Modeling and Simulation of a Pervasive Multimodal Multimedia Computing System for Visually-Impaired Users

[1]Ali Awdé, [2]Moeiz Miraoui, [3]Chakib Tadj, [4]Bellik Yacine
[1,2,3]*Université du Québec, École de technologie supérieure, Montréal, Québec, Canada,*
[4]*LIMSI-CNRS, Université de Paris-Sud, B.P. 133, 91403 Orsay, France*
doi: 10.4156/ijact.vol2.issue3.1

Abstract

Using multimodality in a computing system is advantageous in the sense that it makes computing more accessible to a wide range of users including those with impairments. Our work is aimed at making informatics accessible to the visually impaired, specially the access to mathematical expressions. The selection of modalities, media, and presentation formats depends on the interaction context (i.e. combined contexts of the user, his environment and his computing system) and nature of the expression. The adaptation of a computing system to the needs of a mobile user is essential in order that the user could continue working on his task at anytime anywhere, thereby increasing his productivity. The system design is intended to be pervasive, fault-tolerant and having a great self-adaptation capability under varying conditions (e.g. missing or defective components, change in computing resources). This paper is aimed at presenting the modeling and the simulation of our system through case studies and formal specification using Petri nets. Also we propose ontology for context modeling. A simulation of the system's behavior, along with some test case scenarios, is also presented in this paper.

Keywords*: Multimodality, Visually impaired users, Contextual information*

1. Introduction

Our multimodal multimedia (MM) computing system is aimed at presenting mathematical expressions to visually impaired users. Providing visually-impaired users with ubiquitous access to mathematical expressions is a challenging task, due to the following reasons: First, unlike text, the visual mathematical representation is bi-dimensional. Second, the conversion of a mathematical expression to a non-visual representation (e.g. speech) requires supplementary information to denote some components to allow the visually impaired users to read the expressions easily and avoid any ambiguity. Third, the vocabulary terms used by sighted people are quite large compared to the amount of data accessible by blind people. For example, traditional Braille[1] utilizes 6-dot character, which allows 64 possible characters. This number of symbols is, however, not enough to represent all frequently used mathematical symbols. In addition the large number of symbols is a big challenge to visually impaired users.

Most, if not all, of the current state-of-the art systems and solutions for mathematical expressions representation to visually-impaired users are in general not pervasive, 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 they present mathematical expressions in only one format. We address these weaknesses by providing a system that provides an ubiquitous access to mathematical expressions. In this system, the user is mobile and he can work on his task at anytime and anywhere he wishes. To do so and to accomplish such task, the user would be likely using different applications. As the user moves from one environment to another, computing resources and interaction context are likely to change. The system takes into account user's profile, data and current environment conditions. However, the change in the set-up should not hamper the user from doing what he needs to do.

[1] http://6dotbraille.com

This paper presents the continuation of our research works [1, 2] with the aims to produce a computing system that suits the needs of the visually impaired users. We discuss the concepts and principles used in the design of this system. We present the modeling and the simulation of our multi-agent system through case studies and formal specification using Petri nets. Also, we propose ontology for modeling contextual information. A simulation of the system's behavior, along with some test case scenarios, is also presented in this work.

The rest of this paper is structured as follows. Section 2 presents related researches and highlights the novelty of our work. Section 3 presents the infrastructure and the architecture of our system. In Section 4, we present the design specification of our infrastructure using the formalism of Petri Nets and the ontology of the contextual information. Future work and conclusion are presented in Section 5.

2. Related Work

There have been tools developed to assist visually-impaired users to make computing and information more accessible, such as Mathtalk [3], Maths [4], DotsPlus [5], EasyMath [6] and AudioMath [7, 8]. In Maths, and MathTalk the user can read, write and manipulate mathematics using a multimedia interface containing speech, Braille and audio. Labradoor (LaTex-to-Braille-Door) [9] converts an expression in LaTex into its Braille equivalent using the Marburg code [10]. MAVIS (Mathematics Accessible To Visually Impaired Students) [11] supports LaTex to Braille translation using Nemth code [12]. VICKIE (Visually Impaired Children Kit for Inclusive Education) [13] and BraMaNet (Braille Mathématique sur InterNet) [14] are transcription tools that convert mathematical document (written in LaTex, MathML, HTML, etc.) to its French Braille representation. DotsPlus is a tactile method of printing documents that incorporates both Braille and graphic symbols (e.g.$\prod$, $\sum$, etc.). For EasyMath [6], the objective is to keep the bi-dimensional structure, used by sighted people, using Braille characters and an overlay keyboard.

Our research investigation indicates that most, if not all, of the current state-of-the art systems and solutions for the presentation of mathematical expressions to visually-impaired users do not satisfy their needs. 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 provides the user with opportunities to access mathematical expressions and presents data output in as many suitable presentation formats as possible after considering interaction context and expression complexity.

An agent is some entity that senses its environment and is capable to react proactively and having the capability of social interaction. A group of agents in a system forms a Multi-Agent System (MAS) [15]. Significant works in agency that are applicable for the visually impaired include Tyflos [16] which 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 situation in order to assist the system to determine the appropriate presentation format, detect and correct system faults in order to reduce the human intervention. This provides some autonomy to visually impaired user when dealing with mathematical expressions.

Context-awareness is required in mobile systems such as pervasive (or ubiquitous) computing where computer devices can sense their environment and adapt their behavior accordingly. Context-awareness is concerned with the acquisition, the abstraction, the understanding and the sharing of context knowledge then the adaptation of behavior according to recognized context. The contextual information must be shared and understood among different architectural components. The Ontology-based modeling seems the most powerful tool [17] for context modeling. Ontology permits to define formally the properties and structure of contextual information and provides knowledge sharing, reasoning on context and interoperability in such environment. A number of ontologies have been developed for use in pervasive computing such as : CoBrA (Context Broker Architecture) [18], Gaia [19] and SOCAM [20]. These works aimed to build ontologies that represent the primitives of pervasive computing [21].

Our objective is to propose a generic and extensible ontology for interaction context modeling that addresses the visually-impaired users.

3. Pervasive MM Computing System

In this section, we present the architectural framework of the paradigm and illustrate the data and control transfer between the components of the system. The next subsections are provided to give more details to each of these components.

3.1. Architectural Framework

The architectural framework of a pervasive MM computing system for the visually impaired users is shown in Figure 1. The layers and their functionalities are as follows: (1) Context Gathering Layer – detects current interaction context; (2) Control and Monitoring Layer – controls the system, coordinates the detection of interaction context, the mathematical expression, its presentation and/or manipulation; (3) Data Analysis Layer – here, the presentation format of the mathematical expression is selected based on available resources and user's situation; (4) Data Access Layer – allows search/edit of mathematical expression; and (5) Presentation Layer – presents mathematical expression via optimal presentation format.

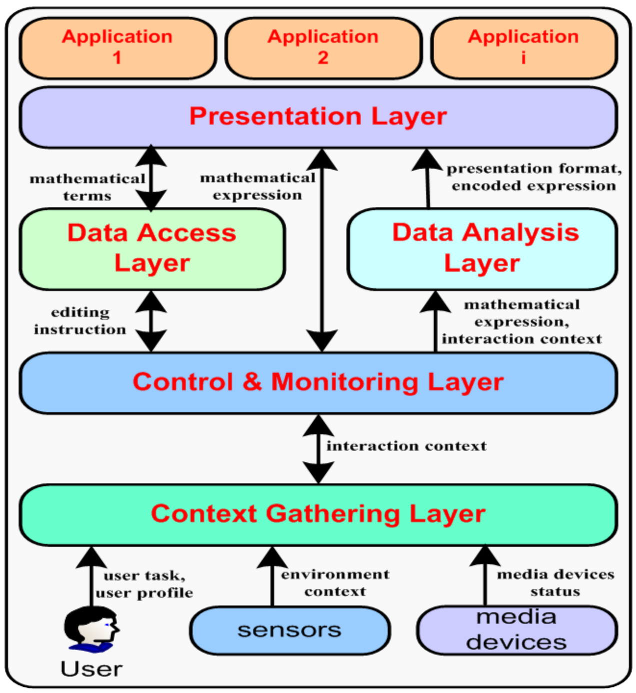

Figure 1. Architectural layer view of the multimodal computing system for visually-impaired users

A multi-agent system is used to implement the architectural framework of Figure 1. There is 3 agents in Context Gathering Layer: User Agent which detects user preferences, such as media and presentation format, Environment Agent which detects the user workplace's noise level and its noise restriction, Device Manager Agent detects available and functional media devices. The Control and Monitoring Layer have a single agent (i.e. System Management Agent) that detects user's interaction context and mathematical expression, and determines how it will be treated. Data Analysis Layer has the intelligent behavior of the system. There are the Analysis and Conversion Agent which obtains MathML expression and converts it to its encoded format and the Machine Learning Agent that determines the optimal presentation format for the given user situation. The Data Access Layer includes the Searching and Editing Agent that obtains editing instructions from the user. Finally, the Presentation Layer (via Translation Agent) converts the encoded format of an expression into its final presentation format and shows the expression by using the appropriate media.

3.2. Contextual Information

Our approach is based on analysis of contextual information and learning techniques to design a system that is able to adapt its behavior according to the current context (i.e. interaction context (IC) and complexity of expression). Here, we present briefly the contextual information, for more details see [2]. IC is formed by combining contexts of the user, his environment, and his computing system.

The user context (UC), in this work, is a function of user profile and preferences. A user profile contains, among others, the user's username, password, his computing devices and identification (i.e. IP addresses) and the user's special needs that determine other affected modalities (i.e. the user is already disqualified from using visual input/output modalities). As a function of modality, the user context 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. For example, being mute prevents the user from using vocal input modality.

The environment context (EC) is the assessment of a user's workplace condition. 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. In our work, the noise level is interpreted from the sampled raw data of a sensor. Also, 50 dB or less is considered "acceptable" while 51 dB or more is considered "noisy". And (2) the environment restriction – identifies whether a workplace imposes mandatory silence or not. In this work, 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. 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.

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 affects the choice of the presentation format. In case of simple expressions (see Figure 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 2). Hence, the complexity of the expression is important for determining the suitable presentation format. In [22], authors proposed a method to determine the complexity of the expression based on: the depth of syntax tree, number of operands and operators.

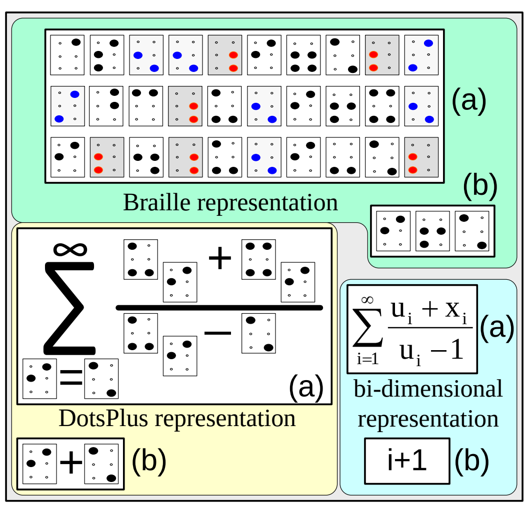

Figure 2. Two sample expressions in bi-dimensional form and its corresponding equivalent in Braille and DotsPlus format

4. Formal Specification and Sample Simulations

4.1. Ontology

In a computing pervasive system, context model have to provide a common definition and understanding of contextual information between components in order to enable a good context sharing in the computing environment. A generic ontology must not be based on predefined aspects of context to enhance its usability and extensibility because these aspects vary from one application to another depending on the domain [23]. Context ontology should be able to capture only the contextual knowledge needed to the adaptation and not all information which can contain useless contextual information. Based on these concepts, we will propose ontology for modeling contextual information. The context ontology should contain only the basic elements which are common to the most pervasive computing systems. The system contains some related components sensors which capture the contextual knowledge. This leads to identify ontology composed of fifteen classes (see Figure 3): User, System, Interaction Context, User task, System Context, Environment Context, User Context, Expression Context, Context negotiator, Sensor, Device, Restriction, Noise Level, Profile and Preferences.

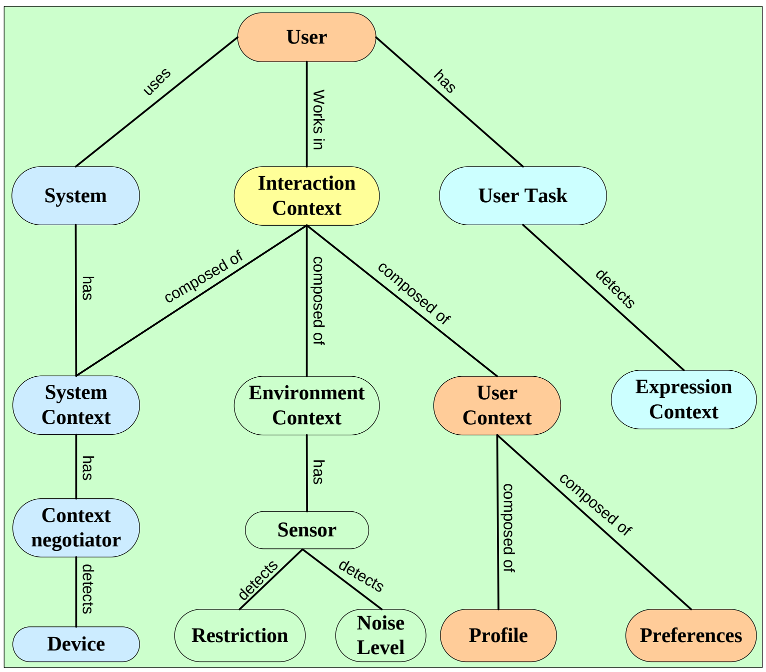

Figure 3. The ontology class identified for a pervasive computing system

The user is the principal actor of the system. He interacts with the system via specific commands. The system allows user to access information; it may be a PC, Laptop, PDA, and Cell phone. The context negotiator detects the media device's context (name, class, characteristics, status, and confidentiality). The device is used to acquire or deliver information or data (e.g. screen, terminal Braille, mouse, keyboard, etc.). The sensor is used to detect the condition of user environment. The user 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.

4.2. Application Scenario

We have used Protégé to implement the ontology of the application and checking its consistency [24]. Protégé is a free open source ontology editor built for working with the Java API for the W3C Web Ontology Language, or OWL [25]. Protégé provides a wide range of features for viewing, manipulating and building ontologies in OWL for interlinked data. In general, it is useful in creating intelligent applications that use those ontologies [26]. We have used protégé for representing the ontology graph shown in Figure 3, after checking its consistency, we get at the end an OWL file coded in XML and specifically in Resource Description Framework (RDF) format. RDF is an application of XML that provides unambiguous methods of expressing semantics. It provides also the facilities to describe the application-specific classes and properties, and to designate which classes and properties are anticipated to be used together [27]. Figure 4 shows a part of the output RDF file we have obtained with protégé.

```
<owl:Ontology rdf:about=""/>
 <owl:Class rdf:ID="Sensor">
  <owl:disjointWith>
   <owl:Class rdf:ID="Preference"/>
  </owl:disjointWith>
  <owl:disjointWith>
   <owl:Class rdf:ID="Media"/>
  </owl:disjointWith>
  <owl:disjointWith>
   <owl:Class rdf:ID="User_Task"/>
  </owl:disjointWith>
  <owl:disjointWith>
   <owl:Class rdf:ID="System_Context"/>
  </owl:disjointWith>
  <owl:disjointWith>
   <owl:Class rdf:ID="Context_Negotiator"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
   <owl:Restriction>
    <owl:someValuesFrom>
     <owl:Class rdf:ID="Restriction"/>
    </owl:someValuesFrom>
    <owl:onProperty>
     <owl:ObjectProperty rdf:ID="detects"/>
    </owl:onProperty>
   </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith>
   <owl:Class rdf:ID="Interaction_Context"/>
  </owl:disjointWith>
```

Figure 4. Sample part of an RDF file for our ontology

When we have an expression, it will be analyzed syntactically and semantically to identify any relevant properties to extract. We present an example of an extracted sentence:

MIKE prefers to use KB in the noisy environments

This sentence is parsed and the type of entities is recognized. For example, in the above sentence *<MIKE>* is recognized as a user name, *<KB>* as a device, and *<NOISY ENVIRONMENT>* is known as a part of an environment context. This knowledge will then be mapped to the ontology classes. If a match does not exist, a module will be used to expand the given terminology. At the end of this stage, three instances will be created for the user, Device, and Environment context to represent the three extracted entities above.

4.3. Formal Specification and Sample Simulations

A formal specification is a mathematical description of software, hardware or system that may be used to develop an implementation. It describes what the system should do, but not how the system should do it. Even without actual implementation, one could determine the overall system behavior via formal specification. There are various techniques and languages to demonstrate formal specification of a system. In the next subsection, a formal specification using Petri Net[2] is demonstrated. It is appropriate for demonstrate a model of our system. The user, his environment, his computing device and all available media device form the input of this net. The interaction finished when the user does not have more commands. We did use HPSim Net[3] for Petri Net specification shown in Figure 5.

The Petri Net was defined by Carl Adam Petri in 1962. It extends the concept of state machine to include concurrency. Petri Net is a formal, graphical, executable technique for the specification and analysis of a concurrent, discrete-event dynamic system. A Petri-Net diagram is represented by an ellipse called place (basically a state), a rectangle called transition (basically 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. 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.

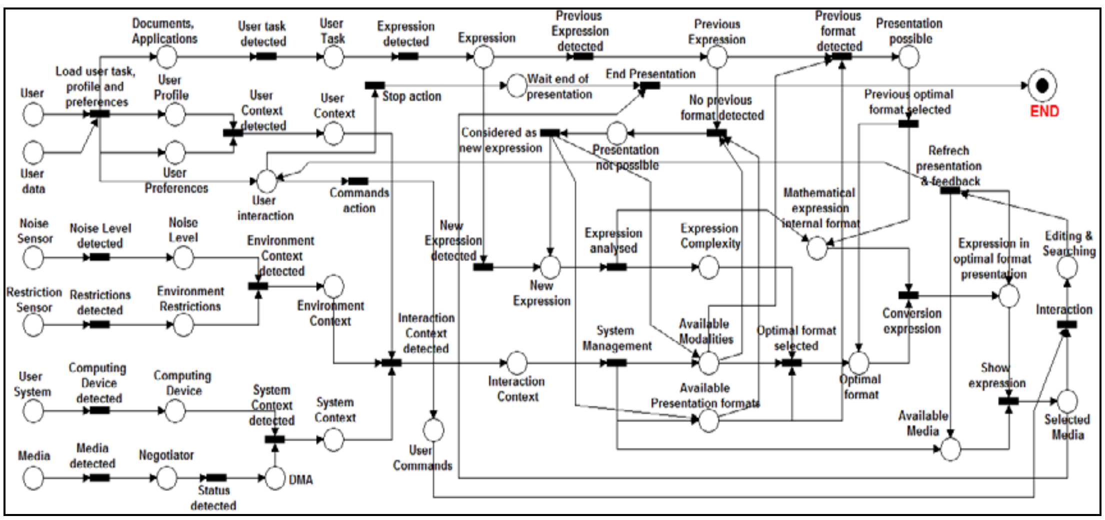

Figure 5. Formal specification depicting different pre-condition scenarios and their resulting post-condition scenarios using Petri Net

A marking of a Petri Net is reachable if, starting in the initial marking, a sequence of transition firings exist that produces it. A Petri Net is bounded if there is a maximum to the number of tokens in its reachable markings. 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 x T → N is the forward incidence function, B: P x T → N is the backward incidence function and N = 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 → N. "Firing" of any transition changes the marking of the Petri Net. The Petri Net specification of the system is shown in Figure 5.

[2] http://www.petrinets.info/
[3] http://www.winpesim.de/

As depicted in Figure 5, there are 6 places that are all activated simultaneously, namely the user which logs in a particular computing device, his task, and the media activated by the system, the level of noise in the environment and the noise restraint imposed by the environment.

The topmost two places yield the activation of the user task, the user context and the user preferences and interaction. The environment noise level produces a final output which is the current noise level (i.e. either quiet/acceptable, or noisy). The noise restraint imposed by the environment could only be silence required or silence optional. The final output of these places represents the environment context. The media and the user computing system places yield a final output that is the system context. Then the interaction context is detected. The expression is extracted from the user task. We demonstrated also the process for find the optimal presentation format based on the user situation and the expression complexity.

In Petri Net specification, we capture only the final output after all the values of the 2 parameters (i.e. interaction context and complexity of the expression) are taken into account. Briefly, the output of the Petri Net is the activation of the appropriate presentation format, the presentation of data and the selection of a set of media for activation as the appropriate input-output devices. The media used to instantiate the presentation format are based on user's preferences found in the user profile. The Petri Net specification illustrates all possible variations of the user interaction; the simulation produces 2 possible outputs (i.e. the user continues working on the expression or he ends the interaction with the system). Simulations of the selection of appropriate devices and others sub-systems are presented in previous works [2, 28].

5. Conclusion

This paper demonstrates the model of a pervasive computing system with MM that supports the needs of the visually impaired users. The paradigm is explained in general via its architectural framework. In this work, we have presented the architecture of our adaptive system. Also, we have presented the agents' functionalities in the system's layers.

The adaptation of a computing system to the needs of a mobile user is essential so that the user could continue working on his task at anytime and anywhere, thereby increasing his productivity. The selection of modalities, media, and presentation formats depends on the contextual information. We have presented the components of this information as interaction context (i.e. combined contexts of the user, his environment and his computing system) and nature of the expression.

We validated the system behavior through a simulation and formal specifications realized with Petri nets. The analysis of the contextual information leads to identify ontology composed of fifteen classes. We have used Protégé to implement and check the ontology consistency of the application.

Using ontology in our work is aimed at increasing the system's portability to other domains. Building a cross-domain system is one of our aims, and will be fully explored in the next stage of development.

6. References

[1] Awde, A., C. Tadj, and 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, No. 1, Vol. 1, pp. 17-27, ISSN 1798-2448, 2009.
[2] Awde, A., et al., An Adaptive Multimodal Multimedia Computing System for Presentation of Mathematical Expressions to Visually- Impaired Users. Journal of Multimedia, 2009. 4(3).
[3] Edwards, A.D.N. and R.D. Stevens, A Multimodal Interface for Blind Mathematics Students, in INSERM. 1994: Paris, France. p. 97 - 104.
[4] Cahill, H., et al., Ensuring Usability in MATHS, in The European Context for Assistive Technology. 1995, IOS Press: Amsterdam. p. 66-69.
[5] Preddy, M., et al. Dotsplus: How-to Make Tactile Figures and Tactile Formatted Math. in Int. Conference on Technology and Persons with Disabilities. 1997. Los Angeles, CA, USA.
[6] Podevin, A., Accès aux formules mathématiques par des personnes non voyantes : étude et définition d'une méthode adaptée. 2002, Université de CAEN.

[7] Ferreira, H. and D. Freitas, Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project, in 9th Int. Conference on Computer Helping People with Special Needs (ICCHP). 2004, Springer Lecture Notes in Computer Science (LNCS): Paris, France. p. 678-685.
[8] Ferreira, H. and D. Freitas, AudioMath: Towards Automatic Readings of Mathematical Expressions, in Human-Computer Interaction International,Las Vegas, Nevada, USA, 2005.
[9] Batusic, M., K. Miesenberger, and B. Stoger. Labradoor, A Contribution To Making Mathematics Accessible For The Blind. in International Conference on Computers Helping People with Special Needs. 1998. Oldenbourg, Wien, München.
[10] Epheser, H., D. Pograniczna, and K. Britz, Internationale Mathematikschrift fur Blinde, . Marburg (Lahn), Deutsche Blindenstudienanstalt, 1992.
[11] Karshmer, A.I., et al., Reading and Writing Mathematics: the MAVIS Project, in Proceedings of the 3rd int. ACM conference on Assistive technologies (ASSETS). 1998, ACM SIGCAPH Computers and the Physically Handicapped Marina del Rey, California, USA. p. 136-143.
[12] Nemeth, A., The Nemeth Braille Code for Mathematics and Science Notation. American Printing House for the Blind, 1972.
[13] Moço, V. and D. Archambault, VICKIE: A Transcription Tool for Mathematical Braille, in 7th European Conference for the Advancement of Assistive Technology in Europe (AAATE). 2003, IOS press: Dublin, Ireland.
[14] Schwebel, F. and R. Goiffon, BraMaNet: Quelques règles simples à connaître pour qu'un aveugle puisse lire vos documents mathématiques et vos pages web, Jour. nationales Caen, France, 2005.
[15] Wooldridge, M., An Introduction to Multiagent Systems. 2002, Chichester, England: Wiley.
[16] 16. 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, IEEE Computer Society. p. 230 – 235.
[17] Guarino, N., Understanding, Building, And Using Ontologies. International Journal of Human-Computer Studies, 1997. 46(2-3): p. 293-310.
[18] Kagal, L., et al., Centaurus: a framework for intelligent services in a mobile environment, in Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCSW'01). 2001, IEEE Computer Society: Washington, DC. p. 195-201.
[19] Roman, M., et al., Gaia: a middleware infrastructure to enable active spaces. IEEE Pervasive Computing, 2002: p. 74-83.
[20] Gu, T., et al., An Ontology-Based Context Model in Intelligent Environments in In Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'04). 2004, Society for Computer Simulation. p. 270-275.
[21] Ye, J., et al., Ontology-based Models in Pervasive Computing Systems. The Knowledge Engineering Review, Cambridge University Press, 2007. 22(4): p. 315-347.
[22] Awde, A., Y. Bellik, and C. Tadj, Complexity of Mathematical Expressions in Adaptive Multimodal Multimedia System Ensuring Access to Mathematics for Visually Impaired Users. Int. Journal of Computer and Information Science and Engineering, 2008. 2(2): p. 103-115.
[23] Miraoui, M., C. Tadj, and C.B. Amar, Context Modeling and Context-Aware Service Adaptation for Pervasive Computing Systems. International Journal of Computer and Information Science and Engineering, 2008. 2(3): p. 148-157.
[24] Horridge, M., et al., A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools. 2004, University Of Manchester.
[25] Alani, H., et al., Using Protege for Automatic Ontology Instantiation, in 7th International Protégé Conference. 2004: 6-9 July 2004, Bethesda, Maryland, USA.
[26] Rubin, D., et al., Protégé-OWL: Creating Ontology-Driven Reasoning Applications with the Web Ontology Language, in AMIA Annual Symposium Proceedings. 2005.
[27] Ye, J., et al., Ontology-based Models in Pervasive Computing Systems. The Knowledge Engineering Review, 2007. 22(4): p. 315-347.
[28] Awde, A., et al., Middleware for Ubiquitous Access to Mathematical Expressions for Visually-Impaired Users. Ubiquitous Computing and Communication Journal, 2008. 3(5).