A Conceptual Framework for Developing Adaptive Multimodal Applications

Carlos Duarte, Luís Carriço
LaSIGE – Dept. Informatics, Faculty of Sciences, University of Lisbon
Edifício C6, Campo Grande, 1749-016 Lisboa, Portugal
cad@di.fc.ul.pt, lmc@di.fc.ul.pt

ABSTRACT

This article presents FAME, a model-based Framework for Adaptive Multimodal Environments. FAME proposes an architecture for adaptive multimodal applications, a new way to represent adaptation rules - the behavioral matrix - and a set of guidelines to assist the design process of adaptive multimodal applications. To demonstrate FAME's validity, the development process of an adaptive Digital Talking Book player is summarized.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: User Interfaces; D.2.2 [**Software Engineering**]: Design Tools and Techniques—*User interfaces*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human factors*

General Terms

Design, Human Factors

Keywords

IUI design, Adaptive multimodal interfaces, Behavioral matrix, Digital talking books

1. INTRODUCTION

Multimodal interaction has the potential to make interacting with a computer similar to what people are used to when interacting with others. By using speech input and output, gaze tracking, visual recognition and other technologies, multimodal interaction allows the user to interact in a more natural way. In addition, by allowing the user to employ the most appropriate modality for current conditions, it is easier to interact anytime, anywhere. This represents an increased accessibility, reaching wider audiences and situations of usage. Multimodal interaction also imposes less cognitive load than traditional interaction methods [21, 15], opening up new perspectives for educational applications, as well as decreasing task completion time and effort.

IUI'06, January 29–February 1, 2006, Sydney, Australia.

The advantages of multimodal interfaces can be better explored by introducing adaptive capabilities. These are important when dealing with users with different physical and cognitive characteristics, goals, preferences and knowledge. Adaptive interfaces can also accommodate changes in environmental conditions, by selecting the most appropriate modality, and in the interaction platform, reacting to the introduction or removal of some devices [6]. With these added benefits the interface will also be better prepared for the challenges of anytime, anywhere interaction.

With a growing consensus around the utility and usefulness of adaptive multimodal interfaces [16], one of the currently relevant problems of this domain remains the development of those interfaces. Techniques used for the development of traditional GUIs are not applicable, or have limited used, due to the differences between GUIs and adaptive multimodal interfaces. To overcome this limitation, several frameworks and conceptual models were proposed over the last years [20, 3, 5, 2]. However, most of these are limited to dealing with particular aspects of the development of multimodal interfaces, like the integration of specific modalities, with only a few proposing a global view over the development process. Even rarer are approaches that introduce adaptation related aspects into the multimodal application development process.

In this article we introduce FAME, a model-based Framework for Adaptive Multimodal Environments. The framework expands on previous frameworks and models, capturing the process of adaptive multimodal interface analysis. The framework's objective is to guide the development of adaptive multimodal applications. FAME is not intended to be a tool for automatic application development. It supports the development process by providing a conceptual basis that relates the different aspects of an adaptive multimodal system, and a set of guidelines for conducting the development process.

FAME proposes an architecture for adaptive multimodal applications. The architecture uses a set of models to describe relevant attributes and behaviors regarding user, platform and environment. The information stored in these models, combined with user inputs and application state changes, is used to adapt the input and output capabilities of the interface. To assist in the adaptation rules development, the concept of behavioral matrix is introduced. The matrix reflects the behavioral dimensions in which a user can interact with an adaptable component. A set of guidelines systematizes the development process, with each phase of analysis building upon previous phases.

FAME's applicability is demonstrated by describing the development process of an adaptive Digital Talking Book (DTB) player. The DTB player supports interaction through voice, keyboard and mouse inputs, and audio and visual outputs. The player adapts to devices' properties, reproduction environment, and users' behaviors and physical characteristics, thus making it a particularly adequate application for exploring FAME's flexibility.

The remainder of this article is organized as follows. The next section describes FAME's architecture. Section 3 introduces the concept of behavioral matrix. In the following section the guidelines are presented and illustrated with a simple example. Section 5 demonstrates the feasibility of FAME by describing the development process of an adaptive DTB player. Section 6 provides a brief overview about some of the existing frameworks and conceptual models for multimodal and adaptive interfaces development. Section 7 concludes the article.

2. FAME'S ARCHITECTURE

FAME's architecture establishes a general base for the development of adaptive multimodal applications. Building upon the benefits of a model-based approach [18], FAME's architecture uses the information stored in several models for controlling the multimodal outputs and presentation layout of the interface, but also the interaction possibilities available to the user, and how they are interpreted by the platform. FAME's architecture is shown in figure 1.

Two levels are identified in FAME's architecture. The inner level, or adaptation module, comprised of the different models and the adaptation engine, is responsible for updating the models and generating the system actions. The outer level, corresponding to the multimodal application layer, is responsible for the multimodal fusion of user inputs, transmitting the application specific generated events to the adaptation core, executing the multimodal fission of the system actions, and determining the presentation's layout.

The adaptation is based in three different classes of inputs: first, user actions, issued from any of the available input devices; second, application generated events and device changes, responsible for changing the state of any of the components the user interacts with; third, environmental changes, acquired by sensors, that impact the user's perception and operation of the application.

The other information with direct influence over the adaptation is stored in the different models:

User Model - This model stores relevant user preferences and characteristics. The model may include physical attributes such as user's visual impairment level, and preferences describing the preferred interface behavior and presentation characteristics.

Platform & Devices Model - The Platform & Devices Model describes the characteristics of the execution platform and of the devices attached to it. Platform attributes relate to invariant characteristics of the platform, and include, for instance, screen size. Devices represent the software and hardware artifacts that can be present or absent in the platform.

Environment Model - This model describes the environmental characteristics that can have an impact on the presentation and interaction aspects of the application. An example is the ambient noise of the interaction environment influencing both speech input and output.

Interaction Model - This model describes the components available for presentation and interaction. Each component has a set of templates available, covering the broadest range possible of devices, user groups and environments. For instance, for a textual component, the visual presentation should be complemented with the possibility to present it audibly. Another set of higher level templates organize the available components into a presentable version of the interface.

The different models are updated in the following situations: 1) the user model is updated in response to behavior changes exhibited by the user; 2) the environment model is updated in response to sensors detecting changes in the environment; 3) the platform & devices model is updated when the system detects the enabling or disabling of interaction devices.

The adaptation module is also responsible for shaping the multimodal fusion and fission components. The adaptation module can influence the output of both modules by determining the weight of each modality and the patterns of integration in the fusion or fission processes. The multimodal fusion component is responsible for determining the intended user's action from the information gathered by the different input modalities available. The choice of the weights can be determined either from perceived user behavior or environmental conditions (in a noisy environment the weight associated with speech recognition can be lowered). The user model could also be used to influence the operation of the multimodal fusion, as it has been shown that different individuals have different patterns of multimodal integration [14]. Also, the output layout is decided, taking into account factors such as the output devices' capabilities and the users' preferences, characteristics and knowledge.

3. BEHAVIORAL MATRIX

The behavioral matrix, being a tool for defining and representing the adaptation rules, relates directly to the adaptation engine in FAME's architecture. Before describing the constituents of the behavioral matrix, the adaptable component will be defined. This central concept of the framework represents the interface components that are the target of adaptation. For instance, in a tourist information system, the content presented to the user might be adapted according to previously visited sites. This content presentation component is an adaptable component.

To help define and represent the adaptation rules, a behavioral matrix is employed. A behavioral matrix describes the adaptation rules, the past interactions, and how the rule's applicability evolves over time reflecting usage behaviors. For each adaptable component, one behavioral matrix is defined. The matrix dimensions reflect the behavioral dimensions governing the interaction with the specific component. For example, one of the dimensions may be the output modality, or combinations of modalities, used by the component to present information to the user. Other dimensions can describe how the information should be presented, what information to present and when to present it, for instance.

Each of the matrix's cells holds a tuple with up to three elements. The first element of the tuple defines the rules

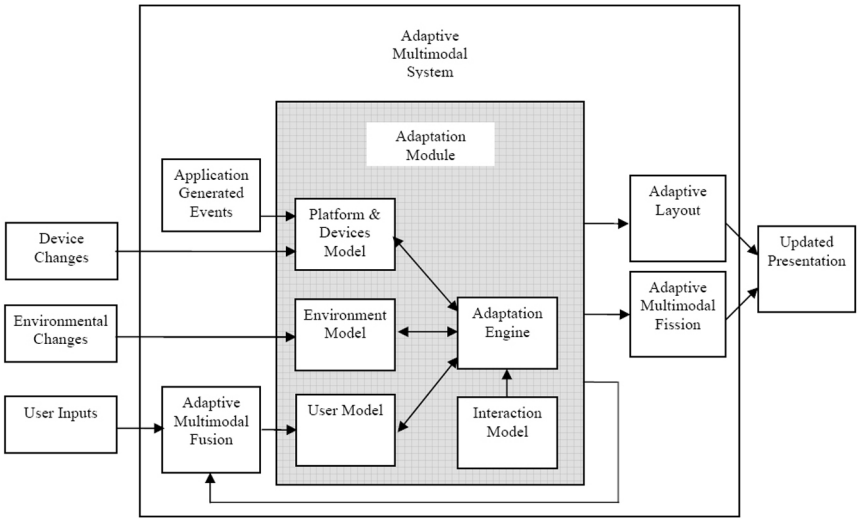

Figure 1: FAME's architecture.

currently in use. If the element is set, the rule defined by the behaviors associated with each of the matrix dimensions is currently in place and triggered whenever the activating events and conditions occur. If the element is not set, then the associated rule is not activated whatever the state of the triggering events. Some of the rules defined in the matrix will be mutually exclusive, according to their triggering events, while other rules may be triggered simultaneously.

The second element of the tuple counts the number of times the rule has been activated by the user's direct intervention. This value stores information about the user's preferred behavior, allowing for adaptation of the behavioral matrix itself.

The third element defines a threshold for rule activation. This threshold can be an absolute value, or defined by expressions that make use of values stored in the second element of all or some of the tuples. In the first case, the values can relate directly to cognitive or perceptual factors of the user. In the second case, the expressions can use the observed user behavior to improve the interface adaptation. This last possibility supports the adaptation of the behavioral matrix itself, by selecting what rules are active based on past user behavior.

The behavioral matrix can be used only as an analysis tool, improving the design of the adaptation rules by helping to understand the relations between the evolving behavior of the users, and the interaction between the available modalities. However, if the developer wishes, it can be implemented in the adaptive multimodal application, controlling the evolution of the adaptation rules, thus contributing to a better fit between users and usage conditions and the interaction capabilities of the application, by selecting the adaptation rules in use.

4. GUIDELINES

In this section we provide a set of guidelines for the development of adaptive multimodal applications, and, accompanying their description, a simple example shows how they can be applied. The example is based on a tourist information system, capable of providing information about user selected sites. Let us suppose the system supports different input and output modalities, such as voice, a pointing device, a location aware device, audio, text and graphics. The goal of presenting the example is just to help the reader in understanding how the architecture and guidelines are used. It is not our intention to provide here a thorough description of the development of an adaptive multimodal application.

FAME's guidelines for the development process are: 1) Identification of the adaptation variables according to the categories presented in the architecture; 2) Identification of the adaptable components; 3) Selection of the attributes for the user, environmental and the platform & devices models; 4) Template design for the interaction model; 5) Definition of the multimodal fusion and fission operations; 6) Definition of the adaptation rules for each adaptable component, using a behavioral matrix. The development process is illustrated in figure 2, where the dependencies between the different steps are illustrated.

The outputs of the steps presented in figure 2 relate to the components of the architecture presented in figure 1. The various models and the fusion and fission operations translate directly between the two figures. The Adaptation Variables in the development process translate to the User Inputs, Environmental Changes, Device Changes and Application Generated Events in the architecture. The Behavioral Matrix translates to the Adaptation Engine. The Adaptable Components impact several elements of the architecture: the Interaction Model, the Adaptation Engine and the Adaptive Layout.

A more detailed explanation of the various steps, along with the application of these guidelines to the tourist information system example, is now presented. Step 1 consists of the identification of the adaptation variables. These would include: every user input, measurements of the ambient noise, measurements of the user current position, events generated by the system to warn the user to the proximity of interesting sites, and events generated by the system to guide the user to desired points of interest.

Step 2 identifies the adaptable components. For the current example, at least two adaptable components can be

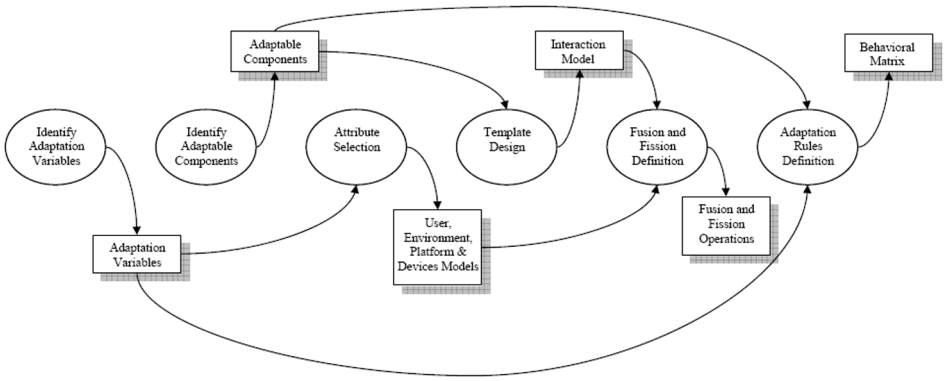

Figure 2: FAME's development process.

identified: a navigation component, with the purpose of providing directions to points of interest, and a site description component, designed to provide individualized information about specific sites of interest.

These components should be analyzed according to three dimensions. 1) Interaction, references the input and output modalities available, their status (enabled or disabled), and their use (cooperatively or individually); 2) Content is related to enhancements and alternative presentations using available sounds, images, or other media; 3) Presentation deals with size and color of fonts used, placement of visual components, type of audio signals used, synchronization units, etc.

The analysis results in a set of behavioral dimensions for each component. For example, for the navigation component, the following dimensions can be devised: from the interaction analysis, input modality (voice recognition and pointing device), input combination (on and off), output modality (voice synthesizing and graphical presentation) and output combination (on and off); from the content analysis, instruction types (simple and detailed); and from the presentation analysis, voice synthesizing type (female and male) and size of font (small, normal and large), amongst others.

Step 3 deals with the selection of the models' attributes. Some of the models' attributes will be equivalent to input variables, while others have to be derived from those, and others yet will have to be acquired by other means [12]. In the tourist information example, user model's attributes may include user preferences about tourist sites, historic periods and characters, and also about interaction preferences. Other attributes may describe previously visited sites. Information about physical characteristics, like hearing or visual impairments can also be relevant to the application. The environment model's attributes should contain information about relevant characteristics of the execution environment, such as ambient sound, and indoor or outdoor localization of the platform. The attributes of the platform & devices model should contain information like the characteristics of the display devices (size, resolution, colors) and properties of the speech recognition and synthesis devices.

In step 4 the various templates comprising the interaction model should be designed. Two types of templates are defined. The first type, named component template, defines how to present a component. The second type, named composite template, defines relationships between components in order to be possible to group components into a presentation. Composite templates can be a combination of two or more component or composite templates.

For each of the interactive components of the tourist information system, the designer should develop a set of templates, considering the input and output devices available. For example, for the site description component, a composite visual template could be composed by three component templates for presenting pictures, a textual description and an overview map. The component templates define how their information is to be presented, and the composite template defines how the component templates share the screen space and relate to each other during the presentation. Each of the component templates should have presentations available using more than one modality, namely visual presentation and audio narrations, and the composite template can adapt the presentation using the different combinations available.

Step 5 consists of the definition of the multimodal fusion and fission operations. In our example, fusion of input modalities will be needed, in order to integrate input from the pointing device with input from the speech recognition module. Output fission will also be performed for presentation of information in several formats. Details of fusion and fission processes are outside the scope of this example.

The sixth and final step is the definition of the adaptation rules. To design the adaptation rules, the designer defines a behavioral matrix for each of the previously identified adaptable components. The behavioral matrix's dimensions have also been previously identified during the interaction, content and presentation aspects analysis.

The construction of a behavioral matrix will be exemplified for the navigation component. The dimensions can be devised from the previous analysis: an inputs dimension, taking the values "voice", "pointing" and "both" combines the previous dimensions input modality and input combination. Similarly, we have an output dimension with the values "voice", "graphical" and "both". Other dimensions and their values include: instructions type ("simple" and "detailed"), voice type ("female" and "male") and font size ("small", "normal" and "large"). In figure 3 we present an example of a matrix with only the instructions type and output dimensions represented, since visually presenting all the matrix's dimensions is not feasible.

This matrix encodes rules to relate the output modality and the instruction type. Voice synthesis should be used with simple instructions (first value of the tuples in

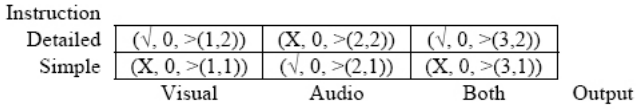

Figure 3: A behavioral matrix for two dimensions of the navigation component in the tourist information system example.

both rows of the second column), while detailed instructions should be presented using visual or a combination of visual and audio modalities (first value of the tuples in both rows of the first and third columns). The matrix also encodes updates to the interface behavior as a result of observed user behavior. This can be seen in the third value in the tuple of the first column, second row (changing behavior to presenting simple instructions with visual output) and in the third value in the tuple of the first column, first row (changing behavior to presenting detailed instructions with visual output). Rules similar to the last two, referencing the use of the voice and of both output modalities at the same time are also encoded. In this fashion, rules can be constructed that define how the behavior of the application evolves by observing the user's behavior.

The guidelines presented here do not have to be followed in this precise order, and not all steps are mandatory (for instance, an application with only visual output can ignore multimodal fission). However, they will help the development of adaptive multimodal applications, even if not using the proposed FAME architecture.

5. USING FAME ON A DTB PLAYER

In this section we summarize the development of an adaptive Digital Talking Book (DTB) player. The development process was based on the FAME guidelines presented above, and shows the feasibility of the framework for developing adaptive multimodal applications. The player was intended for a PC-based platform, sacrificing mobility for greater processing power and the possibility to plug-in more support devices and sensors. The platform's input devices consisted in a keyboard, a mouse and a microphone. Speech recognition software was present. For output, a visual display and sound output were used.

DTBs, as described in the ANSI/NISO z39.86 standard[1], are the digital counterpart of talking books, which have been available for many years to print-disabled readers. The essential sets of features of a DTB include: no need to use visual display to operate device; variable playback speed; document accessible at fine level of detail; usable table of contents; easy skips (moving sequentially through the elements); ability to move directly to a specific target; setting and labeling bookmarks; ability to add information (highlighting and notes); presentation of visual elements in alternative formats (speech); etc. In addition to these features the adaptive DTB player was designed to support DTBs with complementary media content, like background music, ambient noises, video clips and any other media capable of complementing the original work, and with the potential to provide a more entertaining experience to the reader.

[1]Available at http://www.niso.org/standards/resources/Z39-86-2002.html

As presented in section 4 the first step of the development is the determination of the adaptation variables. For a PC-based DTB player the adaptation is mostly a consequence of the behavior of the user and of events resulting from the book's presentation itself, than of changes to the environment. Even so an "ambient noise" variable was considered. The adaptation variables related to user actions include every action that directly influences the playback and presentation of the book, like changing the narration speed, or altering the placement of visual elements. Application generated events that initiate adaptation include all events that are part of the author defined presentation (for instance, the presentation of images may trigger the rearrangement of the visual elements of the interface) and events signaling the presence of user created markings.

The next step consists of the identification of the adaptable components. These were mapped to the elements of a DTB: Book Content, Table of Contents, Annotations and other Miscellaneous content, including tables, images and side notes. The interaction, content and presentation analysis determined the following dimensions. For the Miscellaneous component: *action*, *visibility*, *modality* and *reading*. The Annotations component uses the same four dimensions and adds two new dimensions: *reaction* and *default content*. The Table of Contents component uses the *modality*, *visibility* and *reading* dimensions already presented, and introduces a new dimension, *presentation*. Finally, the Book Content component, repeats the *modality* dimension, and introduces six new ones: *synchronization*, *speed*, *marking presentation*, *marking presentation modality*, *reading path* and *reading path content*. The final two dimensions are related to alternative reading pathways.

The values of each dimension are not presented due to space constraints. Some will be presented later in the article when illustrating the behavioral matrixes.

Step 3 follows with the selection of the models' attributes. The User model possess attributes for identifying the user, in order for the system to be able to store the annotation's authors, and attributes characterizing the user's visual impairment level. The Environment model's only attribute stores samples of the adaptation variable "ambient noise". The Platform & Devices model stores the characteristics of the devices found in the execution platform, including, for instance, the screen resolution and the features of the speech synthesis and recognition modules.

The design of a set of templates for the Interaction model is the next step. Each adaptable component has at least two component templates: one template for visual presentation and another for audio presentation. For the Miscellaneous component one template should be available for each of the different types of content to present. For example, when visually presenting an image, the template specifies that an image title is presented above the image, and the image caption is presented bellow the image. When presenting the image using speech, another template specifies that the title is followed by the caption and a previously recorded image description. A Book Content template defines how the synchronization is presented visually, margin sizes, fonts and other presentation details.

The presentation of all the book's components is handled by a set of composite templates. These templates are responsible for deciding the placement of each of the components, taking into account how many and what components

are currently displayed. The user can act upon the presentation of the different components by moving or resizing them. The adaptation engine stores these user preferences and uses them to decide which composite templates are selected, and also to adapt the results of these templates in order to match the expressed user preferences.

The next step is the definition of the multimodal fusion and fission operations. Currently, fusion is used to combine speech input with the pointing device and the presentation's context, whenever a speech command to show or hide a component is issued by the user. The fission operation is responsible for guaranteeing synchronization between the visual presentation and the audio narration of the Book Content component, and also for handling situations where the user needs to be alerted (for instance, to the presence of an image). In such cases, more than one modality can be used to alert the user, with the fission operation deciding upon the most appropriate actions and modalities to use.

The final step is the definition of the adaptation rules. For each of the four adaptable components one behavioral matrix was defined, using the behavioral dimensions and values previously presented. The designer's role at this point is filling the matrix's tuples. Next, examples of rules encoded in the behavioral matrixes are presented.

Figure 4 shows the relation between two of the dimensions of the miscellaneous component behavioral matrix. According to the rules encoded in the behavioral matrix, if the content is displayed using visual output then the main content narration continues. If the content is displayed using audio output then the main content narration pauses. The main content narration also pauses when both audio and visual outputs are used. In this fashion the overlap of two different audio tracks is prevented. This behavior may also change if the user behavior reflects different preferences.

Reading	Visual	Audio	Both
Continue	(√, 4, >(1,2))	(X, 0, >(2,2))	(X, 1, >(3,2))
Pause	(X, 2, >(1,1))	(√, 5, >(2,1))	(√, 3, >(3,1))

Modality

Figure 4: Two dimensions of the behavioral matrix for the miscellaneous component.

Figure 5 shows the relation between two dimensions of the annotations behavioral matrix. This shows that if the preferred user behavior is to have the annotations presented whenever they are reached during the narration, they should be presented using visual output. If, on the other hand, the user prefers to be alerted to the presence of an annotation, but not have the annotation shown immediately, then both output modalities should be used to warn the user.

Visibility	Visual	Audio	Both
Show	(√, 0,)	(X, 0,)	(X, 0,)
Alert	(X, 0,)	(X, 0,)	(√, 0,)
Ignore	(X, 0,)	(X, 0,)	(X, 0,)

Modality

Figure 5: Two dimensions of the behavioral matrix for the annotations component.

The behavioral matrix for the book content component, displayed in figure 6, relates the synchronization and speed dimensions. The synchronization unit grows (from word, to sentence, to paragraph) according to the reading speed (from slow, to normal, to fast). The definition of what is a slow, normal or fast reading speed may take into account the characteristics of the speech output component, but also cognitive and sensorial characteristics of the user.

Synchronization	Slow	Normal	Fast
Word	(√, 0,)	(X, 0,)	(X, 0,)
Sentence	(X, 0,)	(√, 0,)	(X, 0,)
Paragraph	(X, 0,)	(X, 0,)	(√, 0,)

Speed

Figure 6: Two dimensions of the behavioral matrix for the book content component.

5.1 Features of the DTB player

Figure 7 shows an instance of the player with all the components visible. Two ways of situating the reader can be perceived in the figure. First, the visual synchronization marker highlights the word being narrated (word with grey background). Second, the current section or chapter number is also highlighted (text in red) in the table of contents. Also depicted are marks on text annotated by the user (text with a green background).

One of the player features is the possibility to customize and adapt the disposition of the presented components. The user can move any component to a new position, and the player will rearrange all the other components' positions automatically. This adaptive behavior can also be triggered by an automatic presentation of a previously hidden component, which can happen, for example, when the narration reaches a point where an image should be presented.

The synchronization unit between highlighted text and narrated audio can be automatically set by the adaptation module, in response to user actions. A triggering action is the selection of a new narration speed. The increase in narration speed will move the highlight from word to word faster. A speed will be reached where it will be perceptually difficult to accompany the highlighted word. Recalling the behavioral matrix for the book content presented earlier, the adaptation engine will try to diminish this effect by increasing the synchronization unit as the speed rises. Other events adapting the navigation unit are free jumps in the book, resulting from searches. The navigation unit is chosen taking into account the distance between the starting and ending points of the jump. The reasoning behind this adaptation is that the greater the distance, the bigger the difference in context.

The interface behavior is also adapted in response to the user's behavior relating to the presentation of annotations and miscellaneous content. The default initial behavior is to alert the user to the presence of such content, without displaying it. If the user repeatedly ignores such alerts then the interface's behavior is changed in order to stop alerting the user. This is the "ignore" value of the *action* dimension of the annotation and miscellaneous components' behavioral matrixes. If the user behavior is to acknowledge the alerts and consult the annotations or the miscellaneous content, then the interface's behavior becomes presenting the content without alerting the user. This is the "show" value for the *action* dimension of those matrixes.

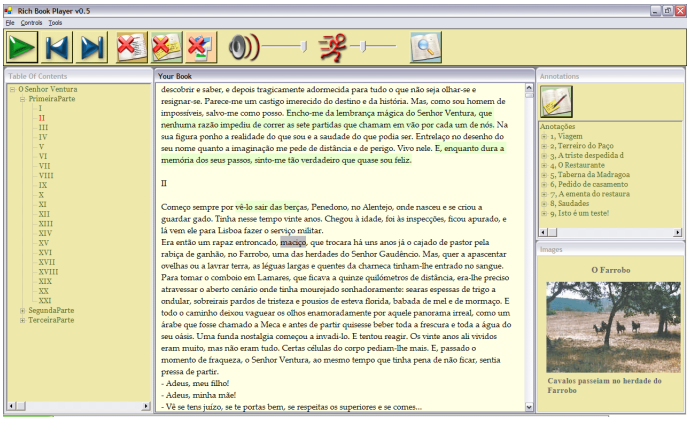

Figure 7: The DTB player visual interface, presenting main content, table of contents, list of annotations and an image.

Another feature of the player aims particularly the reading of technical and reference works. This feature concerns text re-reading of highlighted parts. When reading technical works, the reader usually underlines relevant passages of the text, sometimes using different colors or marking styles, in order to convey different relevance levels or categories. In a later re-reading the reader attention is usually focused on those passages. The player supports this functionality by allowing the reader to annotate the text and categorize the annotations. From these categorizations several behaviors can be devised for further readings of the same book: reading of only the annotated material; reading material of only specific categories; association of different reading speeds to different categories. A possibility opened up by this feature is the user creation and reading of text trails that may constitute content perspectives, sub stories, argumentation paths, etc.

6. RELATED WORK

Multimodal and adaptive interfaces, with their unique characteristics, are still to achieve the same degree of support for application development that standard interfaces have reached. The majority of current approaches either address specific technical problems, or are dedicated to specific modalities. The technical problems dealt with include multimodal fusion [7, 6], presentation planning [6, 11], content selection [9], multimodal disambiguation [13], dialogue structures [1] or input management [5]. Platforms that combine specific modalities are in most cases dedicated to speech and gesture [17, 19]. Other combinations include speech and face recognition [8] or vision and haptics [10]. Even though the work done in tackling technical problems is of fundamental importance to the development of adaptive and multimodal interfaces, it is of a very particular nature, and not suited for a more general interface description. The same can be said of specific modality combinations, where some of the contributions do not generalize for other modality combinations, due to nature of the recognition technologies involved.

Still, frameworks exist that adopt a more general approach to the problem of multimodal interface development. The ICARE project [2], a framework for rapid development of multimodal interfaces, shares some concepts with the framework presented here. The ICARE conceptual model includes elementary and modality dependent components as well as composition components for combining modalities. ICARE is valuable for development of multimodal interfaces, however it targets platforms that do not include adaptation capabilities, and thus leaves adaptation related details out of the development process.

Frameworks to support adaptation can also be found. For example, adaptable hypermedia systems have been developed over the last years, leading to the conception of models for adaptive hypermedia, like the AHAM model [4]. However, these models are usually targeted for platforms with very specific characteristics, with a greater focus on content and link adaptation for presentation purposes, completely ignoring the requirements of multimodal applications. Outside the hypermedia field we can find, for instance, the Unified User Interface (UUI) methodology [20], which argues for the inclusion of adaptive behavior from the early stages of design. This methodology aims to improve interface accessibility by adapting to the user and context. The UUI framework is based on a unified interface specification and an architecture for implementation of the dialogue patterns identified. Although supporting interface development for multi-device platforms, this methodology does not deal with the specificities of multimodal characteristics.

7. CONCLUSIONS

This article presented FAME, a conceptual framework for the development of adaptive multimodal applications. FAME is intended to assist developers of adaptive multimodal applications during the analysis and design process. It is not a tool for automatic interface generation. FAME proposes a model-based architecture for adaptive multimodal systems, designed to adapt to user actions, system events, and environmental changes. The adaptation rules for each adaptable component are encoded in a behavioral matrix. Besides storing information about the adaptation rules, the matrix also stores information about past user interactions, which can be used, in conjunction with other information about the user and the platform, to replace adaptation rules. In this way it is possible to provide a better fit to user and usage conditions. FAME also includes a set of guidelines to assist the developer during the whole process.

To demonstrate the applicability of FAME, we successfully developed an adaptive multimodal DTB player. The DTB player uses voice commands as well as keyboard and mouse commands as input modalities. For output, visual and audio presentations are synchronously available. The player is capable of adapting to the user's behavior, and also to changes in environmental conditions, namely the ambient noise.

Future work will focus on two main directions. One direction will be the development of a DTB player for a non-PC based platform. This will allow us to understand how much of the FAME's acquired knowledge for one platform is reusable when developing a very similar application on a platform with different characteristics. This new platform will allow us to experiment more with environment and location based adaptation. The other direction will be the formalization of the operations allowed over the behavioral matrix, to improve the translation process from the concepts expressed in the behavioral matrix to the final application developed.

8. ACKNOWLEDGMENTS

The work discussed here is based on research supported by FCT (Fundação para a Ciência e Tecnologia) through grant POSC/EIA/61042/2004.

9. REFERENCES

[1] E. Blechschmitt and C. Strödecke. An architecture to provide adaptive, synchronized and multimodal human computer interaction. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 287–290, New York, NY, USA, 2002. ACM Press.

[2] J. Bouchet, L. Nigay, and T. Ganille. Icare software components for rapidly developing multimodal interfaces. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 251–258, New York, NY, USA, 2004. ACM Press.

[3] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, N. Souchon, L. Bouillon, M. Florins, and J. Vanderdonckt. Plasticity of user interfaces: A revised reference framework. In *Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design TAMODIA'2002*, pages 127–134, Bucharest, Romania, 2002.

[4] P. De Bra, G.-J. Houben, and H. Wu. AHAM: a dexter-based reference model for adaptive hypermedia. In *HYPERTEXT '99: Proceedings of the tenth ACM Conference on Hypertext and hypermedia : returning to our diverse roots*, pages 147–156, New York, NY, USA, 1999. ACM Press.

[5] P. Dragicevic and J.-D. Fekete. The input configurator toolkit: towards high input adaptability in interactive applications. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 244–247, New York, NY, USA, 2004. ACM Press.

[6] C. Elting, S. Rapp, G. Möhler, and M. Strube. Architecture and implementation of multimodal plug and play. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 93–100, New York, NY, USA, 2003. ACM Press.

[7] F. Flippo, A. Krebs, and I. Marsic. A framework for rapid development of multimodal interfaces. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 109–116, New York, NY, USA, 2003. ACM Press.

[8] A. Garg, V. Pavlović, and J. Rehg. Boosted learning in dynamic bayesian networks for multimodal speaker detection. *Proceedings of the IEEE*, 91(9):1355–1369, 2003.

[9] D. Gotz and K. Mayer-Patel. A general framework for multidimensional adaptation. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 612–619, New York, NY, USA, 2004. ACM Press.

[10] M. Harders and G. Székely. Enhancing human-computer interaction in medical segmentation. *Proceedings of the IEEE*, 91(9):1430–1442, 2003.

[11] C. Jacobs, W. Li, E. Schrier, D. Bargeron, and D. Salesin. Adaptive grid-based document layout. *ACM Trans. Graph.*, 22(3):838–847, 2003.

[12] A. Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(1-2):49–63, 2001.

[13] S. Oviatt. Mutual disambiguation of recognition errors in a multimodel architecture. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 576–583, New York, NY, USA, 1999. ACM Press.

[14] S. Oviatt. User-centered modeling and evaluation of multimodal interfaces. *Proceedings of the IEEE*, 91(9):1457–1468, 2003.

[15] S. Oviatt, R. Coulston, and R. Lunsford. When do we interact multimodally?: cognitive load and multimodal communication patterns. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 129–136, New York, NY, USA, 2004. ACM Press.

[16] S. Oviatt, T. Darrell, and M. Flickner. Multimodal interfaces that flex, adapt, and persist. *Commun. ACM*, 47(1), 2004.

[17] S. L. Oviatt, P. R. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *Human Computer Interaction*, 15(4):263–322, 2000.

[18] F. Paternò. *Model-Based Design and Evaluation of Interactive Applications*. Springer-Verlag, 1999.

[19] R. Sharma, M. Yeasin, N. Krahnstoever, I. Rauschert, G. Cai, I. Brewer, A. M. Maceachren, and K. Sengupta. Speech-gesture driven multimodal interfaces for crisis management. *Proceedings of the IEEE*, 91(9):1327–1354, 2003.

[20] C. Stephanidis and A. Savidis. Universal access in the information society: Methods, tools and interaction technologies. *Universal Access in the Information Society*, 1(1):40–55, 2001.

[21] C. Wickens and J. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, 1999.