MMWA-ae: boosting knowledge from Multimodal Interface Design, Reuse and Usability Evaluation

Americo Talarico Neto, Renata Pontin M. Fortes, Rafael Rossi and Solange Rezende
Av. Trabalhador Saocarlense,400. São Carlos, SP, Brazil
University of São Paulo
E-mail: {americo@, renata@, solange@, ragero@}icmc.usp.br

Abstract

The technological progress designing new devices and the scientific growth in the field of Human-Computer Interaction are enabling new interaction modalities to move from research to commercial products. However, developing multimodal interfaces is still a difficult task due to the lack of tools that consider not only code generation, but usability of such interfaces. In this paper, we present the MultiModal Web Approach and its authoring environment, whose main goal is boosting the dissemination of project knowledge for future developments benefited by the solutions to recurring problems in that multimodal context.

1. Introduction

The support for various interaction modalities has become a mandatory requirement for the next generation interfaces with the growing proliferation of computing devices, the increasing availability of Web services to the worldwide population and due to the great expressive power, naturalness and portability that multimodal interfaces offer the users to perform their daily tasks [9]. However, the use of combined modalities such as voice, touch and graph, raises a number of usability and interaction issues that the project team is faced with, like syncronization and integration requirements and constraints that should be considered during the design phases[5].

As a result, the designers are exposed to an ever-increasing challenge: despite of learning the novel languages and accessible technologies and applying them to obtain the multimodal application code, they should realize which are the best practices in this research field and how to experiment the designed interfaces with real users with different behaviors in a fast manner prior to the product release. In addition, the project team should be concerned on how to promote reuse of Design Rationale (DR) and application code in order to decrease the cost and effort implementing multimodal interfaces.

In this paper, we endeavor to answer these questions and concentrate on solving such issues by presenting the MultiModal Web Approach (MMWA), which relies on the theoretical and empirical expertise acquired in the design process. Such expertise is documented in the form of Design Rationale (DR), Design Principles and Design Patterns which can be shared and applied by the design team, during multiple iterations to refine the interface or by designers of different applications.

Moreover, based on our experience applying the MMWA in real projects [8] and the observations of the design workflow of MMWA, we have designed an authoring tool that guides the project team, through the MMWA steps and activities. The advantage of this authoring environment, MMWA-ae, is that it suggests design alternatives based on

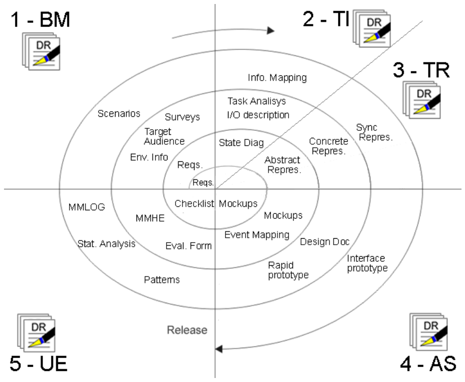

Figure 1. MMWA design process

previous collected DR [8] and the well-known interaction Design Patterns [2] extended to support the multimodal theory and it also implements the previous identified principles and checklists [8] so that the designers are presented with the successful solutions for recurring problems in this context together with their rationale.

This paper is structured as follows: Section 2 details the MMWA. Section 3 describes the case studies that illustrate how MMWA-ae was applied, sharing the most significant results obtained. Section 4 presents a bibliographic review of design methods for multimodal interfaces and their main issues. Conclusions and future work are discussed in Section 5.

2. The MultiModal Web Approach and its authoring environment

The MultiModal Web Approach (MMWA) [8], Figure 1, provides designers with a more practical framework to design, develop and evaluate multimodal projects. It follows a spiral process composed of 5 activities that allow the capture and retrieval of DR, which provides successful solutions for recurring problems in the multimodal interaction context, decisions, justifications, alternatives and arguments that led to the final design, documenting them for future reuse. As a result, it optimizes the methodology for developing interfaces and enables the identification, analysis and discovery of multimodal interaction Design Patterns.

Behavioral Model (BM) consists of scenarios, constraints and information on the environment in which the multimodal tasks will be performed. BM aims to assist in identifying the information exchanged between the user and the system during the task execution. The goal is getting the task domain information that is necessary for selecting the appropriate tasks mapping and their multimodal interface representation. All such information and data are documented as part of the multimodal application Requirements Specification.

Task Identification (TI) and Representation (TR) identify the tasks to be performed by the user analyzing important task aspects such as: goals, initial states, activities and procedures involved, the problems that may occur, the task environment, the target audience and their experience, and the multimodal input and output. It creates abstract representations employing methods such as those described in [10] or [12]. Elements needed to generate and analyze any kind of multimodal input and output including their pros and cons are documented in a DR form. The idea is representing the result as a list of interfaces elements for each candidate input and output modality and modalities combinations. Concrete representations are used to assign GUI and VUI widgets and the methods required to trigger the transitions as well as the events that can be activated in each interface element.

Analysis of Solutions (AS) uses earlier captured DR and filters out the interaction options based on specialists argumentations. It provides the potential modalities to perform the task with their DR. At the end, a design document is obtained and it is evaluated in the next activity.

An elementary step in the development process for any multimodal interface involves **usability evaluation (UE)** of prototypes based on qualitative studies; in MMWA it is represented as the Heuristic Evaluation mechanism (**MMHE**), and authentic user interactions. A main feature of the MMWA is the web-based usability evaluation mechanism that is intended to provide data for problems identification, analysis and correction in early stages of the project; in MMWA it is represented as a mechanism for automatic logging generation and analysis in remote User Tests (**MMLOG**). A benefit of web-based usability studies is that users can be recruited from all over the world and can interact whenever is most convenient.

The **MMWA authoring environment (MMWA-ae)** was developed to automate the MMWA activities and serve as a framework for developing multimodal Web interfaces using voice, touch and graph. With the aid of MMWA-ae the designer specifies the multimodal interaction using Task Analysis as a foundation. It is possible to make use of Design Patterns and DR to resolve design questions and obtain the rationale to solve recurrent problems with the help of solutions used in previous projects in the same context.

The use of DR can be time-consuming, disruptive and costs excessive. To solve this constraint our approach extracts and captures DR information when it is needed. The MMWA-ae holds an abstract mechanism for capturing and storing DR allowing organizations and developers to use any WIKI systems main features to collaborate in the tasks specification, implementation and during tests activities.This mechanism also makes information organization achievable, so that future projects would benefit from efficient searches, ensuring that precise results are achieved.

Another important feature of MMWA-ae is the ability to generate multimodal interface prototype to be target of usability evaluations. For each task specified by the designer, the tool creates a code snippet that contains voice and graphical interface elements and includes functions to capture interaction events. It is possible to perform remote user testing deploying this interface in a Web environment, therefore evaluating the usability of the designed interface looking for problems that can be corrected before release to a final customer. This functionality also enables the user interaction patterns identification, by analysing charts and graphs, and design errors identification or areas of usability improvement In addition it covers the main limitations of user testing, which are: the difficulty of recruiting users, time and resources availability to perform the tests, since the whole process is done remotely.

Finally, MMWA automatically creates the design documentation, given that during the Task Analysis the designer is aided by a wizard, whose main goals are: to perform the analysis of solutions; to obtain the key usability principles to be included in the generated code and to ensure that the checklist available on MMWA is considered in the prototype generation. As future work we intend to include in MMWA a mechanism for the fusion of pre-existing graphical user interfaces with speech interfaces. Thus, the designer would spend less effort in the checklists verification and using the practices compiled in the form of DR during the process of specifying and creating multimodal Web interfaces.

3. Overview of Case Studies & Results

Up to now we have performed 3 case studies using the MMWA that permitted us to exhibit expressive results obtained using the approach, including: the validation of the MMWA cycle and its activities; the controlled mechanism to capture and store DR; and the MMHE and MMLOG mechanisms that allow for a better interaction design, reporting the usability problems in a DR format and the underlying rationale to fix them. A new case study was conducted in order to obtain a number of results on the topic of the authoring tool. The objectives were to determine quality metrics, to analyze the reutilization rate and to measure the usability of the generated source code. The same design used in a previous case study, a Car Rental System was used. The artifacts were obtained by two groups of designers: one following the MMWA approach supported by the MMWA-ae and the other group designing, developing and evaluating the interfaces without the support of the authoring tool.

In order to be able to measure **code reutilization** and to confirm that the **reuse** was increased by the use of MMWA-ae we employed the formula:

ReutilizationRate = [LOCreused + (LOCWhiteBox) - (LOCNonDesiredMethods)] / LOCtotal

Applying the reutilization rate formula in the case study we ended up with 83% of overall reutilization.

Moreover, the **time spent to design the interface and to obtain the prototype** for user testing is one of the main advantages of the MMWA-ae, given that the interfaces prototype is obtained right after the design phase is completed. In contrast, when MMWA-ae was not used a large numbers of tasks were performed before the prototype was obtained, like: DR database queries, team discussions to validate implementation, rounds of usability analysis prior to the final delivery, voice user interfaces grammar creation and testing, etc.

MMWA-ae offers the designer a Wizard to fill all the important prompts to be used in each voice interface. Furthermore by choosing one of the Design Patters, for example the Calendar pattern, the tool automatically creates the graphical and the voice interfaces, the grammar and synchronizes all of them together saving a huge amount of the development time. It also increases the usability of such interfaces because MMWA-ae includes an error recovery strategy mechanism and the well-know voice universal commands, allowing for a more robust multimodal interface [1].

MMWA-ae also promotes the **Design Documentation and Rationale storage and retrieval**. DR recorded in the case studies is intended to be a framework of the reasons behind decisions made when designing artifacts. The understanding of the justification for design decisions made throughout the design process is necessary in order to understand, recreate, reuse or modify the design.

As a result the MMWA-ae enables the use of DR: to verify the design decisions, whether it truly reflected what the designers planned based on the MMWA Behavior Model activity; to evaluate the design alternatives (MMWA Analysis of Solutions activity); to modify the design during the maintenance phases and to promote the Design Reuse; to promote Design Communication among people who are involved in the design process; to document the entire design process.

Another significant outcome obtained was the framework for Desing Patterns identification. Based on the DR collected in various projects, we have developed a mechanism that helps on building a Design Pattern based on repeated issues, recurring positions supported by well-built arguments.

Another important finding is regarding the usability of the MMWA-ae generated interfaces. By performing Heuristic Evaluations we were able to identify that most of the usability issues found during the first case study were not present in the user interface created by MMWA-ae. This could be achievable due to the MMWA-ae inherent DR knowledge and due to the Design Pattern suggestion mechanism, which allows the designers to choose the best design pattern to implement their ideas. Furthermore the usability principle built in the generated code also contributes to the overall usability increase. Comparing data from the three case studies, problems found applying the MMHE decreased from one case study to another due to the use of DR in subsequent projects and due to the use of Heuristic Evaluation prior to the User Tests, validating assumptions made in [6] regarding the complementary nature of those usability evaluation methods.

We also used **Association Rules** in our case studies. An association rule is rule of the type $A \Rightarrow B$ in which A and B are itemsets on the database, and $A \cup B = \emptyset$. The two classical measures to generate association rules are support and confidence. **Support** measures the joint probability of an items set in a database, that is, $sup(A \Rightarrow B) = n(A \cup B)/N$, in which $n(A \cup B)$ is the number of transactions that A and B occur together, and N is the total number of transactions. **Confidence** indicates the occurrence probability of A and B given the occurrence of A, that is, $conf(A \Rightarrow B) = n(A \cup B)/n(B)$.

The users interaction with the prototyped interface generated by MMWA-ae generates log files containing the users actions and system events, one containing all the iteractions in a row of the database, called BASE-1; and another file containing where a single iteraction with an interface element in represented in a row of the database, called BASE-2. There were 70 transactions in the BASE-1 and 1,190 transactions in the BASE-2. Using association rules and extracting modalities and tasks relationships like occurence and confidence, we intend to verify which modalities the users choose to perform a task in the system and in which tasks or moments they use a combination of modalities to interact. Besides that, we want to verify the most common errors, in which part of the interface they occur and how the modalities are used to recover from an error.

Some of the extracted rules using the BASE-1 are shown in Table 1. It can be noticed that in most of the interactions the user chooses both voice and graphical models. In 74.6% of the times multimodality was used, in which the use of voice implies the use of graphical interface with 86.9% of confidence, and the use of graphical interface implies the use of voice with 85.5% of confidence.

Rule	Support	Confidence
$\emptyset \Rightarrow$ VOICE	85.9	85.9
$\emptyset \Rightarrow$ GRAPHICAL	87.3	87.3
VOICE $\Rightarrow$ GRAPHICAL	74.6	86.9
GRAPHICAL $\Rightarrow$ VOICE	74.6	85.5
VXML_ERROR $\Rightarrow$ GRAPHICAL	50.7	92.3

Table 1. Rules from BASE-1

In order to identify common errors using the voice interface the BASE-2 was used to find association rules. The most common error identified was in the name identification task in the voice interface (Table 2). This error occured due to the wide variety of names and the high complexity of the name grammar in the voice interface. The association rules and the confidence/occurence relationship allowed us to verify that users choose to switch modes to recover from the error, after the second frustated attempt to interact using voice. In 98.1% of the interactions with the names graphical interface we found a previous voice interaction with 71.8% confidence (VOICE $\Rightarrow$ name GRAPHICAL (71.8, 98.1)).

Rule	Support	Confidence
VXML_ERROR $\Rightarrow$ name	3.3	33.6
Name $\Rightarrow$ VXML_ERROR	3.3	32.8

Table 2. Rules from BASE-2

4. Related work

A number of tools, methods and frameworks to multimodal interface design have become accessible in recent time. A method for developing Web-based multimodal interfaces was proposed by [12]. In this method a framework splits the interface life cycle into four levels: tasks model, domain model, abstract and concrete interface models, each of them performing transformations on the previous levels until the achievement of the final multimodal interface. All the elements, models and transformations between the levels are specified evenly through a single interface description language, the UsiXML. Thus, the whole project knowledge necessary to lead the changes is explicitly contained in the processing rules, and implementing these rules is granted by a transformation mechanism.

A tool to assist in the development of multiple types of interfaces and different ways to combine graphics and voice in multi-devices environments was proposed by [10]. The interface can be adapted to the interaction resources available to avoid confusing the designer with many details related to devices and programming languages. Bourguet [4] has investigated the creation of a multimodal toolkit in which multimodal scenarios could be modeled using finite state machines. A more theoretical approach is CARE and its component based approach ICARE [3] that has provided inspiration for a comprehensive toolkit: OpenInterface [11].

A multimodal framework, which architecture is based on agents geared toward direct integration into a multimodal application, was designed by [7]. One of the most interesting aspects is the use of a parallel application-independent fusion technique. We could observe that these methods do not consider principles, guidelines, DR or Design Patterns to facilitate the design of multimodal interfaces. We believe such concepts are most important since they guide designers in making consistent decisions through the elements which compose the product and they are efficient techniques to capture, document and communicate the scientifically validated and applied knowledge.

Additional concerns were also identified, given that these tools, methods and frameworks are specific to a particular issue even if multiple interactions are available; there is a lack of systems that convey information using the modalities that are most appropriate to the users and their tasks; there is a lack of experience recording and spreading DR with the employed multimodal interactions. Thus, our aim is to overcome these issues and concerns by providing an approach and a tool as it was explicitly shown in the previous sessions.

5. Conclusions

Multimodal interfaces are viewed as a promising opportunity for achieving universal access in the near future. On the other hand the field is still novel and needs further research to build reliable and usable multimodal applications due to a lack of supporting tools for the project team. With these statements in mind, we have created the MultiModal Web Approach and developed its authoring environment that aim to assist designing, implementing and testing multimodal user interfaces, promoting modality integration, error handling, dialog management and DR reuse in a clear manner. Furthermore we have show the significant results applying our approach and our tool in real projects through the accomplishment of case studies.

In summary the tool had: promoted code reuse; decreased the time to create prototype for usability evaluations; promoted design documentation; supported DR storage and retrieval and decreased the usability issues. We have also applied association rules to identify errors and users behaviors interacting with multimodal interfaces since it is a well know tool to discovering interesting relations between variables in large databases. The next step in our research is to increase the Design Pattern support in the MMWA-ae to allow for a more robust interface prototype generation.

References

[1] D. Bohus and A. Rudnicky. Sorry, I Didn't Catch That! - An Investigation of Non-understanding Errors and Recovery Strategies. In *SIGdial*, 2005.

[2] J. Borchers. *A Pattern Approach to Interaction Design*. John Wiley & Sons, Inc., 2001.

[3] J. Bouchet, L. Nigay, and T. Ganille. ICARE Software Components for Rapidly Developing Multimodal Interfaces. In *Conference Proc. of ICMI 2004*, pages 251–258, State College, 2004. ACM Press, New York.

[4] M. Bourguet. A Toolkit for Creating and Testing Multimodal Interface Designs. In *Companion Proc. of UIST 2002*, pages 29–30, Paris, 2002.

[5] J. Coutaz, L. Nigay, D. Salber, A. Blandford, J. May, and R. Young. Four easy pieces for assessing the usability of multimodal interaction: the CARE properties. In *Proc. of Interact'95*, pages 115–120, ACM Press, 1995.

[6] H. Desurvire. *Usability inspection methods*, chapter Faster, cheaper!! Are usability inspection methods as effective as empirical testing?, pages 173–202. John Wiley & Sons, Inc., New York, 1994.

[7] F. Flippo, A. Krebs, and I. Marsic. A framework for rapid development of multimodal interfaces. In *Proc. of the 5th international conference on Multimodal interfaces*, Vancouver, B.C., Canada., 2003.

[8] A. T. Neto, T. J. Bittar, R. P. M. Fortes, and K. Felizardo. Developing and evaluating web multimodal interfaces - a case study with usability principles. In *SAC '09: Proc. of the 2009 ACM symposium on Applied Computing*, pages 116–120, New York, NY, USA, 2009. ACM.

[9] S. Oviatt, R. Lunsford, and R. Coulston. Individual Differences in Multimodal Integration Patterns: What Are They and Why Do They Exist?. In *Proc. of CHI'05*, pages 241–249. ACM Press, 2005.

[10] F. Paterno and F. Giammarino. Multimodal interaction: research papers: Authoring interfaces with combined use of graphics and voice for both stationary and mobile devices. In *Proc. of the working conference on Advanced Visual Interfaces*, 2006.

[11] M. Serrano, L. Nigay, J.-Y. Lawson, A. Ramsay, R. Murray-Smith, and S. Denef. The OpenInterface framework: a tool for multimodal interaction. In *Adjunct Proc. of CHI 2008*, page 35013506, Florence, Italy, 2008. ACM Press, New York.

[12] A. Stanciulescu, Q. Limbourg, J. Vanderdonckt, B. Michotte, and F. Montero. A Transformational Approach for Multimodal Web User Interfaces based on USIXML. In *Proc of the 7th international conference on multimodal interfaces ICMI 2005*, pages 259–266. ACM Press, New York., 2005.