisy)	Vocal input, Vocal output (speaker)

The unit of noise is *decibel* (dB). In our work, *50 dB or less* is considered “*acceptable*” or “*quiet*” while *51 dB or more* is considered “*noisy*”. In our system design, this range can be modified, through user interface, by the end user based on his perception. In general, when the user's workplace is noisy, the effectiveness of vocal input modality is doubtful; hence an alternative modality is necessary.

In an environment where silence is required, sound-producing media (e.g. speaker) needs to be muted or deactivated. For environment noise restriction, we have defined a database of

pre-defined places (e.g. library, park) and their associated noise restrictions (e.g. library: silence required, park: silence optional). User can modify some database records. Also, new ones can be added through the user interface.

In our work, the *system context (SC)* signifies the user's computing device and the available media devices. See Figure 5.10. The computing device (e.g. PC, laptop, PDA, cellular phone) also affects the modality selection. For example, using a PDA or cell phone prevents the user from using tactile input or output modality.

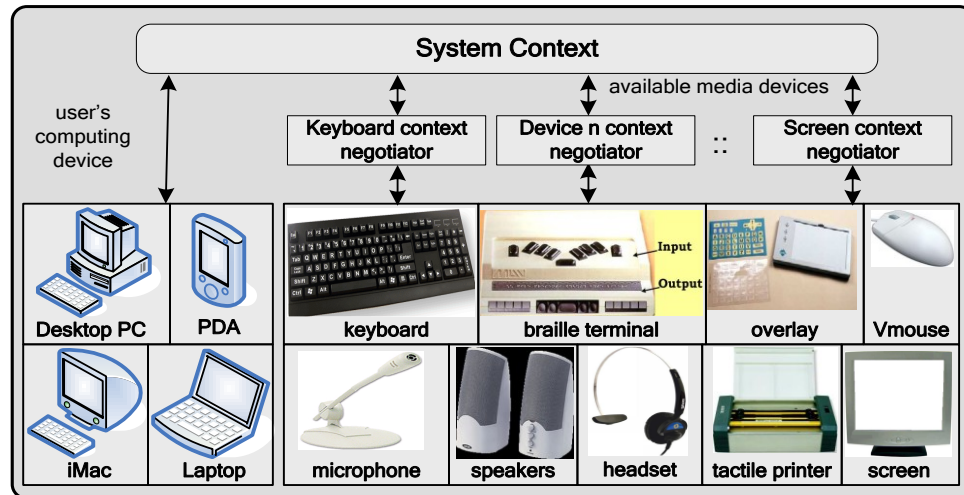


Figure 5.10 The system context as function of user's computing device and available media devices.

Let **SC**, for the purpose of modality selection, be represented by a single parameter, the user's computing device. Let **Computing Device** = {PC, MAC, Laptop, PDA, Cellular phone}. The computing device convention is shown in Tableau 5.2. For example, **SC** = 1 means that the user's computer is either a PC, a laptop or a MAC. Note that when **SC** = 2 (i.e. PDA), **T_{in}** = **Failed** because the computing device has no tactile input device; its **T_{out}** = **Failed** because, in a regular set-up, it is not possible to attach a tactile device (e.g. Braille terminal) onto it.

Tableau 5.2 The convention table of user's computing device and its effect on modality selection.

Convention No.	Computing device	Inappropriate Modality
1	PC / MAC / Laptop	-
2	PDA	T_{in}, T_{out}
3	Cellular Phone	T_{out}

Using ML, the following are the derived rules on modality failures given the parameters of interaction context:

$$V_{in} \text{ Failure} = \text{User} = \text{Mute}) \vee \text{Noise Level} = \text{Noisy}) \vee \text{Environment Restriction} = \text{silence required}) \quad (5.23)$$

$$V_{out} \text{ Failure} = \text{User} = \text{Deaf}) \quad (5.24)$$

$$T_{in} \text{ Failure} = \text{User} = \text{Manually - Disabled}) \vee \text{Computing Device} = \text{'DA}) \quad (5.25)$$

$$T_{out} \text{ Failure} = \text{User} = \text{Manually - Disabled}) \vee \text{(User} = \text{Unfamiliar with Braille}) \vee \text{(Computing Device} = \text{'DA}) \vee \text{(Computing Device} = \text{Cellphone}) \quad (5.26)$$

5.8 Design Specification and Scenario Simulations

Having formulated various ML knowledge to optimize the configuration setting of user's task, this knowledge is then put to test via sample scenarios. The design specification comes along as these scenarios are further explained.

5.8.1 Selection of Modalities

Consider, for example, an interaction context that is composed of the following parameters:

$\mathbf{IC} = (A_1, A_2, A_3, A_4, A_5, A_6, A_7)$ wherein:

- $A_1 = \{\text{true} \mid \text{false}\}$ = if user is manually disabled,
- $A_2 = \{\text{true} \mid \text{false}\}$ = if user is mute,
- $A_3 = \{\text{true} \mid \text{false}\}$ = if user is deaf,
- $A_4 = \{\text{true} \mid \text{false}\}$ = user familiarity with Braille,
- $A_5 = \{\text{quiet} \mid \text{noisy}\}$ = environment's noise level,
- $A_6 = \{\text{silence required} \mid \text{silence optional}\}$ = environment's noise level restriction, and
- $A_7 = \{\text{PC or Laptop or MAC} \mid \text{PDA} \mid \text{Cellphone}\}$ = user's computing device.

The set of possible modalities (i.e. refer to Eq. (5.5)) is given by $\mathbf{M} = \{M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9\}$ wherein $M_1 = \{T_{in}, T_{out}\}$; $M_2 = \{T_{in}, V_{out}\}$; $M_3 = \{V_{in}, V_{out}\}$; $M_4 = \{V_{in}, T_{out}\}$; $M_5 = \{V_{in}, T_{out}, V_{out}\}$; $M_6 = \{T_{in}, T_{out}, V_{out}\}$; $M_7 = \{T_{in}, V_{in}, V_{out}\}$; $M_8 = \{T_{in}, V_{in}, T_{out}\}$; $M_9 = \{T_{in}, V_{in}, T_{out}, V_{out}\}$ (see its derivation from Section 5.5.3). In this example, let us assume the following interaction context: (i) *user context*: blind with no further handicaps, familiar with Braille; hence $A_1 = \text{False}$, $A_2 = \text{False}$, $A_3 = \text{False}$, $A_4 = \text{True}$, (ii) *environment context*: the user is in a classroom, then $A_5 = \text{noisy}$, $A_6 = \text{silence required}$, (iii) *system context*: the user works on a laptop; $A_7 = \text{Laptop}$. The system now finds the modality that suits the given interaction context. The system does so using the principles discussed in Section 5. Let us assume that the computing system's SR contains recorded scenarios as shown in Figure 5.11. The given figure is generated by using WEKA (*Waikato Environment for Knowledge Analysis*) (Witten et 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. In this work, a sample SR is shown using *arff viewer* (i.e. a WEKA tool).

No.	A1	A2	A3	A4	A5	A6	A7	class
1	False	False	False	False	Noisy	Optional	CellPh...	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	True	False	False	False	Quiet	Optional	PC	M3
9	False	True	True	True	Noisy	Optional	Laptop	M1
10	False	False	False	True	Quiet	Optional	CellPh...	M7
11	False	False	True	True	Quiet	Optional	MAC	M8
12	True	False	False	False	Quiet	Optional	PDA	M3
13	False	False	False	True	Quiet	Optional	Laptop	M9
14	False	False	False	True	Noisy	Optional	PC	M6
15	False	False	True	True	Quiet	Optional	PC	M8
16	False	False	False	False	Quiet	Optional	PDA	M3
17	False	False	False	False	Quiet	Optional	MAC	M7
18	False	True	False	False	Quiet	Optional	Laptop	M2
19	False	False	True	True	Noisy	Required	MAC	M1
20	False	False	False	True	Noisy	Optional	Laptop	???

Figure 5.11 A sample of a scenario repository (SR).

As shown in the diagram, there are already 19 scenarios representing the system's acquired knowledge. The 20th scenario represents a new case. Using Equation 5.14, and with reference to the given interaction context and SR, the suitability score of M_j (where $j = 1$ to 9) can be calculated. Let us consider, for instance, the calculations involved with modality M_1 :

$$\text{Suitability_Score}(M_1) = P(A_1 = \text{False} \mid M_1) \times P(A_2 = \text{False} \mid M_1) \times \dots \times P(A_7 = \text{Laptop} \mid M_1) \times P(M_1) = 1 \times 0.67 \times 0 \times \dots \times 3/19 = 0$$

wherein $P(A_1 = \text{False} \mid M_1) = 3/3$, $P(A_2 = \text{False} \mid M_1) = 2/3$, $P(A_3 = \text{False} \mid M_1) = 0/3$, $P(A_4 = \text{True} \mid M_1) = 3/3$, $P(A_5 = \text{Noisy} \mid M_1) = 1/3$, $P(A_6 = \text{silence optional} \mid M_1) = 1/3$, and $P(A_7 = \text{Laptop} \mid M_1) = 1/3$. Also, $P(M_1) = 3/19$.

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

$$\text{Suitability_Score } (M_6) = P(A_1 = \text{False} \mid M_6) \times P(A_2 = \text{False} \mid M_6) \times \dots \times P(A_7 = \text{Laptop} \mid M_6) \times P(M_6) = 1 \times \frac{2}{3} \times 1 \times 1 \times \frac{2}{3} \times \frac{2}{3} \times \frac{1}{3} \times \frac{3}{19} = 0.015671$$

As explained in Section 5.3, $M_6 = \{T_{in}, T_{out}, V_{out}\}$ respects the conditions imposed in Eq. (5.5), hence, it is chosen as the optimal modality for the given IC. This new scenario is added to SR as a newly-acquired knowledge (i.e. scenario #20 in SR).

5.8.2 Selection of Media Devices

As stated in Section 5.7, function f_1 associates selected modalities to available media devices. For simulation purposes, let us assume that currently available media devices are the **MIC**, **Speech Recognition**, **SPK**, **HST**, **Text-to-Speech**, **KB**, and **BRT**. Given this scenario, V_{in} is not part of optimal modalities, then media devices **MIC**, and **Speech Recognition** which support V_{in} are automatically excluded from the set of supporting media devices:

$$f_1 = \{(\cancel{V_{in}}, (\cancel{\text{Mic}}, 1)), (\cancel{V_{in}}, (\cancel{\text{Speech Recognition}}, 1)), (V_{out}, (\text{SPK}, 1)), (V_{out}, (\text{HST}, 2)), (V_{out}, (\text{Text-to-Speech}, 1)), (T_{in}, (\text{KB}, 1)), (T_{in}, (\text{BRT}, 2)), (T_{out}, ((\text{BRT}, 1))\}$$

Since the selection of media device is based on the user preferences, therefore the system automatically activates the media devices having 1 as priority for each modality.

5.8.3 Selection of Application Supplier

Consider for example a user on the go (i.e. in a park) working on a LaTeX document (work1.tex) and another document MathML (work2.xml) as part of his homework. To do so, our user needs instantiation of these files using appropriate suppliers. Finding relevant suppliers can be done using function f_2 which yields the following values:

$$f_2 = \{((.tex, \{\text{KB}, \text{BRT}\}) (\text{MAVIS}, 1)), ((.tex, \{\text{KB}, \text{BRT}\}) (\text{LaBraDoor}, 2)), ((.tex, \{\text{KB}, \text{BRT}\}) (\text{VICKIE}, 3)), ((.tex, \{\text{KB}, \text{SPK} \mid \text{HST}\}), (\text{ASTER}, 1)), ((.xml, \{\text{KB}, \text{BRT}\}), (\text{Lambda}, 1)), (.xml, \{\text{KB}, \text{BRT}\}), (\text{BraMaNet}, 2)), ((.xml, \{\text{KB}, \text{SPK} \mid \text{HST}\}), (\text{AudioMath}, 1)), ((.xml, \{\text{KB}, \text{SPK} \mid \text{HST}\}), (\text{VoiceXML}, 2))\}$$

Formally, $\forall x$: data format d: media devices, $\exists y$: Supplier and p: Priority of type ϵ_1 | $(x, d) \rightarrow (y, p) \in f_2$.

Given that supplier priority is involved in f_2 then the most-preferred supplier is sought. With reference to Eq. (5.19), the numerical values associated with user's preferred suppliers are as follows: (i) priority = 1 (high), user satisfaction = 1, (ii) priority = 2 (medium), user satisfaction = $2/3$, and (iii) priority = 3 (Low), user satisfaction = $1/3$. Now, consider further that the user's preferred supplier, MAVIS, is absent as it is not available in the user's laptop. The method by which the system finds the optimal supplier configuration is shown below:

Case 1: (Lambda, MAVIS) \rightarrow not possible

Case 2: (Lambda, LaBraDoor) \rightarrow alternative 1

Case 3: (Lambda, VICKIE) \rightarrow alternative 2

Then the replacement selection is based on user satisfaction score:

User Satisfaction (Case 2) $= (1 + 1 + 2/3) / 3 = 8/9 = 0.89$

User Satisfaction (Case 3) $= (1 + 1 + 1/3) / 3 = 7/9 = 0.78$

Hence, Case 2 is the preferred alternative. Formally, if f_2 : Document Format, {Media devices} \rightarrow (Supplier, Priority) where Priority: ϵ_1 , then the chosen supplier is given by:

$\exists(\text{doc}, m)$: (Document, Media devices) $\forall y$: Supplier, $\exists p_1$: Priority, $\forall p_2$: Priority | $(\text{doc}, m) \bullet y \rightarrow (y, p_1) \in f_2 \wedge (p_1 < p_2)$.

5.8.4 Optimizing User's Task Configuration

Consider a scenario where all suppliers that convert LaTeX document to its Braille equivalent are available. For example, given the suppliers MAVIS, LaBraDoor, VICKIE, then the corresponding user satisfaction with respect to these suppliers are as follows:

$C_{\text{MAVIS}} = 1.0$, $C_{\text{LaBraDoor}} = 2/3$, $C_{\text{VICKIE}} = 1/3$.

This indicates that the user is most happy with the top-ranked supplier (MAVIS) and least happy with the bottom-ranked supplier (VICKIE). Consider further that these suppliers have $n = 3$ preferred features (i.e. Math Braille code, capability of informing the user of the nature of the next mathematical expression, navigation within the expression). If in MAVIS set-up, the Nemeth Math code is not installed, then the missing feature $m = 1$ and the user satisfaction becomes $f_s = 1 - m/(n + 1) = 1 - 1/4 = 0.75$. This also reduces the user's satisfaction, as given by the relationship $c_{\text{MAVIS}} \times f_{\text{MAVIS}} = (1.0)(0.75) = 0.75$.

Now, consider a case of a dynamic reconfiguration wherein the default supplier is to be replaced by another. Not taking f_s into account yet (assumption: $f_s = 1$), if the current supplier is BraMaNet, then the user's satisfaction is $c_{\text{BraMaNet}} = 2/3 = 0.67$. What would happen if it will be replaced by another supplier through dynamic reconfiguration ($x_{\text{supplier}} = 1.0$)? Using the relationship $h_{\text{supplier}} = (c_{\text{supplier}} + c_{\text{replacement}}) / 2 \times c_{\text{supplier}}$ then the results of possible alternative configurations are as follows:

Replacing BraMaNet (supplier 2):

Case 1: Replacement by MAVIS (supplier 1): $(0.67)(1) \times [(0.67 + 1)/2 \times (0.67)]^1 = 0.835$

Case 2: Replacement by itself (supplier 2): $(0.67)(1) \times [(0.67 + 0.67)/2 \times (0.67)]^1 = 0.67$

Case 3: Replacement by VICKIE (supplier 3): $(0.67)(1) \times [(0.67 + 0.33)/2 \times (0.67)]^1 = 0.50$

Hence, if the reconfiguration aims at satisfying the user, then the second-ranked supplier should be replaced by the top-ranked supplier.

5.8.5 Specification for Detecting Suitability of Modality

*Petri Net*³⁴ is a formal, graphical, executable technique for the specification and analysis of a concurrent, discrete-event dynamic system. Petri nets are used in deterministic and in probabilistic variants; they are a good means to model concurrent or collaborating systems.

34 Petri Net : <http://www.winpesim.de/petrinet>. Consulté le 10/3/2009.

In the specifications in this paper, only a snapshot of one of the many outcomes is presented due to space constraints. We use HPSim³⁵ in simulating Petri Net.

In Figure 5.12, a Petri Net specification is shown with modalities and interaction context. As shown, the combination of interaction context's parameters yields the implementation of some modalities (M_1 up to M_9). The Net illustrates the snapshot simulation of the case cited in Section 5.8.4. As shown, the simulation begins with a token in “*Modality*” place and “*Interaction Context*” place. The firing of the token in Interaction Context yields a specific value for “*User Context*”, “*Environment Context*” and “*System Context based on Computing Device*” places, which is exactly similar to the values of A_1, \dots, A_7 in the previous section. The traversal of the tokens in different places is noted by green colored places. As shown, the result yields modality M_6 being selected as the optimal modality. The Petri Net simulation confirms the result obtained in the previous section. Also, the same case yields a V_{in} failure result (i.e. due to noisy environment).

5.8.6 Simulation Results

Using user's preferences, we have simulated the variations in user's satisfaction as these preferences are modified through dynamic configuration. The results are presented through various graphs in Figure 5.13. The first three graphs deal with application supplier, and the variation of user's satisfaction as additional parameters (supplier features and alternative replacements) are taken into account. The last two graphs deal with QoS dimensions and their variations. In general, user is satisfied if the supplier and its desired features and QoS dimensions are provided. Whenever possible, in a dynamic configuration, the preferred setting is one where the parameters are those of user's top preferences.

35 HPSim : <http://www.winpesim.de>. Consulté le 10/3/2009.

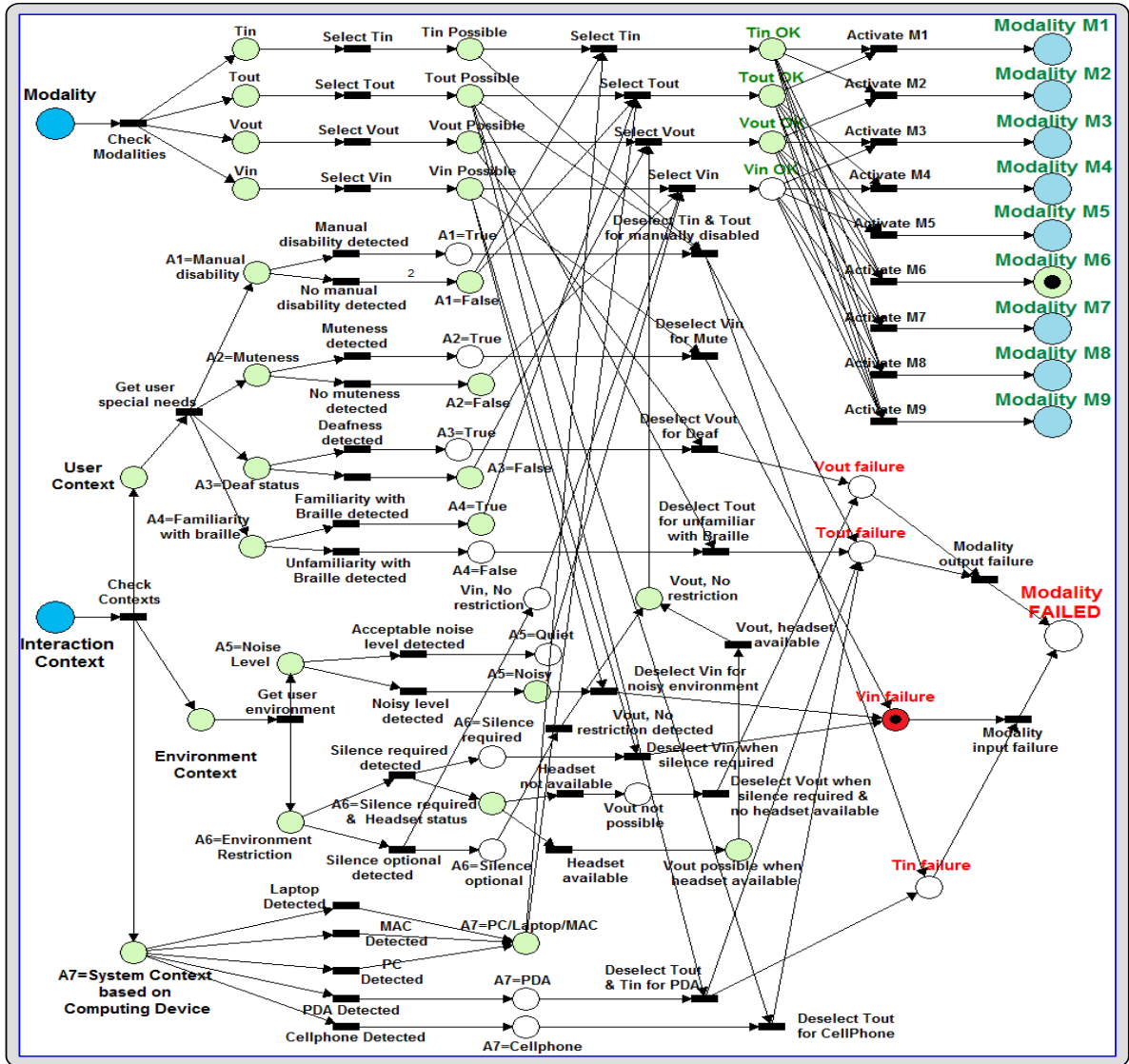


Figure 5.12 A snapshot of the simulated selection of optimal modality based on interaction context.

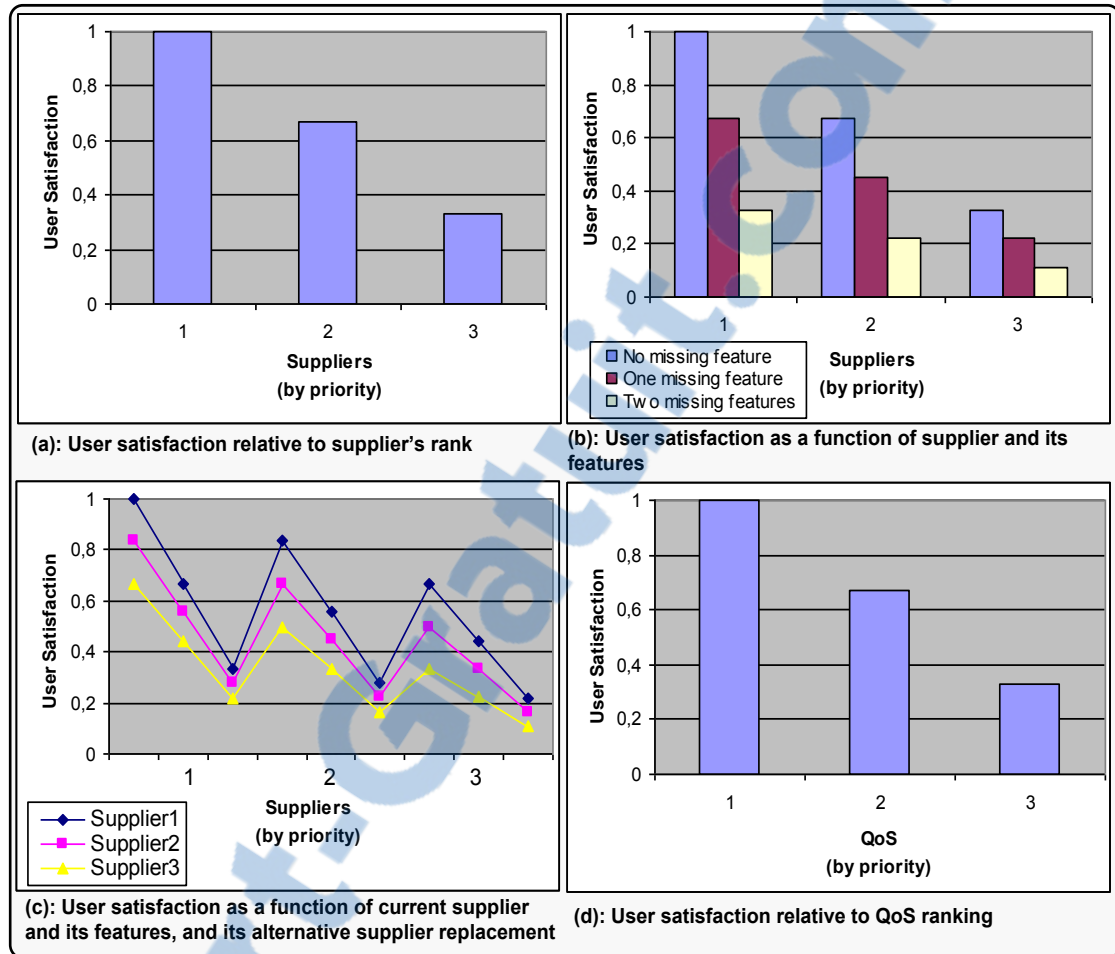


Figure 5.13 Various graphs showing variations of user's satisfaction with respect to its preferred supplier and QoS dimension and their replacements.

5.9 Conclusion

Our investigation on the state of the art systems and solutions for providing visually-impaired users with access to mathematical expressions indicates that none of these systems are pervasive and not a single one adapts its configuration based on the given interaction context. To address these weaknesses, we have designed a multimodal multimedia computing system that would provide mathematical computing to blind users whenever and wherever the users wish. This paper presented the infrastructure design of a middleware that realizes a successful migration and execution of user's task in a pervasive multimodal multimedia computing system, the task being the ubiquitous access to mathematical expressions for

visually-impaired users. Through ML training, we illustrated the acquisition of positive examples to form user's preferred suppliers and QoS dimensions for selected applications. In a rich computing environment, alternative configuration spaces are possible which give the user some choices for configuring the set-up of his application. We have illustrated that configuration could be dynamic or user-invoked, and the consequences, with respect to user's satisfaction, of these possible configurations. Optimization is achieved if the system is able to configure set-up based on user's preferences.

In this work, we have listed modalities suitable to blind users. Given sets of suppliers, modalities, computing devices, and the possible variations of interaction context, we stated conditions in which modality will succeed or fail. Similarly, we showed a scenario wherein even if a specific modality is already deemed possible, still it is conceivable that it would fail if there are not sufficient media devices that would support it or the environment restriction imposes the non-use of the needed media devices. We validated all these affirmations through scenario simulations and formal specifications

5.10 References

- Awde, A., M. D. Hina, C Tadj, Y Bellik et A Ramdane-Cherif. 2006a. « A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users ». In *The First International Conference on Grid and Pervasive Computing (GPC)* (Taichung, Taiwan, 3-5 May 2006). p. 620-633. Lecture Notes in Springer-Verlag.
- Awde, A., M. D. Hina, C. Tadj, A. Ramdane-Cherif et Y. Bellik. 2006b. « Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users ». In *ISIE 2006, IEEE International Symposium on Industrial Electronics* (Montreal, QC, Canada, July 9-13, 2006).
- Batusic, M., K. Miesenberger et B. Stöger. 1998. « Labrador: A Contribution to Making Mathematics Accessible for the Blind ». In *6th International Conference on Computers Helping People with Special Needs (ICCHP)* (Vienna, Austria & Budapest, Hungary, 31 August - 4 September 1998). p. 307-315. Springer Verlag.

- Cahill, H., C. Lineham, J. McCarthy, G. Bormans et J. Engelen. 1995. *Ensuring Usability in MATHS*. Coll. « The European Context for Assistive Technology, Assistive Technology Research Series ». Amsterdam: IOS Press, 66-69 p.
- Commission Evolution du Braille Francais*. 2001. Notation Mathematique Braille, mise à jour de la notation mathematique de 1971.
- Edwards, A. D. N., H. McCartney et F. Fogarolo. 2006. « Lambda: a Multimodal Approach to Making Mathematics Accessible to Blind Students ». In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. Portland, Oregon, USA.
- Edwards, A. D. N., et R. D. Stevens. 1994. « A Multimodal Interface for Blind Mathematics Students ». In *INSERM*. p. 97 - 104. Paris, France.
- Epheser, H., D. Pograniczna et K. Britz. 1992. « Internationale Mathematikschrift fur Blinde ». Marburg (Lahn), Deutsche Blindenstudienanstalt.
- Ferber, J. 1995. *Les systemes multi-agents*. Paris: InterEditions.
- Ferreira, H., et D. Freitas. 2004. « Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project ». In *9th International Conference on Computer Helping People with Special Needs (ICHP)*. p. 678-685. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Ferreira, H., et D. Freitas. 2005. « AudioMath: Towards Automatic Readings of Mathematical Expressions ». In *Human-Computer Interaction International (HCII)*. Las Vegas, Nevada, USA.
- Jacquet, C., Y. Bellik, Y. Bourda et R. Farcy. 2004. « A Context-Aware Locomotion Assistance Device for the Blind ». In *9th International Conference on Computers Helping People with Special Needs (ICHP)*. Paris, France.: Springer Lecture Notes in Computer Science (LNCS).
- Jennings, N. R., et M. J. Wooldridge. 1998. « Applications of Intelligent Agents ». In *Agent Technology: Foundations, Applications, and Markets*, sous la dir. de Jennings, Nicholas R., et Michael J. Wooldridge. p. 3-28. Springer-Verlag: Heidelberg, Germany.
- Kallenberg, O. 2002. *Foundations of Modern Probability*. Springer, 650 p.
- Karshmer, A. I., G. Gupta, S. Geiiger et C. Weaver. 1998. « Reading and Writing Mathematics: the MAVIS Project ». In *Proceedings of the third international ACM conference on Assistive technologies (ASSETS)*. p. 136-143. Marina del Rey, California, USA: ACM SIGCAPH Computers and the Physically Handicapped

- Mitchell, Tom 1997. « Machine Learning ».
- Moço, V., et D. Archambault. 2003. « VICKIE: A Transcription Tool for Mathematical Braille ». In *7th European Conference for the Advancement of Assistive Technology in Europe (AAATE)* (September 2003). Dublin, Ireland: IOS press.
- Nemeth, A. 1972. « The Nemeth Braille Code for Mathematics and Science Notation ». American Printing House for the Blind.
- Raman, T. V. 1994. « Audio System For Technical Readings ». Ph.D thesis, Cornell University.
- Ross, D. A., et A. Lightman. 2005. « Talking Braille: A Wireless Ubiquitous Computing Network for Orientation and Wayfinding ». In *7th International ACM SIGACCESS Conference on Computers and Accessibility*. Baltimore, MD, USA: ACM.
- Satyanarayanan, M. 2001. « Pervasive Computing: Vision and Challenges ». *IEEE Personal Communications*, n° August, p. 10-17.
- Schwebel, F., et R. Goiffon. 2005. « BraMaNet: Quelques règles simples à connaître pour qu'un aveugle puisse lire vos documents mathématiques et vos pages web ». In *Journées nationales Caen* (Caen, France, October 2005).
- Sousa, J. P., V. Poladian, D. Garlan, B. Schmerl et M. Shaw. 2006. « Task-based Adaptation for Ubiquitous Computing ». *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews* vol. 6, n° 3, p. 328-340.
- Stöger, B., K. Miesenberger et M. Batusic. 2004. « Mathematical Working Environment for the Blind Motivation and Basic Ideas ». In *9th International Conference on Computers Helping People with Special Needs (ICCHP)*. p. 656-663. Springer Verlag, Lecture Notes in Computer Science (LNCS).
- Utgoff, P. 1988. « ID5: An incremental ID3 ». In *the fifth International Workshop on Machine Learning*. p. 107-120.
- Weiss, G. 1999. « Multiagent Systems ». MIT-Press.
- Witten, I. H., et E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed.* San Francisco: Morgan Kaufmann, 525 p.

CONCLUSION

Cette thèse s'inscrit dans le cadre d'un projet de recherche portant sur la conception d'un système multimodal permettant l'accès aux expressions mathématiques pour les non-voyants. Nous allons récapituler, de manière synthétique, les contributions apportées par cette thèse, sachant que les publications qui en sont issues sont listées en annexe I.

Contributions

L'utilisation de l'ordinateur et d'équipements informatiques par une personne non-voyante est devenue incontournable puisque ces outils constituent un formidable tremplin, lorsqu'ils sont bien maîtrisés, pour faciliter les apprentissages scolaires, accélérer et amplifier l'accès à la connaissance et aux savoirs. L'accès aux informations numériques a de plus en plus un impact sur la réussite de l'individu au sein de la société (performances scolaires et professionnelles). Ne pas avoir accès à cette technologie risque d'isoler l'individu et de limiter ses ressources et ses ambitions.

Dans cette optique, plusieurs outils ont été développés pour présenter les mathématiques aux non-voyants. Nous avons analysé les techniques utilisées par ces outils et nous avons mis en évidence leurs forces et leurs faiblesses par rapport aux besoins des utilisateurs. Suite à cette analyse, nous avons proposé un système multimodal multi-agents qui exploite les points forts des outils déjà existants et les incorpore dans un système adaptatif qui tient compte de la diversité des utilisateurs, des systèmes, des environnements et de la complexité de l'expression mathématique. Face au caractère dynamique du contexte d'interaction, le système doit être conçu selon une architecture qui s'adapte dynamiquement aux différentes variations.

Le contexte de l'interaction est défini par trois éléments : l'utilisateur, l'environnement et le système. Le contexte de l'utilisateur comporte le profil de l'utilisateur et ses préférences pour les modalités, les médias et les formes de présentations. Le contexte de l'environnement comporte deux paramètres : le niveau de bruit et les restrictions imposées par

l'environnement (ex. le silence est demandé dans une bibliothèque). Le contexte du système implique la machine de l'utilisateur (ex. ordinateur personnel, PDA, MAC, etc.), les médias disponibles (ex. clavier, terminal braille, overlay, etc.) et les formes de présentations (ex. braille linéaire, forme sonore, DotsPlus, EasyMath, etc.).

Nous avons élaboré une méthode basée sur l'arbre syntaxique de l'expression mathématique pour déterminer sa complexité. À la suite des expériences effectuées durant nos travaux, nous avons retenu les trois paramètres suivants : Hauteur (hauteur de l'arbre), le nombre de feuilles terminales (opérandes) et le nombre de sommets non terminaux (opérateurs).

Nous avons aussi illustrée à travers des exemples, la méthode utilisée par l'agent d'apprentissage (MLA) de notre système pour déterminer la meilleure forme qui convient au contexte de l'interaction. En effet, nous avons développé des techniques d'automatisation pour renforcer l'adaptativité. Ces techniques sont résumées par des relations établies entre le contexte et les modalités appropriées et un modèle qui permet de quantifier ces relations. De même, nous avons défini les relations entre les modalités et les formes de présentations. Il s'agit de sélectionner, dans un premier temps, les meilleures modalités appropriées au contexte de l'interaction et en deuxième lieu, les applications et les outils les plus appropriés pour ensuite trouver leur configuration optimale. L'agent MLA sélectionne alors les relations, disposant des meilleurs scores. Cette technique d'apprentissage permet de rendre l'adaptation automatique et de sauvegarder un historique des situations rencontrées, dans une base de connaissance. Chaque élément de cette base représente une situation. Chaque situation est modélisée par : un scénario, avec une pré-condition qui constitue la partie problème de la situation, une post-condition qui constitue la partie solution de la situation, un degré de convenance « le score » et la partie évaluation de la situation qui représente le taux de pertinence de la solution. La pré-condition comporte les informations contextuelles provenant des différents agents tels que l'agent environnement, utilisateur, gestionnaire des médias et d'analyse. La post-condition constitue la présentation appropriée.

Notre système se veut adaptatif dans le sens où il est apte à détecter les modifications du contexte d'interaction et à prendre des décisions en conséquence. L'agent d'apprentissage

permet à notre système de réagir de façon adéquate en tenant compte des modifications affectant dynamiquement le contexte d'interaction.

Notre système permet de présenter des expressions mathématiques en utilisant une ou plusieurs formes simultanément. En exploitant la multimodalité, il permet de combiner différentes formes afin a) de présenter les informations pertinentes à l'utilisateur, b) de diminuer le plus possible la charge cognitive, et c) de rendre ainsi plus simple l'accès aux formules mathématiques pour les personnes aveugles. Nous avons vu que la multimodalité, par la richesse des interactions qu'elle offre, permet d'adapter le système aux différentes situations d'interaction et au profil de l'utilisateur.

Nous avons présenté l'infrastructure d'un intergiciel qui permet la mobilité de l'utilisateur et son activité. Nous avons décrit le modèle qui assure la migration d'une activité d'un environnement à un autre au sein d'un système diffus « pervasive system ». Ce modèle doit aussi vérifier si les changements au niveau du contexte de l'interaction affectent le choix de la modalité appropriée (et par conséquent des médias qui supportent la modalité ainsi que les formes optimales de la présentation). Le système est capable de s'adapter aux changements et de fournir continuellement des services à l'utilisateur non-voyant, sans qu'une intervention humaine ne soit nécessaire. Dans ce sens, nous avons examiné la stabilité de notre système face aux différents changements. Nous proposons une architecture capable de se reconfigurer dynamiquement pour maintenir le système persistant face aux pannes qui pourraient survenir dans les composants ou les médias utilisés. Nous avons validé notre approche par des spécifications formelles et des simulations via JADE et les réseaux de Pétri.

Pour tester notre SMA, nous l'avons implémenté en utilisant JADE qui permet de concevoir et observer les comportements des agents. Cette simulation nous a permis de valider l'architecture de notre SMA ainsi que les comportements de chaque agent. Le langage de communication multi agents (FIPA-ACL) utilisé dans JADE a mis à l'épreuve les interactions (la coopération, la coordination et la négociation) entre les différents agents.

Nous avons également montré comment modéliser, avec les réseaux de Pétri, le mécanisme de la sélection des modalités. Nous avons exposé comment la combinaison des éléments du contexte d'interaction produit plusieurs possibilités de modalités. Nous avons ensuite exposé comment l'agent MLA sélectionne les meilleures formes de présentations en se basant sur les modalités optimales, toujours grâce aux modèles de réseaux de Pétri. Finalement, nous avons montré comment la multimodalité pourrait réussir ou échouer en se basant sur les médias disponibles, les modalités et le contexte de l'environnement.

Perspectives

Les modèles et les paradigmes que nous avons proposés dans ce travail constituent un premier pas pour répondre aux problèmes de la présentation de l'information à un utilisateur déficient visuel. Dans le futur, il convient de tester le système en situation réelle. L'évaluation d'une telle application doit être réalisée sur une variété de médias et de machines avec plusieurs utilisateurs.

Si notre système vise principalement les expressions mathématiques, il pourra être facilement adapté pour accéder aux informations de natures différentes telles que les données textuelles et les graphiques. Une des perspectives serait d'explorer les différents formats de données dont l'utilisateur peut en avoir besoin.

Notre système actuel s'adresse à des utilisateurs ayant une déficience visuelle, il serait très intéressant dans le futur d'employer et d'explorer les potentiels de notre système dans les projets s'adressant aux personnes ayant d'autres types d'handicaps. Nous pensons qu'un système intelligent comme le nôtre est une voie prometteuse pour offrir un accès simple à l'information.

Une perspective serait ainsi de modéliser à partir d'ontologies les informations contextuelles. Nous devons proposer une méthodologie visant à évaluer l'adéquation entre une ontologie et un corpus. Ceci permettrait de choisir une ontologie qui spécifie effectivement la connaissance abordée dans la collection de données considérées. L'ontologie a fait ses

épreuves pour résoudre les problèmes liés à la réutilisabilité et l'interopérabilité (Hernandez, 2005). Nous proposerons alors un modèle de représentation et de gestion des connaissances contextuelles.

Une autre perspective serait d'incorporer d'autres aspects du contexte à notre modèle. Une intégration facilement réalisable serait la prise en compte des préférences de l'utilisateur pour l'accès aux différents formats des données. Ces préférences pourraient être intégrées au profil de l'utilisateur. Un autre aspect serait d'ajouter des ontologies représentant les connaissances a priori de l'utilisateur sur ses activités.

Enfin, notre système se veut évolutif. Il est apte à accueillir de nouvelles technologies et de nouveaux médias. Une perspective à explorer davantage serait de mener des recherches pour le développement de nouveaux médias d'interaction pour les non-voyants pour ne pas se restreindre uniquement aux médias existants.

En résumé, les travaux et les développements envisagés concernent spécifiquement les aspects suivants:

- Réaliser une analyse comparative des performances de différents algorithmes d'apprentissages et des techniques utilisées dans ce travail.
- Mettre à l'épreuve l'architecture intergicielle du système.
- Prendre en compte la gestion de la sécurité et la confidentialité de l'utilisateur.
- Construire des ontologies représentant les connaissances contextuelles en employant le standard « OWL ».
- Définir formellement les rôles de chaque agent pour faciliter l'adaptation de notre système dans un autre domaine.
- Développer un environnement ambient intelligent, pour l'assistance aux personnes ayant un handicap.

ANNEXE I

CONTRIBUTION DÉTAILLÉE

Articles acceptés ou publiés dans une revue avec comité de lecture:

Awde, A., C. Tadj, Y. Bellik, "Infrastructure of an Adaptive Multi-agent System for Presentation of Mathematical Expression to Visually Impaired Users", International Society for Advanced Science and Technology (ISAST) Transactions on Intelligent Systems, to be published in 2009.

Awde, A., M. D. Hina, C. Tadj, Y. Bellik, A. R. Cherif, "An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually- Impaired Users", Journal of Multimedia (JMM), Vol. 4, No. 4, 2009.

Awde, A., M. D. Hina, C. Tadj, Y. Bellik, A. R. Cherif, "Middleware for Ubiquitous Access to Mathematical Expressions for Visually-Impaired Users", Ubiquitous Computing and Communication Journal (UBICC), Vol. 3, No. 5, 2008.

Awde, A., Y. Bellik, C. Tadj, "Complexity of Mathematical Expressions in Adaptive Multimodal Multimedia System Ensuring Access to Mathematics for Visually Impaired Users", International Journal of Computer and Information Science and Engineering, International Journal of Computer and Information Science and Engineering (IJCISE), Vol. 2, No. 2, 2008.

Articles acceptés et publiés dans des congrès avec comité de lecture :

Awde, A., C. Tadj, Y. Bellik, "Un système multi-agent pour la présentation d'expressions mathématiques à des utilisateurs non-voyants", CCGÉI 2008, 21^{ème} Conférence Canadienne de génie électrique et génie informatique, IEEE Canada, Niagara Falls, Ontario, Canada, May 2008.

Awde, A., M. D. Hina, Y. Bellik, A. Ramdane-Cherif, C. Tadj, "Task Migration in a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users", GPC 2007, 2nd International Conference on Grid and Pervasive Computing, Lecture Notes in Springer-Verlag, Paris, France, May 2007.

Awde, A., M. D. Hina, C. Tadj, A. Ramdane-Cherif, Y. Bellik, "Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users", Proceedings - ISIE 2006, IEEE International Symposium on Industrial Electronics, Vol. 4, Montréal, Canada, July 2006, pp. 2834-2839.

Awde, A., M. D. Hina, C. Tadj, A. R. Cherif, Y. Bellik, "A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users", International Conference on Grid and Pervasive Computing, Lecture Notes in Springer-Verlag, Tunghai University, Taiwan, 3-5 May 2006, pp. 620-633.

BIBLIOGRAPHIE

- Alpaydin, E. . 2004. *Introduction to Machine Learning*. Cambridge, Massachusetts: MIT Press.
- Ando, H., Y. Kitahara et N. Hataoka. 1994. « Evaluation of Multimodal Interface Using Spoken Language and Pointing Gesture on Interior Design System ». In *Third International Conference on Spoken Language Processing (ICSLP)* (Yokohama, Japan, September 18-22, 1994). p. 567-570.
- André, E. 2000. « The Generation of Multimedia Presentations ». In *A Handbook of Natural Language Processing*. p. 305-327. Marcel Dekker.
- Archambault, D. 1999. « BrailleSurf : An HTML Browser for Visually Handicapped People ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA.
- Archambault, D., M. Batusic, F. Berger, D. Fitzpatrick, K. Miesenberger, V. Moço et B. Stöger. 2005. « The Universal Maths Conversion Library: An Attempt to Build an Open Software Library to Convert Mathematical Contents in various Formats ». In *International Conference on Universal Access in Human-Computer Interaction (UAHCI)* (July 22-27, 2005, Las Vegas, Nevada, USA).
- Archambault, D., et D. Burger. 2002. « The Vickie Project: Visually Impaired Children Kit for Inclusive Education ». In *8th International Conference on Computers Helping People with Special Needs (ICCHP)* (Linz, Austria, July 15-20, 2002). p. 90-97. Springer Verlag, Lecture Notes in Computer Science (LNCS).
- Awde, A., Y. Bellik et C. Tadj. 2008. « Complexity of Mathematical Expressions in Adaptive Multimodal Multimedia System Ensuring Access to Mathematics for Visually Impaired Users ». *International Journal of Computer and Information Science and Engineering*, vol. 2, n° 2, p. 103-115.
- Awde, A., M. D. Hina, Y. Bellik, A. Ramdane-Cherif et C. Tadj. 2007. « Task Migration in a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users ». In *2nd International Conference on Grid and Pervasive Computing (GPC)* (Paris, France, 2-4 May 2007). p. 459-471. Lecture Notes in Springer-Verlag.
- Awde, A., M. D. Hina, C. Tadj, Y. Bellik et A. Ramdane-Cherif. 2006a. « A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users ». In *The First International Conference on Grid and Pervasive Computing (GPC)* (Taichung, Taiwan, 3-5 May 2006). p. 620-633. Lecture Notes in Springer-Verlag.

- Awde, A., M. D. Hina, C. Tadj, Y. Bellik et A. Ramdane-Cherif. 2009. « An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually- Impaired Users». *Journal of Multimedia (JMM)*, vol. 4, n° 4.
- Awde, A., M. D. Hina, C. Tadj, A. Ramdane-Cherif et Y. Bellik. 2006b. « Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users ». In *ISIE 2006, IEEE International Symposium on Industrial Electronics* (Montreal, QC, Canada, July 9-13, 2006).
- Awde, A., C. Tadj et Y. Bellik. 2008. « Un système multi-agent pour la présentation d'expressions mathématiques à des utilisateurs non-voyants ». In *21ième Conférence Canadienne de génie électrique et génie informatique*. Niagara Falls, Ontario, Canada: IEEE Canada.
- Balvet, A., O. Grisvard et P. Bisson. 2002. « Tutoriel : Open Agent Architecture Développement d'applications de TALN distribuées, multiagents et multiplates-formes ». In *9ième conférence annuelle de Traitement automatique des langues naturelles (TALN)* (Nancy, France, 24-27 juin 2002).
- Bass, L., R. Faneuf, R. Little, N. Mayer, B. Pellegrino, S. Reed, R. Seacord, S. Sheppard et M. Szczur. 1992. « A Metamodel for the Runtime Architecture of an Interactive System ». *SIGCHI Bulletin*, vol. 24, n° 1, p. 32-37.
- Bastide, R., Y. Bellik, J. Bouchet, T. Ganille, L. Nigay et C. Rousseau. 2005. *Rapport de synthèse*. PEA INTUITION, PHASE 3, Réf. CDS/05/010847 I.R.00, Fourniture F24.1, Marché DGA/DSP/STTC 02.34.049.
- Batusic, M., K. Miesenberger et B. Stöger. 1998. « Labrador: A Contribution to Making Mathematics Accessible for the Blind ». In *6th International Conference on Computers Helping People with Special Needs (ICCHP)* (Vienna, Austria & Budapest, Hungary, 31 August - 4 September 1998). p. 307-315. Springer Verlag.
- Batusic, M., K. Miesenberger et B. Stöger. 2003. « Parser for the Marburg Mathematical Braille Notation ». *Universal Access in HCI - Inclusive Design in the Information Society*. p. 1260-1264.
- Bellifemine, F. L., G. Caire et D. Greenwood. 2007. *Developing Multi-Agent Systems with JADE*. Wiley, 300 p.
- Bellik, Y. 1995. « Interfaces multimodales : concepts, modèles et architectures. ». Thèse de doctorat en informatique, Paris, Université de Paris-Sud XI Orsay.
- Bernareggi, C. 2007. « Speech and Braille Tools to Improve Access to University Scientific Courses ». In *Conference & Workshop on Assistive Technologies for People with*

Vision & Hearing Impairments: Assistive Technology for All Ages (CVHI) (Granada, Spain, 28-31 August, 2007). CEUR-WS.org.

- Bernsen, N. O. 1993. « Modality Theory: Supporting Multimodal Interface Design ». In *ERCIM Workshop on Multimodal Human-Computer Interaction*. p. 37-44.
- Berrami, N. 2001. « Étude et définition de mécanismes de conversion transmodaux pour la présentation d'informations ». DEA Sciences Cognitives, Université de Paris-Sud XI (Orsay).
- Biblioteca Italiana per Ciechi "Regina Margherita" O.N.L.U.S.* . 1998. Codice Braille Italiano.
- Blattner, M., D. Sumikawa et R. Greenberg. 1989. « Earcons and Icons: Their Structure and Common Design Principles ». *Human Computer Interaction*, vol. 4, n° 1, p. 11-44.
- Bolt, R. A. 1980. « Put-That-There : Voice and Gesture at the Graphics Interface ». In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques (SIGGRAPH)* (Seattle, Washington, USA, July 14-18, 1980). p. 262-270. ACM Press.
- Bourbakis, N. G., et D. Kaviraki. 2001. « An Intelligent Assistant for Navigation of Visually Impaired People ». In *The 2nd IEEE International Symposium on Bioinformatics and Bioengineering Conference* (Bethesda, MD, USA, 04-06 Nov. 2001). p. 230-235. IEEE Computer Society.
- Bourguet, M. L. 1992. « Conception et réalisation d'une interface de dialogue personne-machine multimodale ». Thèse de doctorat, ICP, INPG, Université Joseph Fourier Grenoble 1.
- Braille Authority of the United Kingdom*. 1987. Braille Mathematics Notation, Mathematics Committee.
- Brewster, S. A. 1998. « Using Nonspeech Sounds to Provide Navigation Cues ». *ACM Transactions on Computer-Human Interaction*, vol. 5, n° 2, p. 224-259.
- Brewster, S. A., P. C. Wright et A. D. N. Edwards. 1993. « An Evaluation of Earcons for Use in Auditory Human-Computer Interfaces ». In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. p. 222-227. ACM Press.
- Buisine, S. 2005. « Conception et évaluation d'agents conversationnels multimodaux bidirectionnels ». Thèse de doctorat, Université Paris 5, 264 p.

- Burger, D., et J. Hadjadj. 1999. « Non Visual Surfing on the Internet: The BrailleSurf Browser ». In *5th European Conference for the Advancement of Assistive Technology in Europe (AAATE)* (Düsseldorf, Germany, November 1-4, 1999).
- Caelen, J. 2000. « 10 ans de recherches... en multimodalité(s) ». In *Colloque sur les Interfaces Multimodales* (Toulouse, France, 09/10 Mai 2000). IRIT.
- Caelen, J., et M.F. Bruandet. 2001. « Interaction multimodale pour la recherche d'information ». In *Environnements évolués et évaluation de l'IHM*, sous la dir. de Kolski, C. p. 175-205. Paris: Hermès éd.
- Cahill, H., C. Lineham, J. McCarthy, G. Bormans et J. Engelen. 1995. *Ensuring Usability in MATHS*. Coll. « The European Context for Assistive Technology, Assistive Technology Research Series ». Amsterdam, Netherlands: IOS Press, 66-69 p.
- Caprotti, O., et D. Carlisle. 1999. « OpenMath and MathML: Semantic Markup for Mathematics ». *Crossroads, the ACM Student Magazine* vol. 6, n° 2, p. 11-14.
- Casson, A. 2005. « Analyse des besoins pour un logiciel d'aide au sein du projet MICOLE ». DESS, Université Paris 8, Vincennes - Saint-Denis.
- Catinis, L. 1998. « Étude de l'usage de la parole dans les interfaces multimodales ». Thèse de PhD, Grenoble, Institut National Polytechnique de Grenoble.
- Chen, G., et D. Kotz. 2000. *A Survey of Context-Aware Mobile Computing Research*. TR2000-381. Department of Computer Science, Dartmouth College.
- Cheyner, A., et L. Julia. 1995. « Multimodal maps: An Agent-based Approach ». In *International Conference on Cooperative Multimodal Communication (CMC)* (Eindhoven, The Netherlands, May 24-26, 1995). p. 103-113.
- Cohen, P. R., A. Cheyer, M. Wang et S. C. Baeg. 1994. « An Open Agent Architecture ». In *Proceedings of the AAI Spring Symposium on Software Agents*. p. 1-8. Menlo Park, CA, USA: AAI Press.
- Cohen, P. R., M. Johnston, D. McGee, S. L. Oviatt, J. Pittman, I. Smith, L. Chen et J. Clow. 1997. « QuickSet: Multimodal Interaction for Distributed Applications ». In *Proceedings of the Fifth ACM International Conference on Multimedia* (Seattle, Washington, USA, November 09-13, 1997). p. 31-40. Seattle, WA, USA: ACM Press.
- Cohen, P.R., M. Dalrymple, D.B. Moran, F.C.N. Pereira, J.W. Sullivan, R.A. Gargan Jr, J.L. Schlossberg et S.W Tyler. 1989. « Synergistic Use of Direct Manipulation and Natural Language ». In *Human Factors in Computing Systems Conference* (Austin, Texas, USA, April 30 - Mai 4, 1989). p. 227-234. ACM Press.

- Commission Evolution du Braille Francais*. 2001. Notation Mathematique Braille, mise à jour de la notation mathematique de 1971.
- Cormen, T. H., C. E. Leiserson, R. L. Rivest et C. Stein. 2001. *Introduction to Algorithms*, Second. MIT Press and McGraw-Hill.
- Coutaz, J. 1987. « PAC, an Object-Oriented Model for Dialog Design. ». In *Proceedings of 2nd International Conference on Human-Computer Interaction (INTERACT'87)* (September 1-4, 1987, Stuttgart, Germany). p. 431-436.
- Coutaz, J., et J. Caelen. 1991. « A taxonomy for Multimedia and Multi-modal User Interfaces ». In *Proceedings of the ERCIM Workshop on User Interfaces and Multimedia* (Lisbon, Portugal, November 14-15, 1991). p. 143-147.
- Coutaz, J., L. Nigay, D. Salber, A.E. Blandford, J. May et R.M.Y. Young. 1995. « Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The CARE Properties ». In *Proceedings of Interact'95, Human-Computer Interaction* (Lillehammer, Norway, June 25-29, 1995). p. 115-120.
- Crombie, D., R. Lenoir, N. McKenzie et A. Barker. 2004. « Math2Braille: Opening Access to Mathematics ». In *9th International Conference on Computers Helping People with Special Needs, (ICCHP)*. p. 670-677. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Demeure, A., et G. Calvary. 2006. « Le Modèle d'Evolution en Plasticité des Interfaces: Apport des Graphes Conceptuels ». In *15ème Conférence francophone sur l'Interaction Homme-Machine (IHM 2003)*. Caen, France: ACM Press.
- Denda, A., T. Itoh et S. Nakagawa. 1997. « Evaluation of Spoken Dialogue System for a Sightseeing Guidance with Multi-modal Interface ». In *15th international Joint Conference on Artificial Intelligence (IJCAI-97) Workshop on Intelligent Multimodal Systems* (Nagoya, Japan, August 23-24, 1997).
- Dix, A. J. 1996. « Closing the Loop: Modelling Action, Perception and Information ». In *Proceedings of the workshop on Advanced visual interfaces* (Gubbio, Italy, May 27-29, 1996). p. 20-28. ACM Press.
- Djenidi, H. 2007. « Architectures logicielles dynamiques dédiées aux applications multimodales ». Thèse de doctorat en génie, Montréal, École de technologie supérieure, 344 p.
- Djenidi, H., A. Ramdane-Cherif, C Tadj et N. Levy. 2003. « Generic Multimedia Multimodal Agents Paradigms and their Dynamic reconfiguration at the Architecture Level.

Architecture for Software Environment ». *Eurasip : European journal of Applied Signal Processing- Special Issue on Multimedia Applications*.

Djennane, S. 2003. « 3D-Audio News Presentation Modeling ». In *Lecture Notes in Computer Science (LNCS)*. Vol. 4556, p. 280-286.

Dragicevic, P. 2004. « Un modèle d'interaction en entrée pour des systèmes interactifs multi-dispositifs hautement configurables ». Thèse de doctorat, Université de Nantes

Duncan, L., W. Brown, C. Esposito, H. Holmback et P. Xue. 1999. *Enhancing Virtual Maintenance Environments with Speech Understanding*. Technical Report TECHNET-9903. Boeing Mathematics and Computing Technology.

Edwards, A. D. N., H. McCartney et F. Fogarolo. 2006. « Lambda: a Multimodal Approach to Making Mathematics Accessible to Blind Students ». In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. Portland, Oregon, USA.

Edwards, A. D. N., et R. D. Stevens. 1994. « A Multimodal Interface for Blind Mathematics Students ». In *INSERM* (1994). p. 97-104. Paris, France.

Epheser, H., D. Pograniczna et K. Britz. 1992. « Internationale Mathematikschrift fur Blinde ». Marburg (Lahn), Deutsche Blindenstudienanstalt.

Favre, S. 2003. « Accès aux formules mathématiques par des personnes non voyantes ». DEA, Université Paris Sud 11.

Ferber, J. 1995. *Les systèmes multi-agents*. Paris: InterEditions.

Ferreira, H., et D. Freitas. 2004. « Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project ». In *9th International Conference on Computer Helping People with Special Needs (ICCHP)*. p. 678-685. Paris, France: Springer Lecture Notes in Computer Science (LNCS).

Ferreira, H., et D. Freitas. 2005. « AudioMath: Towards Automatic Readings of Mathematical Expressions ». In *Human-Computer Interaction International (HCII)*. Las Vegas, Nevada, USA.

Frohlich, D. M. 1991. « The Design Space of Interfaces ». In *Multimedia principles, systems and applications*. p. 53-69. Springer Verlag éditions.

Fukuda, R., N. Ohtake et M. Suzuki. 2000. « Optical Recognition and Braille Transcription of Mathematical Documents ». In *7th International Conference on Computer Helping People with Special Needs (ICCHP)*. p. 711-718. Karlsruhe, Germany: Springer Lecture Notes in Computer Science (LNCS).

- Gardner, J. A. 1993. « Dotsplus-Better than Braille? ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA.
- Gardner, J. A. 1998. « Random Access: The DotsPlus Tactile Font Set ». *Journal of Visual Impairment and Blindness*, vol. 92, n° 12 (December 1998), p. 836-840.
- Gardner, J. A., R. Stewart, J. Francioni et A. Smith. 2002. « Tiger, Agc, and Win-Triangle, Removing the Barrier to Sem Education ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA.
- Gardner, J., L. Ungier et J. Boyer. 2006. « Braille Math Made Easy with the Tiger Formatter ». In *10th International Conference on Computers Helping People with Special Needs (ICCHP)* (Linz, Austria, July 11-13, 2006). p. 1215-1222. Springer Verlag, Lecture Notes in Computer Science (LNCS).
- Garlini, P., et F. Fogarolo. 2003. *LAMBDA: Linear Access to Mathematics for Braille Device and Audio Synthesis - Analysis of the User's Needs*. Padova, Italy: University of Padova.
- Gillan, D. J., P. Barraza, A. I. Karshmer et S. Pazuchanics. 2004. « Cognitive Analysis of Equation Reading: Application to the Development of the Math Genie ». In *9th International Conference on Computer Helping People with Special Needs (ICCHP)*. p. 630-637. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Gorny, P. 2000. « Typographic semantics of Webpages Accessible for Visual Impaired Users, Mapping Layout and Interaction Objects to an Auditory Interaction Space ». In *Proceedings of 7th International Conference on Computer Helping with Special Needs* (Karlsruhe, Germany, 17-21 July 2000). p. 17-21.
- Hernandez, N. 2005. « Ontologies de domaine pour la modélisation du contexte en recherche d'information ». Thèse de doctorat, Université Paul Sabatier de Toulouse 248 p.
- Horchani, M. 2007. « Vers une communication humain-machine naturelle : stratégies de dialogue et de présentation multimodales ». Thèse de doctorat en informatique, Université Joseph Fourier-Grenoble 1.
- Jacquet, C. 2006. « Présentation opportuniste et multimodale d'informations dans le cadre de l'intelligence ambiante ». Thèse de doctorat en informatique, Paris, Université de Paris-Sud XI (Orsay).
- Jacquet, C., Y. Bellik, Y. Bourda et R. Farcy. 2004. « A Context-Aware Locomotion Assistance Device for the Blind ». In *9th International Conference on Computers Helping People with Special Needs (ICCHP)*. Paris, France.: Springer Lecture Notes in Computer Science (LNCS).

- Jennings, Nicholas R., et Michael J. Wooldridge. 1998. « Applications of Intelligent Agents ». In *Agent Technology: Foundations, Applications, and Markets*, sous la dir. de Jennings, Nicholas R., et Michael J. Wooldridge. p. 3-28. Springer-Verlag: Heidelberg, Germany.
- Jürgensen, H., et C. Power. 2005. « Information Access for the Blind - Graphics, Modes, Interaction ». In *Proceedings of GOTH'05 Guidelines on Tactile and Haptic Interactions* (Saskatoon, Saskatchewan, Canada, October 24-26, 2005). USERLab, University of Saskatchewan.
- Julia, L. 1995. « Interface homme-machine multimodale pour la conception et l'édition de documents ». Thèse de doctorat, École nationale supérieure des télécommunications, Paris, FRANCE.
- Kallenberg, O. 2002. *Foundations of Modern Probability*. Springer, 650 p.
- Karshmer, A., G. Gupta et E. Pontelli. 2008. *Mathematics and Accessibility: a Survey*.
- Karshmer, A. I., C. Bledsoe et P. B. Stanley. 2004. « The Architecture of a Comprehensive Equation Browser for the Print Impaired ». In *9th International Conference on Computer Helping People with Special Needs (ICCHP)*. p. 614-619. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Karshmer, A. I., G. Gupta, S. Geiiger et C. Weaver. 1998. « Reading and Writing Mathematics: the MAVIS Project ». In *Proceedings of the third international ACM conference on Assistive technologies (ASSETS)*. p. 136-143. Marina del Rey, California, USA: ACM SIGCAPH Computers and the Physically Handicapped
- Karshmer, A. I., G. Gupta, E. Pontelli, K. Miesenberger, N. Ammalai, D. Gopal, M. Batusic, B. Stger, B. Palmer et H. F. Guo. 2004. « UMA: a System for Universal Mathematics Accessibility ». In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility*. p. 55-62. Atlanta, GA, USA: ACM Press.
- Knuth, D. 1986. *TeX: The Program* Vol. B. Coll. « Computers & Typesetting ». Addison-Wesley Professional,, 624 p.
- Komada, T., K. Yamaguchi, F. Kawane et M. Suzuki. 2006. « New Environment for Visually Disabled Students to Access Scientific Information by Combining Speech Interface and Tactile Graphics ». In *10th International Conference on Computers Helping People with Special Needs (ICCHP)* (Linz, Austria, July 11-13, 2006). p. 1183-1190. Springer Verlag, Lecture Notes in Computer Science (LNCS).

- Koons, D. B., et C. J. Sparrell. 1994. « ICONIC: Speech and Depictive Gestures at the Human-Machine Interface ». In *Conference on Human Factors in Computing Systems* (Boston, Massachusetts, USA, April 24-28, 1994). p. 453-454. ACM Press.
- Kramer, G. 1994. *Auditory Display: Sonification, Audification, and Auditory Interfaces*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Kramer, J., et J. Magee. 1998. « Analysing Dynamic Change in Software Architectures: A Case Study ». In *Proceedings of the International Conference on Configurable Distributed Systems* Washington, DC, USA: IEEE Computer Society.
- Krasner, G. E., et S. T. Pope. 1988. « A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80 ». *Joop, Aug/Sep 1988*.
- Kumar, S., et P. R. Cohen. 2000. « Towards a Fault-Tolerant Multi-Agent System Architecture ». In *The fourth International Conference on Autonomous Agents* (Barcelona, Spain, June, 3-7, 2000). p. 459-466. ACM Press.
- Lamport, L. 1994. *LaTeX: A Document Preparation System*, 2nd Edition. Addison-Wesley Professional.
- Lécuyer, A., P. Mobuchon, C. Mégard, J. Perret, C. Andriot et J. Colinot. 2003. « HOMERE: a Multimodal System for Visually Impaired People to Explore Virtual Environments ». In *Proceedings of the IEEE Virtual Reality* (Washington, USA, 22-26 March 2003), sous la dir. de Society, IEEE Computer. p. 251-258. Washington, USA: IEEE Computer Society.
- Lightfoot, D. 2001. *Formal Specification Using Z*, 2nd. McMillan Press.
- Martin, J. C. 1996. « Types et buts de coopération entre modalités dans les interfaces multimodales ». *Techniques et Sciences Informatiques*, vol. 15.
- Martin, J. C., R. Veldman et D. Béroule. 1995. « Towards Adequate Representation Technologies for Multimodal Interfaces ». In *International Conference on Cooperative Multimodal Communication* (Eindhoven, The Netherlands, May 24-26, 1995). p. 207-223.
- Martin, J. C. 1995. « Coopérations entre modalités et liage par synchronie dans les interfaces multimodales ». Thèse de doctorat, Université Paris XI (Orsay).
- McCowan, I., D. Moore, J. Dines, D. Gatica-Perez, M. Flynn, P. Wellner et H. Bourlard. 2005. *On the Use of Information Retrieval Measures for Speech Recognition Evaluation*. Coll. « IDIAP Research Report ».

- McGookin, D. K., et S. A. Brewster. 2006. « MultiVis: Improving Access to Visualisations for Visually Impaired People ». In *Vol II Proceedings of ACM CHI 2006*. p. 267-270. Montreal, Canada: ACM Press.
- Mitchell, Tom 1997. « Machine Learning ».
- Moço, V., et D. Archambault. 2003a. « VICKIE: A Transcription Tool for Mathematical Braille ». In *7th European Conference for the Advancement of Assistive Technology in Europe (AAATE)* (September 2003). Dublin, Ireland: IOS press.
- Moço, V., et D. Archambault. 2004. « Automatic Conversions of Mathematical Braille: A Survey of Main Difficulties in Different Languages ». In *9th International Conference on Computers Helping People with Special Needs (ICCHP)*. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Moço, V., et D. Archambault. 2003b. « Automatic Translator for Mathematical Braille ». *Universal Access in HCI, LEA*, p. 1335-1339.
- Moran, D., A. Cheyer, L. Julia, D. Martin et S. Park. 1997. « Multimodal User Interfaces in the Open Agent Architecture ». In *International Conference on Intelligent User Interfaces (IUI'97)* (Orlando, Florida, USA, January 6-9, 1997). Orlando, FL, USA: ACM.
- Mynatt, E. D. 1994. « Designing with Auditory Icons: How Well Do We Identify Auditory Cues? ». In *Conference Companion on Human Factors in Computing Systems* (Boston, Massachusetts, USA, 24-28 April 1994). p. 269-270. ACM Press.
- Nemec, V., A. Sporka et P. Slavik. 2004. « Haptic and Spatial Audio Based Navigation of Visually Impaired Users in Virtual Environment Using Low Cost Devices ». In *User-Centered Interaction Paradigms for Universal Access in the Information Society*. p. 28-29. Vienna, Austria: Springer.
- Nemeth, A. 1972. « The Nemeth Braille Code for Mathematics and Science Notation ». American Printing House for the Blind.
- Nigay, L. 1994. « Conception et modélisation logicielles des systèmes interactifs ». Thèse de doctorat, Université Joseph Fourier Grenoble 1.
- Nigay, L., et J. Coutaz. 1993. « A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion ». In *Conference on Human Factors in Computing Systems, Voices and Faces* (Amsterdam, The Netherlands, April 24-29, 1993). p. 172-178. ACM Press.

- Nigay, L., J. Coutaz et D. Salber. 1993. « MATIS : a Multimodal Airline Travel Information System ». *SM/WP10, System Modelling, Working Paper 10, The Amodeus Project, Esprit Basic Research Action 7040*.
- Nigay, L., et J. Coutaz. 1995. « A Generic Platform for Addressing the Multimodal Challenge ». In *International Conference on Computer-Human Interaction*. p. 98-105. Denver (Colorado): ACM.
- Oviatt, S. L. 2000. « Multimodal System Processing in Mobile Environments ». In *Proceedings of the 13th Annual ACM Symposium on User Interface Software Technology (UIST'2000)*. CM Press. p. 21-30. ACM Press.
- Oviatt, S.L. 1999. « Mutual Disambiguation of Recognition Errors in a Multimodal Architecture ». In *Proceedings of Conference on Human Factors in Computing Systems (CHI'99)*. p. 576-583.
- Pfaff, G. E. 1985. *User Interface Management Systems*. NJ, USA: Springer-Verlag New York, Inc.
- Pfaff, G., et P. J. W. Ten Hagen. 1993. « Seeheim workshop on User Interface Management Systems ». *Springer Verlag* (Berlin).
- Podevin, A. 2002. « Accès aux formules mathématiques par des personnes non voyantes : étude et définition d'une méthode adaptée ». DEA, Université de CAEN.
- Preddy, M., J. Gardner, S. Sahyun et D. Skrivanek. 1997. « Dotsplus: How-to Make Tactile Figures and Tactile Formatted Math ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA.
- Raman, T. V. 1994. « Audio System For Technical Readings ». Ph.D. thesis, Cornell University.
- Raman, T. V. 1998. *Audio System for Technical Readings*, 1410, 1st edition. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 137 p.
- Reddy, H., N. Annamalai et G. Gupta. 2004. « Listener-Controlled Dynamic Navigation of VoiceXML Documents ». In *9th International Conference on Computer Helping People with Special Needs (ICCHP)*. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Reithinger, N., J. Alexandersson, T. Becker, A. Blocher, R. Engel, M. Löckelt, J. Müller, N. Pflieger, P. Poller, M. Streit et V. Tschernomas. 2003. « SmartKom: Adaptive and Flexible Multimodal Access to Multiple Applications ». In *Proceedings of the 5th international conference on Multimodal interfaces (ICMI '03)* (Vancouver, British Columbia, Canada, November 05-07, 2003). p. 101-108. ACM Press.

- Roberts, J., et T. Ertl. 2004. « Tactile Access to Scalable Vector Graphics for People with Visual Impairment ». In *3rd Annual Conference on Scalable Vector Graphics* (Tokyo, Japan, Sept 7-10, 2004).
- Ross, D., et A. Lightman. 2005. « Talking Braille: a Wireless Ubiquitous Computing Network for Orientation and Wayfinding ». In *7th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)* (Baltimore, MD, USA, 9-12 Oct 2005). p. 98-105. ACM.
- Rousseau, C. 2006. « Présentation multimodale et contextuelle de l'information ». Thèse de doctorat en informatique, Paris, Université de Paris-Sud XI Orsay.
- Rousseau, C., Y. Bellik et F. Vernier. 2005. « WWHT: Un modèle conceptuel pour la présentation multimodale d'information ». In *Proceedings of 17th French-speaking conference on Human Computer Interaction (IHM)* (Toulouse, France, septembre 2005). p. 27-30.
- Rousseau, C., Y. Bellik, F. Vernier et D. Bazalgette. 2006. « A Framework for the Intelligent Multimodal Presentation of Information ». *Signal Processing Journal, Special issue on Multimodal Interfaces*.
- Satyanarayanan, M. 2001. « Pervasive Computing: Vision and Challenges ». *IEEE Personal Communications*, n° August, p. 10-17.
- Schilit, B., N. Adams et R. Want. 1994. « Context-Aware Computing Applications ». In *IEEE Workshop on Mobile Computing Systems and Applications*. p. 85-90. Santa Cruz, CA, USA.
- Schwebel, F. 2000. *BraMaNet: logiciel de traduction des mathématiques en Braille from <http://handy.univ-lyon1.fr/projets/bramanet/>*.
- Schwebel, F., et R. Goiffon. 2005. « BraMaNet: Quelques règles simples à connaître pour qu'un aveugle puisse lire vos documents mathématiques et vos pages web ». In *Journées nationales Caen* (October 2005). Caen, France.
- Schweikhardt, W. 1987. *Stuttgarter Mathematikschrift für Blinde*. Report Nr.3/87. Institut für Informatik, Universität Stuttgart.
- Schweikhardt, W. 1998. « 8-Dot-Braille for Writing, Reading, and Printing Texts which Include Mathematical Characters ». In *6th International Conference on Computers Helping People with Special Needs (ICCHP'98)-Computers and Assistive Technology* (Vienna, Austria & Budapest, Hungary, 31 August - 4 September 1998). p. 324-333. Springer Verlag.

- Schweikhardt, W., C. Bernareggi, N. Jessel, B. Encelle et M. Gut. 2006. « LAMBDA: a European System to Access Mathematics with Braille and Audio Synthesis ». In *10th International Conference on Computers Helping People with Special Needs (ICCHP)* (Linz, Austria, July 11-13, 2006). p. 1223-1230. Springer Verlag, Lecture Notes in Computer Science (LNCS).
- Serrano, M., L. Nigay, R. Demumieux, J. Descos et P. Losquin. 2006. « Multimodal Interaction on Mobile Phones: Development and Evaluation using ACICARE ». In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '06)*. p. 129-136. Helsinki, Finland: ACM Press.
- Sheehan, K. 2005. « Your Day Just Got Easier with Overlay Maker 3 ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA.
- Sjöström, C. 2001. « Using Haptics in Computer Interfaces for Blind People ». In *Conference on Human Factors in Computing Systems (CHI)* (Seattle, Washington, 31 March-5 April 2001), 245-246. Seattle, Washington: ACM.
- Soiffer, N. 2005. « Advances in Accessible Web-based Mathematics ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA.
- Soiffer, N. 2007. « MathPlayer v2.1: Web-based Math Accessibility ». In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*. p. 257-258. Tempe, Arizona, USA: ACM.
- Sousa, J. P., V. Poladian, D. Garlan, B. Schmerl et M. Shaw. 2006. « Task-based Adaptation for Ubiquitous Computing ». *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 6, n° 3, p. 328-340.
- Stevens, R. 1996. « Principles for the Design of Auditory Interfaces to Present Complex Information to Blind People ». Ph.D. Thesis, University of York.
- Stevens, R. D., et A. D. Edwards. 1994a. « MathTalk: Usable Access to Mathematics ». *Information Technology and Disability*, vol. 1, n° 4.
- Stevens, R. D., et A. D. N. Edwards. 1994b. « Mathtalk: The Design of an Interface for Reading Algebra Using Speech ». In *4th International Conference on Computers for Handicapped Persons (ICCHP)* (Vienna, Austria, September 14-16, 1994). p. 313-320. Springer-Verlag, Lecture Notes in Computer Science.
- Stevens, R. D., A. D. N. Edwards et P. A. Harling. 1997. « Access to Mathematics for Visually Disabled Students Through Multi-modal Interaction ». *Human Computer Interaction (Special issue on Multimodal Interfaces)*, vol. 1&2, n° 12, p. 47-92.

- Stöger, B., K. Miesenberger et M. Batusic. 2004. « Mathematical Working Environment for the Blind Motivation and Basic Ideas ». In *9th International Conference on Computers Helping People with Special Needs (ICCHP)*. p. 656-663. Springer Verlag, Lecture Notes in Computer Science (LNCS).
- Suzuki, M., T. Kanahori, N. Ohtake et K. Yamaguchi. 2004. « An Integrated OCR Software for mathematical Documents and Its Output with Accessibility ». In *9th International Conference on Computer Helping People with Special Needs (ICCHP)*. p. 648-655. Paris, France: Springer Lecture Notes in Computer Science (LNCS).
- Thompson, D. M. 2005. « LaTeX2Tri: Physics and Mathematics for the Blind or Visually Impaired ». In *International Conference on Technology and Persons with Disabilities (CSUN)*. Los Angeles, CA, USA.
- Thorisson, K., D. B. Koons et R. A. Bolt. 1992. « Multi-modal Natural Dialogue ». In *Conference on Human Factors in Computing Systems* p. 653-654. Monterey, California, USA: ACM Press.
- Utgoff, P. 1988. « ID5: An incremental ID3 ». In *the fifth International Workshop on Machine Learning*. p. 107-120.
- Van Belle, W. , J. Fabry, K. Verelst et T. D'Hondt. 2001. « Experiences in Mobile Computing: The CBorg Mobile Multi-Agent System ». In *Conference on Technology of Object Oriented Languages and Systems, Eastern Europe (TOOLSEE 2001): Components for Mobile Computing (Zurich Switzerland; March 2001)*. .
- Vani, K., M. Krupa Devi, Lavanya Dhanapalan et K. Prabhu. 2005. « Dynamic e-Braille Map for the Blind ». In *Conference Proceedings of Map Asia 2005 (Jakarta, Indonesia, August 22-25, 2005)*.
- Vo, M.T., et A. Waibel. 1993. « Multi-modal Human-Computer Interface: Combination of Gesture and Speech Recognition ». In *Conference on Human Factors in Computing Systems (Amsterdam, The Netherlands, April 24-29, 1993)*. p. 69-70. ACM Press.
- Walsh, P., et J. Gardner. 2001. « TIGER, a New Age of Tactile Text and Graphics ». In *The 16th Annual International Technology and Persons with Disabilities Conference (CSUN) Los Angeles, CA, USA*.
- Weiss, G. 1999. « Multiagent systems ». MIT-Press.
- Witten, I. H., et E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 525 p.
- Wooldridge, M. 2002. *An Introduction to Multiagent Systems*. Chichester, England: Wiley.

- Xydas, G., V. Argyropoulos, T. Karakosta et G. Kouroupetroglou. 2005. « An Experimental Approach in Recognizing Synthesized Auditory Components in a Non-Visual Interaction with Documents ». In *Proceedings of HCI International 2005 - The 11th International Conference on Human-Computer Interaction* (Las Vegas, Nevada, USA, 22-27 July 2005). Vol. 3, p. 411-420.
- Xydas, G., D. Spiliotopoulos et G. Kouroupetroglou. 2003. « Modeling Emphatic Events from Non-Speech Aware Documents in Speech Based User Interfaces ». In *Proceedings of HCI International 2003 - The 10th International Conference on Human-Computer Interaction* (Crete, Greece, 22-27 June 2003). p. 806-810.
- Zanello, M. L. 1997. « L'utilisateur et l'interface multimodale : contribution à la connaissance sur son utilisation et sa gestion ». Thèse de doctorat, Université Joseph Fourier-Grenoble.