Using Re-Usable, Secure Software Engineering Principles for Designing User Focused Mixed Reality Systems

Niharika Aggarwal
Department Of Computer Science and Engineering
Jaypee Institute of Information Technology, Noida

Aditya Garg
Department Of Computer Science and Engineering
Jaypee Institute of Information Technology, Noida

ABSTRACT
Today gaming is an inherent part of the lives of all people and the focus has shifted from fixed display gaming to Mixed Reality, leaving a gaping hole for secure software engineering approaches addressing both technical and human factors; along with the limitations of the current practices which shadow the outlook of the overall gaming experience. In this paper, we propose solutions to overcome the drawbacks of currently implemented software solutions for development of Mixed Gaming Systems.

As stated above, in the absence of secure software engineering approaches addressing both technical and human factors, the proposed solution needs to give equal priority to developers as well as users to overcome the setbacks. By using software engineering principles, methodologies as well as a new architecture, the features / focal point of Mixed Reality can be created for new and enhanced games with better user functionality, a smoother and robust development process. To lay the foundation for the development of newer games created for a better, more holistic and realistic game experience, the Software engineering principles need to be incorporated on all levels, ranging from abstract standards to operational development, to integration of user centered design activities. To achieve the desired result, implementation of the agile methodology for the software development life cycle is a proposed solution with emphasis on the creation of architecture MVCE (Model – View – Controller – Environment). The MVCE Architecture encompasses the common MVC (Model – View – Controller) pattern with an additional component named Environment to address the specific requirement of mixed reality interfaces.

General Terms
Design, Human Factors, Management, Usability Engineering, Software Engineering, Development Process, Mixed Reality.

Keywords
Augmented and Mixed Reality, Agile Development, User Centered Design, Prototyping, Adaptive, Usability Engineering, Software Engineering, Development Processes.

1. INTRODUCTION
Mixed Reality (MR), also known as Augmented Reality (AR), as defined by Milgram and Kishino [6], covers the complete spectrum between Virtual Environment (VE) and Augmented Reality, where interactive computer graphics and other media are integrated into real world environment.

Let us begin with the user interface and the interactions, or, simply put the human factor of any software. It can be said that in the games designed with a conventional user interface in mind, a fixed display, i.e. complete display being generated from an internal model completely in control of the software, is utilized. These games had spatial aspects as well as task of navigation which were in the form of simulation. With the emerging technologies for positioning, context recognition, and advancements in wireless communication technologies, the possibility of exploiting the dynamic context of the user has opened new door for taking gaming to an entirely different level. With the limitations of numbered interaction interfaces and hardware and its availability at competitive rates for the masses being overcome, exploitation of the positioning technology to capture the physical movement of players for gaming purposes is one example where advancements have been made [1]. Though this limits the use of the player's real world context to positioning, large real world geographical environment can be used as 'game areas' which are incorporated into game content. Physical movement can be captured as interactions within the game and with the context changing continuously in a partially controlled environment, such as this; it is a challenge to ensure that the game remains playable and enjoyable.

Currently, we have been discussing MR applications with emphasis on features as well as ways to implement them with emphasis on technologic limitations, their improvements and ways to use them. With the technological advancement reaching its peak, the creation of applications that use MR in their user interface, and its practability in terms of use are points to consider. [5]

The development of MR applications with interactions in user interface is hampered by a number of factors:

1. More stress is given on technical details rather than actual usability.
2. Lack in information of the limitations and capabilities of MR, unexplored usability in terms of user interface for most of MR techniques results in a poor user interface being designed by the designer.
3. While in terms of Software, a large amount of functionality needs to be developed due to incompatibility of MR application with the existing soft wares components, the requirement of specific

non-standard hardware like 3D tracking systems and specialized displays, which must be taken into account during the design process.

4. There is limited interaction between the user and virtual content.
5. There is little interfacing or linkage with other applications, users or the internet.
6. There is little support for the editing as well as customization of the content.
7. There is generally little or no repeat value provided by AR experiences.
8. There is not much thought given to interaction with other users or a community, or to group collaboration.

A user centered design process for MR applications is required that balances the influences of MR technology with SE and UE considerations. An integrated approach is needed that considers the technology-driven procedures of MR development while maintaining the systematic, controllable and manageable processes advocated by SE, and integrating appropriate methods and procedures from UE in order to develop usable solutions that are applicable in practice. The basis for a solid foundation and streamlined process and architecture in the form of agile methodology and MVCE architecture is discussed below.

2. CONVENTIONAL GAMES TO MIXED REALITY - CHALLENGES FACED

Mixed Reality differs in context from the traditional user games in terms of the available devices, characteristics of the environment as well as the user's activities. Capturing the user interactions and changes in the game resulting from the same ensure that the context of the game changes continuously. The four most important aspects of contexts which are in consideration here are:

1. Type of Device
2. Technology Constraints of Device
3. Physical Context of the Player
4. Activities of the Player's

The type of device ranging from smartphones to PDA's, laptops to netbooks etc., limits the functionality and the scope of the game severely. The type of device, the kind of sensors and hardware available for the gaming purpose define the functionality, features as well as rendering of the designed game. While the locomotion and its rendering may be accurate for players utilizing a mobile phone, the same would seriously hamper the chances of playing a game which has hand motion sensor requirements, requires inputs from user to allow his avatar to mimic the same in the game, has high end graphic display requirement etc. [1]

The technological constraints of displaying and interaction can be overcome with the help of Adaptation. Traditionally, for presentation, the use of text and 2-D graphics was widespread but with advancement in technology and better hardware, we have traversed and moved to 3-D graphics, multi-touch mechanism. In mobiles and wireless enabled laptops, it is common to use GPS to sense if a person is approaching a gaming zone. However, one major flaw in this is that it is not so accurate as to augment the images of the actual user environment which is spatially correct with the game info, the photographs taken by the player / user from his mobile camera / taken from a normal camera and uploaded to the game running on the laptop so that the image can be analyzed and taken as static background for rendering.

Taking the physical context of the player as well as his activities in the game, it is found that they serve as interactions, content as well as presentation within the game. The user activities are constrained by the available input modalities. In addition to the common activities like orientation, context awareness, playing etc., it is a good to include user's locomotion and actions as inputs to the game itself. Traditional games were designed for shorter durations, achieved using the concept of levels, to ensure the player's / user's complete attention during the limited time span which was brought forth in mixed reality games, however, there has been a growing need to provide enough status information that the user can continue after a break / interruption. The four types of drawbacks, namely, Type of Device as Device Independence; Technology Constraints of Devices as Adaptive Presentation; Physical Context of Player and Activities of Player as Context Refresh and Interaction Techniques are discussed below [1].

2.1 Device Independent

This can be achieved by keeping the narrative content in a device independent format so that all the users are presented with the same information and storyline, independent of the device. This ensures that the presentation format as well as media is adaptive. This is achieved by using finite automatons controlled by variable pool. Each hot spot accessible by user changes a certain variable, and hence the state of the automaton, which in turn is linked to the narrative information. The way the interactions are received by the game must also be rendered, understood and stored in a format that is device independent so that in case of a context refresh, from any device, the state of the game remains playable.

2.2 Adaptive Presentation

Many different types of media / displays are supported by the various types of hardware available in the market today. Hence the game engine has to consider all of these differences and take them into account by offering various types of presentations. Though Text and 2D Graphics are supported by all, 3-D offer a more attractive representation and is the more suitable presentation modality in many entertainment computing scenarios. Nevertheless, the complexity of 3D scenes that can be rendered in real-time on most devices remains quite limited for mobile devices compared to current game consoles or PCs.

2.3 Context Refresh and Interaction Techniques

As stated earlier, there is a growing need to overcome the time limit constraint ensuring the creation of longer games which maintain the state so that the player / user can continue the game even after a break / interruption. In mixed Reality, this is a big challenge as the context saved may not be available when the user resumes from the initial saved mode. Context Refresh is a method wherein a user can see his recent activities in the application, especially in cases in which the game content has been adapted to context which is no longer available.

The context refresh feature needs the history of all the interactions of a player. However, since the information from the starting of the game may prove to be cumbersome to the user to go though as the information may not have any relation to the current context, we can apply a filter for checking that only the recent and important interactions are displayed.

Also, the implementation of mixed reality in real world space, the users sparingly use their hardware devices to check for new information once they reach the selected game zone. Once in the area, the users are busy looking for markers and clues in the real world area which will help them proceed further.

3. MR DEVELOPMENT - USER CENTERED DESIGN

To address the problems associated with designing applications having MR in their user interface, suitable processes are required that guide the development and support effective communication between the players. Apart from the general processes that are applicable for MR development process, specific extensions of the same are also required. The urgent requirement for discovery oriented design work for design space exploration is much more prominent and also needs to be addressed [3].

The base technologies developed or in the process of being developed are available as software libraries which provide low level functionality and certain more complex components which are supported by MR specific toolkits. These differ from the traditional UI in two specific areas. Firstly the established interaction techniques that can be identified and reused as components are missing and secondly, there is no standard way to create exchange and manage MR. An integrated approach is required that balances the influences technology driven procedures of MR development with due consideration to the systematic controllable and manageable processes of SE, and integrating the appropriate methods and procedures of UE. This requires:

1. An iterative design and development process needed to explore and evaluate the feasibility of specific MR technologies as interaction techniques.
2. A flexible process to handle the changing requirements smoothly, especially when evaluating the technical feasibility of a proposed design.
3. Equal consideration of hardware development, software engineering practice and UI design activities, especially if application specific hardware is implemented.
4. Integration of user centered design activities to create solutions that focus on real users, their needs and the context of use.
5. Accessibility of design representations and tools to all stakeholders in the design process to ensure effective communication.

The software principles required for development like Agile Methodology as process for product development as well as the MVCE architecture are discussed below [5][2][4].

3.1 MR Development – Agile Methodology

A lightweight process in Software Engineering (SE) used for the entire software development lifecycle is the Agile Methodology. This methodology is characterized by short timed iterative and evolutionary development, adaptive planning, evolutionary delivery, as well as a rapid and flexible response to change. Agile software is development uses a more delivery as well as code quality oriented approach. It does not rely on heavy documentation and compensates the same by information exchange between the team members in an informal manner and aspire towards early and frequent feedback. Scrum is one such method, agile and iterative incremental, which has become very popular for small sized projects. Here, the development activities are organized into short iterations, called sprints, each of which starts with a sprint planning meeting. The unaddressed items from the previous sprints, termed as backlogs, are stored in product backlog, a repository for storing all requirements related to development and the high priority backlog items are picked by sprint teams, a small interdisciplinary group comprising of 7-9 people having full authority to decide the best way to achieve the desired outcome, to bring them to completion. Every sprint is ended with a sprint review meeting where the current product is showcased to the stakeholders and their opinion and feedback is taken [5].

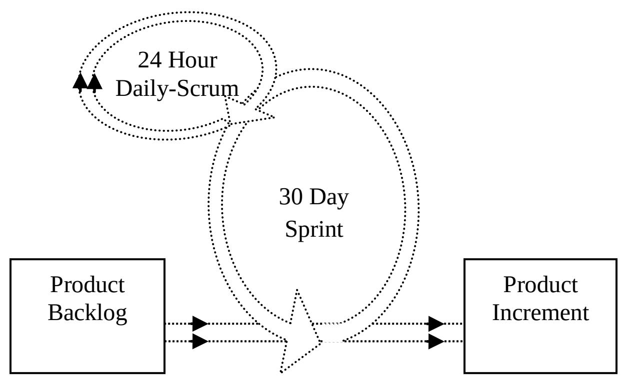

Figure 1: Iteration Cycled of Scrum

With a clearer picture of the agile methodology and improved and enhanced user interface, a special process has been derived to address the requirements for development of applications with MR based user interface. The specialized process designed to create these applications can be divided roughly into two stages:

1. Preliminary Exploration, Analysis and Design
2. Development and Evaluation of the Product

The exploration phase of the scrum development module is a rather short phase where the following details are worked out by the sprint teams; namely, system architecture, technology topics to discuss feasibility, while the stakeholders develop the product backlog. It has characteristic property of being a single iteration process. As, the scrum method does not have a design phase, this phase is used to establish user requirements, to generate alternate MR designs that address these requirements, and start exploring the available design space from the functional, interface and hardware perspective.

Based on the roles of different people, different tasks are addressed simultaneously in the exploration phase. UE experts analyze the users' needs, goal and context in which they work by the process of interviews and contextual inquiries. The MR experts identify potential base technologies and architectural concepts suitable for the application context and also analyze the same on technical feasibility, development, potential and possible restrictions and constraints. The results and finding by the experts are communicated within the entire team to design the application based on these findings and a short document is made to record the same. The product backlog created by the stakeholders is a working document that is implemented during the process of development. After this, the MR interface to be implemented is shortlisted and decided and potential design alternatives are also explored, designed and a variety of possible solutions are generated. To be effective, the design representations should be quick and cheap to generate and can be discarded without a high penalty. A few examples of such kind of designs are design sketches, paper prototypes, mock ups etc.

The evaluation phase of the scrum development module the verification of the technical implementation is done in the form of software testing in addition to the following. In the design centric initial iterations the evaluation can focus on design review and critiquing conducted by the development team. This ensures that the design becomes more robust and stable shifting the focus to interface. The intermediate design representations (paper prototypes, mock ups, sketches etc.) are used to review the design as well as conduct initial evaluations with user. Once a complete system is available, to validate the interface and system functionality, usability testing with real users is performed. With the partial completion of the software, the technical standard software test activities are commensurate and for MR specific technologies, technology tests need to be performed, either as a part of the main process or as a spun off to a separate scrum cycle, to ensure that the envisioned technology solution is viable.

The Software development team consists of experts in the field of user interface designers, usability engineers, MR experts along with software engineers and developers, each serving a specific purpose in the process. The User interface designers and usability engineers are focused on the user requirements, human factors and usability issues. While the former are required to guide the initial design centric iterations, the latter control the usability required to validate the proposed designs. MR experts are required to provide expertise on potential base technologies. They complement the implementation knowledge of software engineers with specific know-how of MR technology and interaction techniques and focus on technology studies, MR technology adaptation, integration and implementation and ensure that the evolving system remains within the constraints of available technology.

The product backlog captures the desired functionality as well as the more specific sprint backlogs and the evolving (partial) implementation in the code repository. The extended design activities and usability engineering activities require extensions / additional artefacts i.e. product backlog to capture design representations like sketches, paper prototypes, mock-ups, specialized representations of the user's context, goals and workflows etc. which are required for creation and modification. Versioning and other features commonly applied in code repositories can be useful for these representations as well. Because these design representations are easily accessible to all stakeholders they help to integrate software engineers and end-users into the design process.

In a sprint planning meeting, a subset of usability requirements is selected for implementation in the next sprint. For each usability requirement being addressed in the sprint, refined and specific workflow requirements are derived. The goal in the sprint is to satisfy the usability requirement – the corresponding more detailed workflow requirements serve to map these to user interaction sequences and technical requirements. User Interface Requirements define properties of the intended system that are derived from Usability or Workflow Requirements, During the development, they provide guidance for designers regarding the information and navigation model, which can then be aligned with other technical models. The UI requirements are derived and refined during the sprints – typically in a participatory design approach involving UI, UE and MR experts as well as end-users. They provide the link from the other views to a technical implementation solution.

For the software development phase, the items and functionalities selected in the sprint planning meeting are developed, tested and made ready for deployment in repeated iterations and at the sprint review meeting, the same are showcased to the project stakeholders. The items pending consideration are put in the product backlog and the high priority tasks are taken up to be developed in the next sprint, after approval for the same has been obtained in the sprint meeting.

3.2 MR Development – MVCE Architecture

For the development of graphical user interfaces (GUIs) in MR, a dynamic and adaptable system, a separation of concerns is required to enable the reuse of individual components as well as exchange of presentation elements and interaction techniques independently from the underlying application as well as make the additional functionality available to the user without increasing the user's workload too much. The next generation user interfaces (NGUIs) provide a set of new technologies for human-computer interaction that has high potential in this domain. NGUIs diverge from the WIMP paradigm (window, icon, menu, pointer) and employ novel techniques like virtual, augmented and mixed reality interaction or tangible, embodied and multi-modal interfaces. Though this seems like the ideal solution to build our MR interface applications on, the challenges are that NGUIs are less representational, i.e. no icons representing objects, and focus on reality-based interaction styles which leverage the users built in intuitive abilities by exploiting pre-existing skills and expectations from real-world experiences rather than computer trained skills which deviated from the purpose of designing and developing MR applications [2][4].

The Model-View-Controller (MVC), introduced by Reenskaug (1979) [2], is an architectural model specializing in this regard. The MVC structures user interfaces into three components to ensure modular design, that is the visual and interaction aspects of a user interfaces are separated from the underlying application. The model (M) represents the application data and encapsulates the functionality of the application. The view (V) encapsulates the visual elements of the user interface, like buttons, text fields etc. The controller (C) handles the interaction details and communicates the necessary actions to the model. The modular design enables multiple views and controllers to be present for the same application/model which is desirable for MR interfaces that rely on specialized hardware that may not be available in all situations.

In standard GUIs the complete visual presentation that the user sees is provided by the view component, which determines the presentation and depends only on data from the model. In contrast to this, MR interfaces combine the real environment, the main difference between conventional UI and MR, with additional information provided by the application. This makes the use of both an environment model of the real world that links the MR application to the real world and an augmentation model a necessity. The augmentation model has a close correspondence to the model in MVC and can be manipulated, either by the user through interaction techniques or by the business logic of the software; the same is not true for the environment model. In general changes in the environment can't be controlled through interaction techniques or the software. While sensor information could be handled as controller events in the MVC model, this can lead to complex and obscure models.

This leads to the introduction an additional environment (E) component, which captures the "real world" model of the application [4].

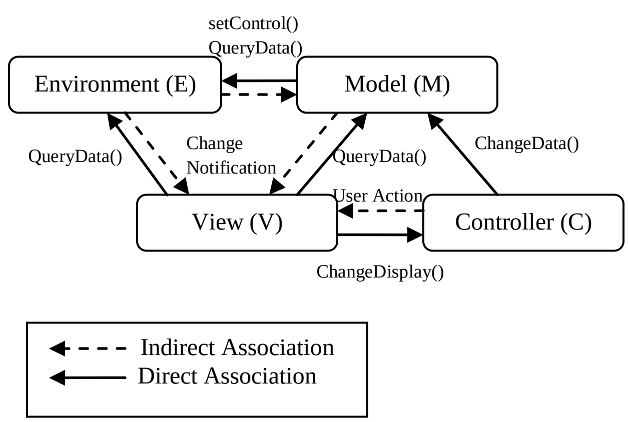

Figure 2: MVCE Design Pattern

Both the model (M) and the view (V) can query the environment (E). This allows capturing spatial association, controlling relations as well as allowing the components to be refined independently. This enables common interaction and user interface scenarios like environment sensing sensors or display devices to be addressed by modifying only the environment model or the view-model respectively.

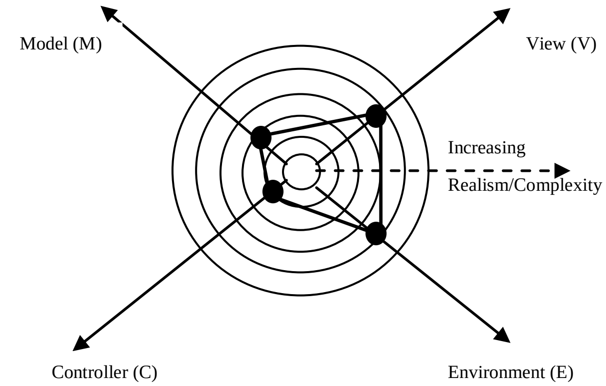

Figure 3: Illustration of MVCE Design Pattern

4. CONCLUSION AND OUTLOOK

To exploit the full potential of mixed reality during the design process of complex technical systems, a systematic development is required. The focal points creation of a mixed reality with specific focus on the user adaptability has been provided in the first phase. The possible approaches to map interactions vary from game to game and also on the size and magnitude on which the game is played. Like to identify the position of a person, the marker based approach may work for small scale whereas use of differential GPS, orientation as well as accelerator sensors are a likely solution for mixed reality games being played on large scale geographic locations. The development of these applications with MR specific UI vary from traditional applications due to restricting boundaries in the area of base technologies, existing design specialization as well as tools required for development. An agile methodology approach to develop such MR applications taking the contribution of experts in the field of designing, development, software engineering life cycles for effective exploration have been discussed.

5. FUTURE WORK

The idea of being able to complete a game in conventional mode at a later time as well as games being played over extended periods of time are in the initial phase of discussion and more can be said on it after further research in this area. Also, the possibility of being able to create a compelling gaming experience in which the actual physical environment is employed as a meaningful part of the game is to be explored along with the idea of using user body motion as an interface technique. To do this, questions in the field of technology, game design and content authoring that will have to be addressed. Other areas which requires considerable work is defining new innovative methods to facilitate better interaction with the virtual content as well as ways to encourage more interaction between multiple users of AR applications. A way to describe such models in a standardized and interchangeable format is still lacking and clearly requires more research.

6. ACKNOWLEDGMENTS

We would like to thank Prof. Vikas Saxena of the Dept. of Computer Science and Engineering at Jaypee institute of Information Technology for his support and guidance without which this would not have been possible.

7. REFERENCES

[1]. Christian Reimann, and Volker Paelke, "Adaptive Mixed Reality", in ACE 2005, Valencia, Spain.

[2]. Volker Paelke, and Karsten Nebe, "Interacting Usability Engineering and Software Engineering in Mixed Reality System Development", in 2009 in ACM 978-1-60558-401-0/09/05.

[3]. Shamal Failey, and Ivan Flechais, "A Meta-Model for Usable Secure Requirements Engineering", in ACM in SESS 2010, Cape Town, South Africa, May 2, 2010.

[4]. Jorg Stocklein, Patrick Pogscheba, Christian Geiger, and Volker Paelke, "MiReAS – A mixed reality Software Framework for iterative Prototyping of Control Strategies for an Indoor airship", in Association of Computing Machinery, in Afrigraph 2010, Franschhoek, South Africa, June 21-23, 2010.

[5]. Volker Paelke, and Karsten Nebe, "Integrating Agile Methods for Mixed Reality Design Space Exploration", in ACM in DIS 2008, Cape Town, South Africa, February 25-27, 2008.

[6]. Milgram, P., Kishino, F.: A Taxonomy of Mixed Reality Visual Display, IEICE Trans, on Information and Sstems (Special Issue on Networked Reality), vol. E77-D, pp. 1321--1329, 1994.

[7]. Mercedes Amor, Lidia Fuentes, Mónica Pinto, "A Survey of Multimedia Software Engineering", Journal of Universal Computer Science, Vol 10, Issue 4, 2004.