MIDDLEWARE FOR UBIQUITOUS ACCESS TO MATHEMATICAL EXPRESSIONS FOR VISUALLY-IMPAIRED USERS

Ali Awde[1], Manolo Dulva Hina[1,2], Chakib Tadj[1]
[1] Université du Québec, École de technologie supérieure, Montréal Québec, Canada
[2] PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, France
ali.awde.1@ens.etsmtl.ca, manolo-dulva.hina.1@ens.etsmtl.ca, ctadj@ele.etsmtl.ca
Yacine Bellik,
Université de Paris-Sud, France LIMSI-CRNS, Université de Paris-Sud, France
yacine.bellik@limsi.fr
Amar Ramdane-Cherif [2,3]
[3] LISV Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines, France
rca@prism.uvsq.fr

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.

1 INTRODUCTION

In pervasive computing [1], 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 [2] or LaTex [3] 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 2 provides a brief review of the state of the art related to our work; Section 3 lists down the challenges, proposed solutions and contributions of our work. Section 4 presents the infrastructure and system architecture of our middleware. In Section 5, we explain the concept of machine learning in relation to the system design so it adapts seamlessly to the given interaction context. Section 6 presents our methodology to configure an application supplier and optimize settings, taking into account user's satisfaction. Section 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 8, we show the design specification of our infrastructure. Future works and conclusion are presented in Section 9.

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 [4]. Fortunately, some applications were developed to lessen the complexity of performing similar task, some of them are MathTalk [5], Maths [6] and AudioMath [7, 8]. 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) [9] and BraMaNet (Braille Mathématique sur InterNet) [10] are transcription tools that convert mathematical document (in LaTex, MathML, HTML, etc.) to its French Braille representation. Labradoor (LaTex-to-Braille-Door) [11] converts an expression in LaTex into its Braille equivalent using the Marburg code. MAVIS (Mathematics Accessible To Visually Impaired Students) [12] 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 [13] which is used in the USA, Canada, New Zealand, Greece and India; the Marburg code [14] used in Germany and Austria, the French Math code [15].

Lambda [16] translates expression in MathML format into multiple Braille Math codes that are used in the European Union. ASTER (Audio System for Technical Readings) [17] 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 [18], 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) [19, 20] have been widely used in many applications, such as on a large, open, complex, mission-critical systems like air traffic control [21]. 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 [18, 22]. 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.

Our focus has always been pervasive multimodality for the blind. This work was initially inspired by [23]. 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 [24] and a talking Braille [25]. 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.

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.

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.

4.1 System Architecture

Figure 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:

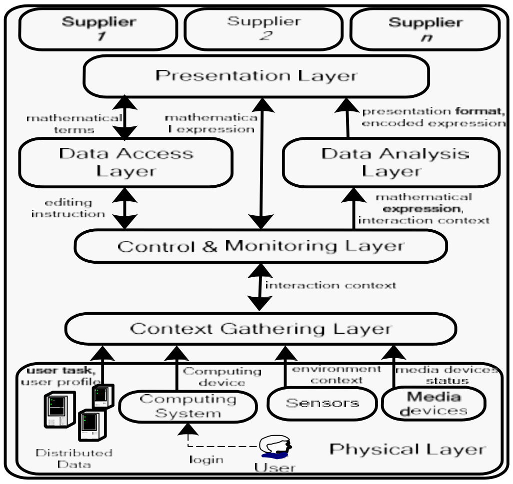


Figure 1: The architectural abstraction of a generic MM computing system for visually-impaired users.

(1) 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.

(2) The Data Access Layer – Here, users may edit or search for a term in a mathematical expression.

(3) 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.

(4) 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.

(5) 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.

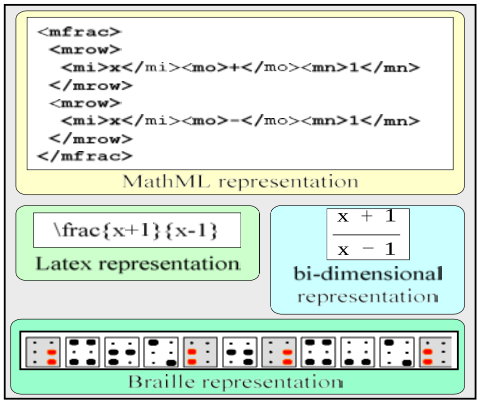
(6) The Physical Layer – It contains all the physical entities of the system, including media devices and sensors. The raw data from this layer are sampled and interpreted and forms the current instance of interaction context.

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.

4.2.1 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 2 shows a specimen fraction (in bi-dimensional form) and its equivalent representation in MathML, LaTex and Braille.



```
<mfrac>
 <mrow>
  <mi>x</mi><mo>+</mo><mn>1</mn>
 </mrow>
 <mrow>
  <mi>x</mi><mo>-</mo><mn>1</mn>
 </mrow>
</mfrac>
```

```
\frac{x+1}{x-1}
```

$$\frac{x + 1}{x - 1}$$

Figure 2: A fraction in bi-dimensional form and its corresponding equivalent in LaTex, MathML and Braille.

4.2.2 Reading and presenting the mathematical expression in a document

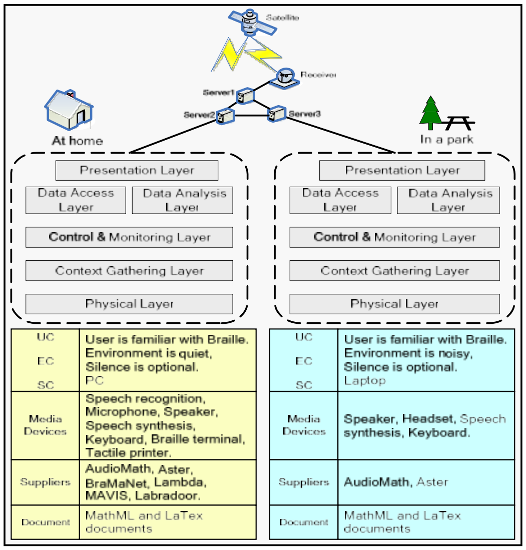
A ubiquitous mathematical document becomes readily available to the user once a connection is made between the user's terminal and a member of the server group. In general, the processes for reading or presenting a mathematical document to the user are as follows: (i) Our middleware retrieves the mathematical 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.

4.2.3 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 4.1) allows users to search for a term within the equation and edit same. Once editing is done, the document is saved accordingly.

4.3 Anytime, Anywhere Mathematical Computing

As shown in Figure 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.



UC EC SC	User is familiar with Braille. Environment is quiet, Silence is optional. PC
Media Devices	Speech recognition, Microphone, Speaker, Speech synthesis, Keyboard, Braille terminal, Tactile printer.
Suppliers	AudioMath, Aster, BraMaNet, Lambda, MAVIS, Labradoor.
Document	MathML and LaTex documents

UC EC SC	User is familiar with Braille. Environment is noisy, Silence is optional. Laptop
Media Devices	Speaker, Headset, Speech synthesis, Keyboard.
Suppliers	AudioMath, Aster
Document	MathML and LaTex documents

Figure 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 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 DESIGNING AN INTERACTION CONTEXT ADAPTIVE SYSTEM

Here, we elaborate on the system's knowledge acquisition that makes it adaptive to the given interaction context.

5.1 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 [26]. 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 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.

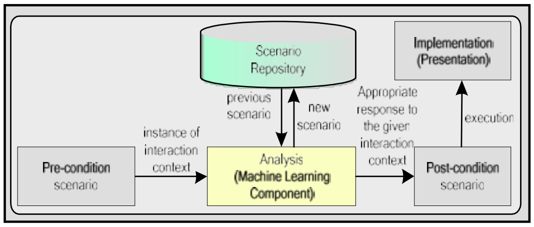


Figure 4: A generic ML system: an instance of interaction context serves as input to a ML component yielding a corresponding post-condition result.

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).

Incremental machine learning (IML) is a progressive acquisition of knowledge. In the literature, various IML algorithms exist, such as the candidate elimination [26], 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) [28]. 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 X with n components and a set of outputs Y with m components. Let f be the function to be learned which will map some elements in X to the elements in Y, and h be the hypothesis about this function. Furthermore, let X be the set of interaction context instances, i.e. set of pre-condition scenarios. Hence, Y would be a set of post-condition scenarios, and the mapping between the pre- and post-condition scenarios is denoted by f: X -7 Y. See Figure 5.

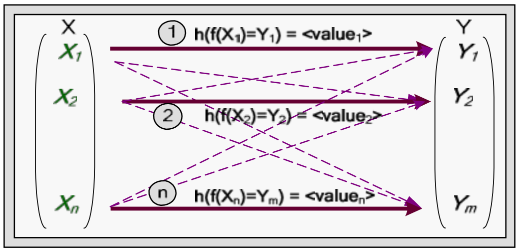


Figure 5: The relation between pre-condition scenarios and the post-condition scenarios using supervised learning.

As shown in the diagram, X_i = an element of X, and $i \in 1 \ldots n$, and Y_j = a element of Y, and $j \in 1 \ldots m$. 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 $h(f(Xnew) = Y1)$, $h(f(Xnew) = Y2)$, … , $h(f(Xnew) = Ym)$ 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.

5.2 Basic Principles of Interaction Context

In this paper, the following logic symbols are used: $\otimes$ = Cartesian product yielding a set composed of tuples, $£_1$ = 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, $IC = \{IC_1, IC_2, \ldots, IC_{max}\}$, is a set of all possible interaction contexts. At any given time, a user has a specific interaction context i denoted IC_i, $1 \le i \le$ max. Formally, an interaction context is a tuple composed of a specific user context (UC), environment context (EC) and system context (SC). An instance of IC may be written as:

$$IC_i = UC_k \otimes EC_l \otimes SC_m \qquad (1)$$

where $1 \le k \le max_k$, $1 \le l \le max_l$, and $1 \le m \le max_m$, 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:

$$UC_k = \bigotimes_{x=1}^{max_k} ICParam_{kx} \qquad (2)$$

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

$$EC_l = \bigotimes_{y=1}^{max_l} ICParam_{ly} \qquad (3)$$

$$SC_m = \bigotimes_{z=1}^{max_m} ICParam_{mz} \qquad (4)$$

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 $M = \{V_{in}, T_{in}, V_{out}, T_{out}\}$ wherein V_{in} = vocal input, V_{out} = vocal output, T_{in} = tactile input and T_{out} = tactile output then modality is possible under the following condition:

$$\text{Modality Possible} = (V_{in} \vee T_{in}) \wedge (V_{out} \vee T_{out}) \qquad (5)$$

The failure of modality, therefore, can be specified by the relationship:

$$\text{Modality Failure} = ((V_{in} = \text{Failed}) \wedge (T_{in} = \text{Failed})) \vee ((V_{out} = \text{Failed}) \wedge (T_{out} = \text{Failed})) \quad (6)$$

Let M_j = element of the power set of M, that is, $M_j \in P(M)$ 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, $IC_i = (A_1, A_2, ..., A_n)$ 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,

$$h\left(f\left(IC_i\right) \rightarrow M_j\right) = <\text{suitability\_score}> \quad (7)$$

In Mathematics, function h can be written as:

$$h = P(M_j/IC_i) \quad (8)$$

which should be read as the probability of the occurrence of M_j given IC_i. To simplify calculation, Bayes Theorem [29], given below, can be adopted:

$$P(M_j/IC_i) = \frac{P(IC_i/M_j) \times P(M_j)}{P(IC_i)} \quad (9)$$

The implementation of Bayes Theorem leads to the Naive Bayes algorithm [26]. The Naive Bayes algorithm is a classification algorithm that assumes that IC_i attributes $A_1,…,A_n$ are all conditionally independent of one another given a post condition M_j. The representation of $P(IC_i|M_j)$ becomes:

$$P(IC_i \mid M_j) = \prod_{i=1}^{n} P(A_i \mid M_j) \quad (10)$$

Here, our goal is to train a classifier that, given a new IC_i to classify, will provide the probability distribution over all possible values of M (i.e. M_1, M_2, …, M_m). Given that $IC_i = (A_1, A_2, ..., A_n)$, then Eq. (9) becomes:

$$P(M_j \mid A_1 ... A_n) = \frac{P(M_j) P(A_1 ... A_n \mid M_j)}{\sum_{k=1}^{m} P(M_k) P(A_1 ... A_n \mid M_k)} \quad (11)$$

Equation 11 can also be written as:

$$P(M_j \mid A_1 ... A_n) = \frac{P(M_j) \prod_{i=1}^{n} P(A_i \mid M_j)}{\sum_{k=1}^{m} P(M_k) \prod_{i=1}^{n} P(A_i \mid M_k)} \quad (12)$$

which is the fundamental equation for the Naive Bayes classifier. Given a new instance of interaction context $IC_{new} = (A_1 … A_n)$, the equation shows how to calculate the probability that M_j will take on any given value, given the observed attribute values of IC_{new} and given the distributions $P(M_j)$ and $P(A_i|M_j)$ estimated from the training data (SR). If we are interested only in the most suitable value of M_j, then we have the Naive Bayes classification rule:

$$h_{best} = \hat{M} = \arg\max_j \left[\frac{P(M_j) \prod_{i=1}^{n} P(A_i \mid M_j)}{\sum_{k=1}^{m} P(M_k) \prod_{i=1}^{n} P(A_i \mid M_k)} \right] \quad (13)$$

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

$$h_{best} = \hat{M} = \arg\max_j \left[P(M_j) \prod_{i=1}^{n} P(A_i \mid M_j) \right] \quad (14)$$

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

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 $P(M) = \{\{V_{in}\}, \{T_{in}\}, \{V_{out}\}, \{T_{out}\}, \{V_{in}, T_{in}\}, \{V_{in}, V_{out}\}, \{V_{in}, T_{out}\}, \{V_{in}, T_{in}, V_{out}\}, \{V_{in}, T_{in}, T_{out}\}, \{V_{in}, V_{out}, T_{out}\}, \{V_{in}, T_{in}, V_{out}, T_{out}\}, \{T_{in}, V_{out}\}, \{T_{in}, T_{out}\}, \{T_{in}, V_{out}, T_{out}\}, \{V_{out}, T_{out}\}, \{\}\}$. M_j, therefore, evaluates the suitability score of each element of P(M). $\hat{M} = h_{best}$ is then chosen from one of these elements. The selected element is one that satisfies Eq. (14) and one with the highest suitability score.

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: Modality -7 Media, Priority. See Figure 6.

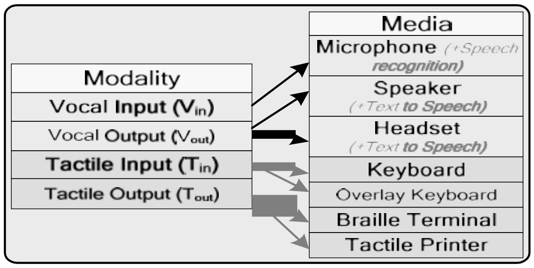


Figure 6: Media selections to support a modality.

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 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 f_1, we can obtain the relationship in implementing multimodality:

$f_1(V_{in})$ = (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)

Therefore, the assertion of modality, as expressed in Eq. (5), with respect to the presence of media devices becomes:

$$\begin{aligned}&\text{Modality Possible} = \\ &((\text{MIC} \wedge \text{Speech Recognition}) \vee (\text{KB} \vee \text{OKB})) \wedge \\ &(((\text{SPK} \vee \text{HST}) \wedge \text{Text-to-Speech}) \vee (\text{BRT} \vee \text{TPR}))\end{aligned} \tag{15}$$

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 7 shows the ML knowledge acquisition.

A mathematical document of type LaTex (with sion .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: Document, {media devices} -7 Preferred supplier, 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)), etc.}

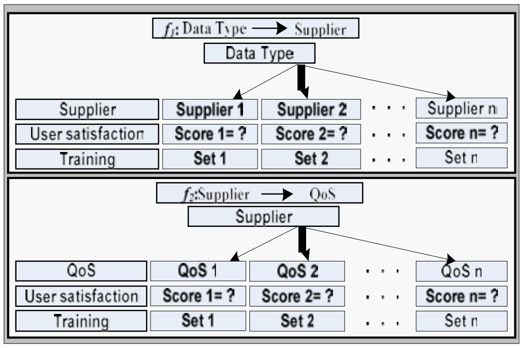


Figure 7: The training process for ML knowledge acquisition: (Up) the mapping of data type to a supplier and (Down) building a user's preferred QoS dimensions for a supplier.

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 -7 QoS dimension j, Priority) Also, 1 ≤ j ≤ qos_max (maximum number of QoS dimensions). Priority is of type £$_1$. 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 characters per line, 1)), (Lambda, (60 characters per line, 2)), (Lambda, (80 characters per line, 3)), (Lambda, (French Math code, 1)), (Lambda, (Nemeth (high volume, 2)), (AudioMath, (low volume, 3)), etc.}

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.

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:

$$\text{QoS Dimension space} = \bigotimes_{i=1}^{qos\_max} D_i \qquad (16)$$

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.

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:

$$\text{Supplier preferences} = h_s^{x_s} \bullet \quad f_s \qquad (17)$$

c_s

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 $c_{supplier1} = 1$, $c_{supplier2} = 1 - 1/n$, $c_{supplier3} = 1 - 1/n - 1/n$, 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:

$$c_{supplier\ i} = 1 - \sum_{1}^{i-1} (1/n) \qquad (18)$$

The term f_s: 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 $f_s = 1 - m/(n + 1) = 1 - ¼ = 0.75$. In general, the user satisfaction with respect to supplier features is given by:

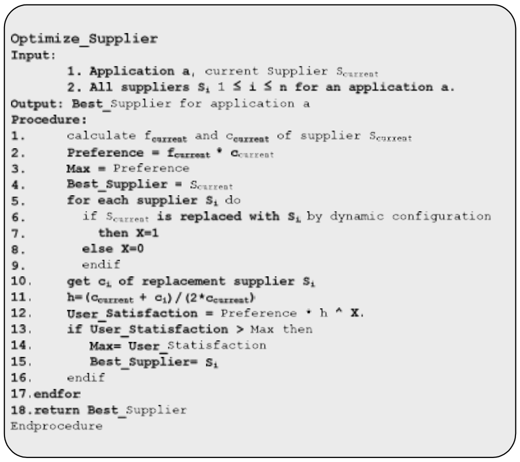
$$f_{supplier} = 1 - \frac{m}{n+1} \qquad (19)$$

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:

$$h_s = \arg\max \left(\frac{c_s + c_{rep}}{2 \times c_s} \right) \qquad (20)$$

where c_{rep} is value obtained from Eq. (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 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 $S_{current}$
        2. All suppliers $S_i$ $1 \le i \le n$ for an application a.
Output: Best_Supplier for application a
Procedure:
        calculate $f_{current}$ and $c_{current}$ of supplier $S_{current}$
        Preference = $f_{current}$ * $c_{current}$
        Max = Preference
        Best_Supplier = $S_{current}$
        for each supplier $S_i$ do
          if $S_{current}$ is replaced with $S_i$ by dynamic configuration
            then X=1
          else X=0
          endif
        get $c_i$ of replacement supplier $S_i$
        h=($c_{current}$ + $c_i$)/(2*$c_{current}$)
        User_Satisfaction = Preference * h ^ X.
        if User_Statisfaction > Max then
           Max= User_Statisfaction
           Best_Supplier= $S_i$
        endif
endfor
return Best_Supplier
Endprocedure
```

Figure 8: Algorithm for optimized supplier and its QoS configuration.

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 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>* password: *<password>*	Manually-Disabled : *<Yes/No>*	computing device1: *<MAC address 1>* :: computing device n : *<MAC address n>*
	Mute : *<Yes/No>*	
	Deaf : *<Yes/No>*	
	Unfamiliarity with Braille: *<Yes/No>*	

Figure 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. (6), with respect to UC parameters is:

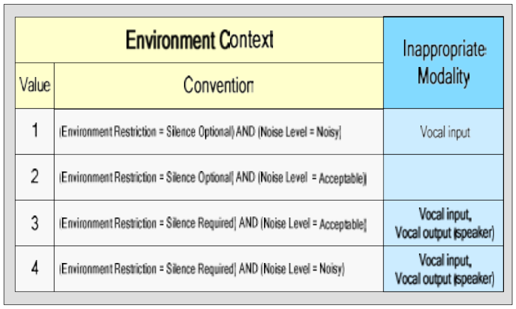
$$\begin{aligned}\text{Modality Failure} &= ((\text{Manually- Disabled}) \wedge (\text{Mute})) \\ &\vee ((\text{Manually- Disabled} \vee \text{Unfamiliar with Braille}) \\ &\wedge (\text{Deaf}))\end{aligned} \tag{21}$$

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 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:

$$EC = (\text{Noise Level}) \wedge (\text{Environment Restriction}) \tag{22}$$

Table 1 shows the affected modality based on environment's context. It also shows the convention table we have adopted for EC.

Table 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 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.

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 Table 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.

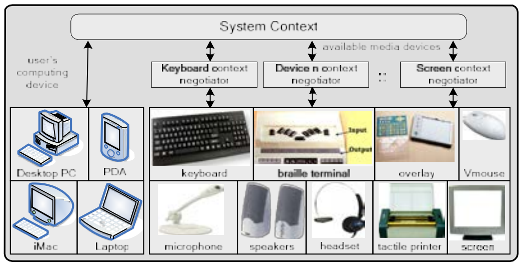
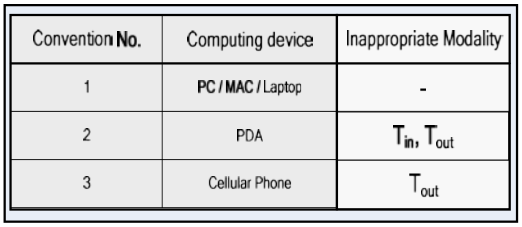


Figure 10: The system context as function of user's computing device and available media devices.

Table 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:

$$V_{in}\ \text{Failure} = (\text{User} = \text{Mute}) \vee (\text{Noise Level} = \text{Noisy}) \vee (\text{Environment Restriction} = \text{Silence required}) \tag{23}$$

$$V_{out}\ \text{Failure} = (\text{User} = \text{Deaf}) \tag{24}$$

$$T_{in}\ \text{Failure} = (\text{User} = \text{Manually-Disabled}) \vee (\text{ComputingDevice} = \text{PDA}) \tag{25}$$

$$\begin{aligned} T_{out}\ \text{Failure} = {} & (\text{User} = \text{Manually-Disabled}) \vee \\ & (\text{User} = \text{Unfamiliar with Braille}) \vee \\ & (\text{Computing Device} = \text{PDA}) \vee \\ & (\text{Computing Device} = \text{Cellphone}) \end{aligned} \tag{26}$$

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.

8.1 Selection of Modalities

- A_2 = {true | false} = if user is mute,
- A_3 = {true | false} = if user is deaf,
- A_4 = {true | false} = user familiarity with Braille,
- A_5 = {quiet | noisy} = environment's noise level,
- A_6 ={silence required | silence optional} = environment's noise level restriction, and
- A_7 = {PC or Laptop or MAC | PDA | Cellphone} = user's computing device.

The set of possible modalities (i.e. refer to Eq. (5)) is given by M = {M_1,M_2,M_3,M_4,M_5,M_6, M_7, M_8, M_9} wherein $M_1 = \{T_{in}, T_{out}\}$; $M_2 = \{T_{in}, V_{out}\}$; $M_3 = \{V_{in}, V_{out}\}$; $M_4 = \{V_{in}, T_{out}\}$; $M_5 = \{V_{in}, T_{out}, V_{out}\}$; $M_6 = \{T_{in}, T_{out}, V_{out}\}$; $M_7 = \{T_{in}, V_{in}, V_{out}\}$; $M_8 = \{T_{in}, V_{in}, T_{out}\}$; $M_9 = \{T_{in}, V_{in}, T_{out}, V_{out}\}$ (see its derivation from Section 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 = False, A_2 = False, A_3 = False, A_4 = True, (ii) environment context: the user is in a classroom, then A_5 = noisy, A_6 = silence required, (iii) system context: the user works on a laptop; A_7 = 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 11. The given figure is generated by using WEKA (Waikato Environment for Knowledge Analysis) [30] 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).

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 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:

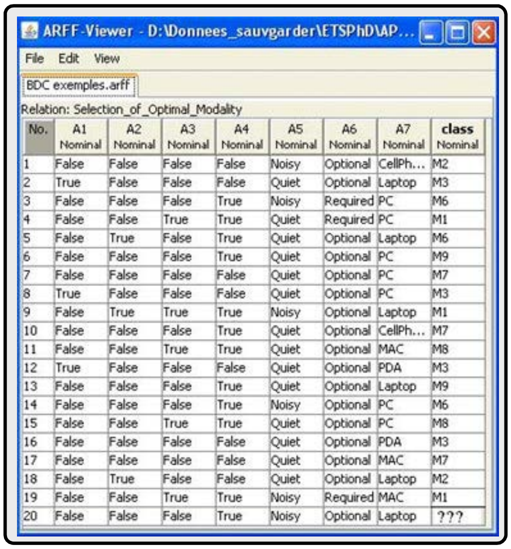
$\text{Suitability\_Score}(M_1) = P(A_1 = \text{False} \mid M_1) \times P(A_2 = \text{False} \mid M_1) \times \ldots \times P(A_7 = \text{Laptop} \mid M_1) \times P(M_1) = 1 \times 0.67 \times 0 \times \ldots. \times 3/19 = 0$

Consider, for example, an interaction context that is composed of the following parameters: IC = (A_1, A_2, A_3, A_4, A_5, A_6, A_7) wherein:

- A_1 = {true | false} = if user is manually disabled, wherein $P(A_1 = False \mid M_1) = 3/3$, $P(A_2 = False \mid M_1) = ⅔$, $P(A_3 = False \mid M_1) = 0/3$, $P(A_4 = True \mid M_1) = 3/3$, $P(A_5 = Noisy \mid M_1) = ⅓$, $P(A_6 = silence\ optional \mid M_1) = ⅓$, and $P(A_7 = Laptop \mid M_1) = ⅓$. 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:

$Suitability\_Score\ (M_6) = P(A_1 = False \mid M_6) \times P(A_2 = False \mid M_6) \times \ldots \times P(A_7 = Laptop \mid M_6) \times P(M_6) = 1 \times ⅔ \times 1 \times 1 \times ⅔ \times ⅔ \times ⅓ \times 3/19 = 0.015671$



Relation: Selection_of_Optimal_Modality

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

As explained in Section 5.3, $M_6 = \{T_{in}, T_{out}, V_{out}\}$ respects the conditions imposed in Eq. (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).

8.2 Selection of Media Devices

As stated in Section 5.4, 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 = {~~(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}, (BRT,2)), (T_{out}, ((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.

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, {KB, BRT}) (MAVIS, 1)), ((.tex, {KB,BRT}) (LaBraDoor, 2)), ((.tex, {KB,BRT}) (VICKIE, 3)), ((.tex, {KB, SPK |HST}), (ASTER, 1)),((.xml, {KB,BRT}), (Lambda, 1)), (.xml, {KB,BRT}), (BraMaNet, 2)),((.xml, {KB, SPK|HST}), (AudioMath,1)), ((.xml, {KB, SPK|HST}), (VoiceXML, 2))}

Formally, ∀x: data format d: media devices, ∃y: Supplier and p: Priority of type $Ƚ_1$| (x, d) -7(y, p) ∈ f_2.

Given that supplier priority is involved in f_2 then the most-preferred supplier is sought. With reference to Eq. (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) -7 not possible
Case 2: (Lambda, LaBraDoor) -7 alternative 1
Case 3: (Lambda, VICKIE) -7 alternative 2

Then the replacement selection is based on user satisfaction score:

User Satisfaction (Case 2) =(1+1+⅔)/3 = 8/9 = 0.89
User Satisfaction (Case 3) =(1+1+⅓)/3 = 7/9 = 0.78

Hence, Case 2 is the preferred alternative. Formally, if f_2: Document Format, {Media devices} -7 (Supplier, Priority) where Priority: $Ƚ_1$, then the chosen supplier is given by: ∃(doc, m): (Document, Media devices) ∀y: Supplier, ∃p1: Priority, ∀p2: Priority | (doc, m) ● y -7(y, p1) ∈ f_2 ∧ (p1 < p2).

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:

$c_{MAVIS} = 1.0$, $c_{LaBraDoor} = 2/3$, $c_{VICKIE} = 1/3$.

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 $f_s = 1 – m/(n + 1) = 1 – ¼ = 0.75$. This also reduces the user's satisfaction, as given by the relationship $c_{MAVIS} * f_{MAVIS} = (1.0)(0.75) = 0.75$.

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 $c_{BraMaNet} = 2/3 = 0.67$. What would happen if it will be replaced by another supplier through dynamic reconfiguration ($x_{supplier} = 1.0$)? Using the relationship $h_{supplier} = (c_{supplier} + c_{replacement}) / 2* c_{supplier}$ then the results of possible alternative configurations are as follows:

Replacing BraMaNet (supplier 2):

Case 1: Replacement by MAVIS (supplier 1): $(0.67)(1) * [(0.67 + 1)/2*(0.67)]^1 = 0.835$

Case 2: Replacement by itself (supplier 2): $(0.67)(1) * [(0.67 + 0.67)/2*(0.67)]^1 = 0.67$

Case 3: Replacement by VICKIE (supplier 3): $(0.67)(1) * [(0.67 + 0.33)/2*(0.67)]^1 = 0.50$

Hence, if the reconfiguration aims at satisfying the user, then the second-ranked supplier should be replaced by the top-ranked supplier.

8.5 Specification for Detecting Suitability of Modality

Petri Net [1] 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. In the specifications in this paper, only a snapshot of one of the many outcomes is presented due to space constraints. We use HPSim[2] in simulating Petri Net.

In Figure 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 7. 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 $A_1,\ldots,A_7$ 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).

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 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.

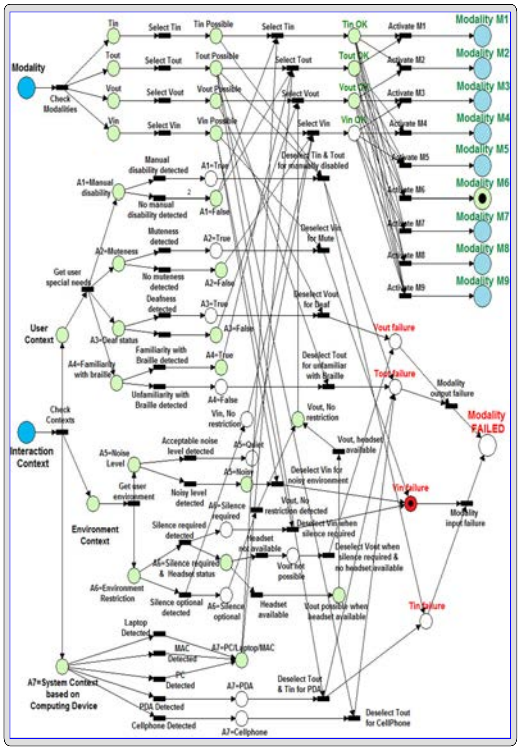
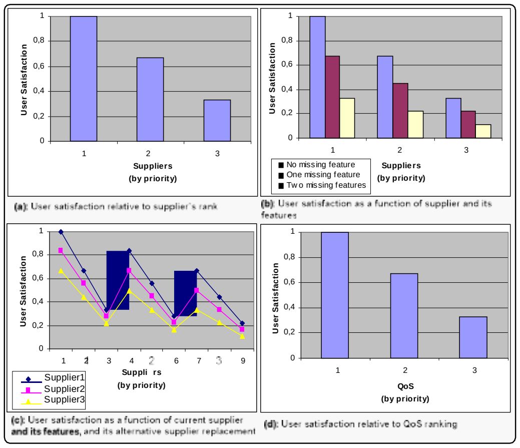


Figure 12: A snapshot of the simulated selection of optimal modality based on interaction context.



(a): User satisfaction relative to supplier's rank

(b): User satisfaction as a function of supplier and its features

(c): User satisfaction as a function of current supplier **and its features**, and its alternative supplier replacement

(d): User satisfaction relative to QoS ranking

Figure 13: Various graphs showing variations of user's satisfaction with respect to its preferred supplier and QoS dimension and their replacements.

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

---

[1] http://www.winpesim.de/petrinet

[2] "HPSim. http://www.winpesim.de

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.

10 REFERENCES

1. Satyanarayanan, M., Pervasive Computing: Vision and Challenges. IEEE Personal Communications, 2001(August): p. 10-17.
2. MathML. http://www.w3.org/Math [cited.
3. Lamport, L., LaTeX: The Macro Package, http://web.mit.edu/texsrc/source/info/latex2e.pdf. 1994.
4. Stöger, B., et al., Mathematical Working Environment for the Blind Motivation and Basic Ideas, in ICCHP. 2004, Springer. p. 656-663.
5. Edwards, A.D.N. et al., A Multimodal Interface for Blind Mathematics Students. in INSERM. 1994. Paris, France.
6. Cahill, H., et al., Ensuring Usability in MATHS, in The European Context for Assistive Technology. 1995, IOS Press: Amsterdam. p. 66-69.
7. Ferreira, H., et al., Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project. in ICCHP-9th Intl Conf. on Computer Helping People with Special Needs. 2004. Paris, France.
8. Ferreira, H., et al., AudioMath: Towards Automatic Readings of Mathematical Expressions. in Human-Computer Interaction Intl (HCII). 2005. L. V., USA.
9. Moço, V. et al., VICKIE: A Transcription Tool for Mathematical Braille. in AAATE-Asso. for the Adv. of Assistive Tech. in Europe. 2003. Dublin, Ireland.
10. Schwebel, F., et al., 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.
11. Batusic, M., et al., A Contribution To Making Mathematics Accessible For The Blind. in International Conf. on Computers Helping People with Special Needs. 1998. Oldenbourg, Wien, München.
12. Karshmer, A.I., et al., Reading and writing mathematics: the MAVIS project, in Proceedings of the third international ACM conference on Assistive technologies. 1998, ACM Press. p. 136-143.
13. Nemeth, A., The Nemeth Braille Code for Mathematics and Science Notation. American Printing House for the Blind, 1972.
14. Epheser, H., et al., Internationale Mathematikschrift fur Blinde, Deutsche Blindenstudienanstalt. Marburg (Lahn) 1992.
15. Commission Evolution du Braille Francais. Notation Mathematique Braille, mise à jour de la notation mathematique de 1971. 2001.
16. Edwards, A.D.N., et al., Lambda: a Multimodal Approach to Making Mathematics Accessible to Blind Students. in Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility. 2006. Portland, Oregon, USA.
17. Raman, T.V., Audio System For Technical Readings 1994, Cornell University.
18. Awde, A., et al., Information Access in a Multimodal Multimedia Computing System for Mobile Visually-Impaired Users, in ISIE 2006, IEEE Intl. Symp. on Industrial Elec. 2006: Montreal, Canada.
19. Weiss, G., Multiagent systems. MIT-Press, 1999.
20. Ferber, J., Les systemes multi-agents, ed. V.u.i. collective. 1995, Paris: InterEditions.
21. Jennings, N.R., et al., Applications of Intelligent Agents, in Agent Technology: Foundations, Applications, and Markets, 1998, Springer-Verlag: Heidelberg, Germany. p. 3-28.
22. Awde, A., et al., A Paradigm of a Pervasive Multimodal Multimedia Computing System for the Visually-Impaired Users, in The First International Conference on Grid and Pervasive Computing. 2006: Tunghai University, Taichung, Taiwan.
23. Sousa, J.P., et al., Task-based adaptation for ubiquitous computing. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 2006. 6(3): p. 328-340.
24. Jacquet, C., et al. A Context-Aware Locomotion Assistance Device for the Blind. in ICCHP 2004, 9th International Conference on Computers Helping People with Special Needs. Université Pierre et Marie Curie, Paris, France. .
25. Ross, D.A., et al., Talking Braille: A Wireless Ubiquitous Computing Network for Orientation and Wayfinding in Proceedings of the 7th International ACM SIGACCESS Conf. on Computers and Accessibility 2005. Baltimore, MD, USA ACM Press.
26. Mitchell, T., Machine Learning. 1997, McGraw-Hill.
28. Utgoff, P., ID5: An incremental ID3, in 5th Intl. Workshop on Machine Learning. 1988. p. 107-120.
29. Kallenberg, O., Foundations of Modern Probability, ed. S.S.i. Statistics. 2002: Springer. 650.
30. Witten, I.H. and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. 2005, San Francisco: Morgan Kaufmann. 525.