

Virtual and augmented environments and realistic user interactions to achieve embedded accessibility designs, the VERITAS project Use Cases methodological framework and outcomes

Eleni Chalkia
CERTH/HIT
Athens, 15231, Attica, Greece
and
Evangelos Bekiaris,
CERTH/HIT
Athens, 15231, Attica, Greece

ABSTRACT

VERITAS (Virtual and augmented environments and realistic user interactions to achieve embedded accessibility designs), is a 7th Framework project, which core concept is the research and development of an open framework for providing inbuilt accessibility support at all the stages of realisation of mainstream ICT and non-ICT technologies. The project aims at delivering to product and software developers ‘generic’ instructions - embedded in an empowering virtual reality platform, for exploring new concepts, designing new interfaces and testing interactive prototypes that will inherit universal accessibility features, including compatibility with established assistive technologies.

When developing interactive systems, within a specific oriented project like VERITAS, the correlation between the system and the users is a challenging task, especially when the main purpose is providing solution to a problem considering users’ environment. The Use Cases come to provide a representation of the contract between the stakeholders and the system’s behavior. The Use Cases are used as an assistive tool to the designers, to help them understand and create computer systems and applications as artifacts of human activity. In this paper we will present the VERITAS project, the methodology followed for the extraction of its Use Cases providing selected examples of the procedure followed.

Keywords: Use Cases, Virtual environment, user interaction, simulation and accessible design.

1. INTRODUCTION

One of the highest priorities of the European Commission, as well as of the European Society, during the last years, is the equal and yet cost-efficient support of the growing disabled and senior citizens. People with

disabilities are not just a tiny minority of the population of the European Union. The lowest estimate, based on the currently defined disablement categories, estimates their total number at around 74 Million persons. However, when including in this group also the people with cognitive impairments, as well as people who are in the hinterland between fully able bodied and classically impaired, the percentage of people with disabilities across Europe reaches 15% [1].

In VERITAS, our target is to create tools that assist the developers in creating accessible ICT and non-ICT products for these disabled users that are embedded in the application they are using for designing these products, so as to have an inbuilt accessibility environment. The main VERITAS innovation lies in the fact that, even if there have been some limited and isolated attempts to support accessibility testing of novel products and applications, there is a clear lack of a holistic framework that supports comprehensively virtual user modelling, simulation and testing at all development stages and realistic/immersive experience of the simulation.

To this end, VERITAS aims to develop tools for inbuilt accessibility support at all stages of ICT and non-ICT product development. The goal is to introduce simulation-based and virtual reality based all-inclusive models at all stages of product design and development into Automotive, Smart living spaces, Workplace, Infotainment and Healthcare application domains.

Thus, the goal of VERITAS is to ensure that future products and services are being systematically designed for all people, including those with disabilities and functional limitations, as well as older people. In the current paper we will present the way the end users interact with the outcomes of VERITAS as initially introduced, in the form of narratives which constitute the project’s Use Cases and Application Scenarios.

2. USE CASES CONCEPT

When developing interactive systems, within a specific oriented project like VERITAS, we create possibilities for learning, work, and leisure, for interaction and information. In the design process of these systems, the correlation between the system and the users is a challenging task. This task is very demanding, especially when the main purpose is to provide a solution to a problem considering the users' environment. The Use Cases come to provide an indicative solution and representation of the contract between the stakeholders and the system's behaviour. The Use Cases are used as an assistive tool to the designers, so as to help them understand and create computer systems and applications as artefacts of human activity, as things to learn from, as tools to use in one's work, as media for interacting with other people [2].

Use Cases were initially presented by Ivar Jacobson in 1967 as usage scenarios and became immediately attractive because the term implies "the ways in which a user uses a system" [3]. In the mid-1980s, Jacobson coined the Swedish term "anvendningsfall", which roughly means "situation of usage" or "usage case", but when publishing into English translated it in use case [4].

Since Jacobson, the Use Cases have been used extensively in different points of most of the system development. They can be used to stimulate discussion within a team about a system-to-be. They might be used in order to report the actual system requirements, which will emerge from the user needs. Additionally, the systems final design can be documented using the same use case form. The Use Cases can be used for a system as large as an entire company, or as small as a piece of a software application program.

The main purpose of the Use Cases is to present, in a detailed and also clear and easy-to-understand way, the functional requirements of a system in a non-technical manner, but in a way that will describe the user's environment and yet will be easy from the developer to translate into technical characteristics of the system. The Use Cases can also be considered as a description of a system's behaviour, written from the point of view of a user who has told the system to do something specific. In this way, the Use Cases have the unique ability to help teams to understand the value that the system provides to its stakeholders [5]. In a more simple approach, Use Cases describe who is doing what and when, and also what is expected from the system for each request. To this end, Use Cases comprise a powerful tool to capture functional requirements for software systems, in order to evaluate them [6].

There are many ways in which a researcher can write the Use Cases. They may be presented as simple scenarios,

like narrative stories as it has been proposed from Carroll & Rosson in 2002 at the "Scenario-based design" [2], or it can consist of various different parts that decompose the scenarios, as proposed from Ivar Jacobson in 1986 with the detailed templates (fully dressed or casual) [4]. In all cases, the Use Cases should be well-written and easy to be read from the designer. They should be consisted of sentences written in only one grammatical form, a simple action step, in which an actor achieves a result or passes information to another actor.

In VERITAS case we have used the scenario based design in order to identify our Use Cases and during the design process of the project, we have enriched their descriptions using the Ivar Jacobson fully dressed template. The fully dressed template is characterized by:

- One column of text (not a table).
- Numbered steps.
- A numbering convention in the extensions sections that involves combinations of digits and letters (e.g. 2a, 2a1, 2a2, etc.).

The fully dressed template includes the most basic elements for a Use Case representation like the Use Case description, which is a small narrative that gives the goal of the Use Case, the scenarios that decompose the Use Case in smaller parts, the actors that participate in the Use Case, the priority level, the system input and output as well as the interaction steps between the user and the system in order to achieve the goal. Thus, a Use Case is a collection of related success and failure scenarios that describe actors using the system to support a goal. So, a goal holds together all the scenarios (success and failure). Scenarios and Use Cases go until goal success or abandonment.

But what is the deference between a scenario and a Use Case? The scenario is a sequence of interactions happening under certain conditions, to achieve the primary actor's goal, and having a particular result with respect to that goal. The interactions start from the triggering action and continue until the goal is delivered or abandoned, and the system completes whatever responsibilities it has with respect to the interaction. A Use Case is a collection of possible scenarios between the system under discussion and external actors, characterized by the goal the primary actor has towards the system's declared responsibilities, showing how the primary actor's goal might be delivered or might fail.

Another very important element of the Use Case is the actors involved, namely the stakeholders. A stakeholder is usually an actor of the system, primary or secondary. A primary actor is a stakeholder that calls upon the system to deliver one of its services. The primary actor has a goal with respect to the system, one that can be satisfied by its functions. The primary actor is usually the actor, who triggers the use case, so the use case starts because the primary actor sends a message, pushes a button, enters a

keystroke, or in some other way initiates the story. There are two common situations in which the initiator of the use case is not the primary actor. The first is when a Use Case is triggered from another Use Cases, like a sequence of events, and the second is when the Use Case is triggered by time. The primary actors are very important for the system under development because they are the ones that the system is actually designed for. They are the users of the system, which should cover their needs and requirements. The list of the primary actors of the system should be done in the very beginning of the project and this list should be used for the whole duration of the project.

Despite the fact that the primary actor is very important during the requirements gathering process, it could appear to be not so important for the Use Cases itself. What happens is that, over time, the use case writers discover that a use case can be used by multiple sorts of actors, whose needs are more or less the same. So, in the Use Cases when we refer to a primary actor, we mainly refer to the user who will do the specific task and might have, or not, the profile of the actor that has been set during the User Needs phase. To this end, sometimes, in a Use Case, by an actor we mean an individual, but we also mean the general category of individuals who can play that role. Of course, the primary actor becomes important again just before the finalization of the system, when the different actors are called to test and evaluate the system prototypes.

3. VERITAS USE CASES METHODOLOGICAL FRAMEWORK

Use Cases generally are designed to be used by the designer/developer. Despite that, in VERITAS, when developing the Use Cases, we need to bear in mind both the end users (developers and designers), but also the beneficiaries (elderly and disabled). Even if the Use Cases, since are technical driven, will only address the end users (developers, designers), the beneficiaries (elderly and disabled) and their requirements should be also included, since the final product, which will be developed using the VERITAS tools, will be used by them. Under this framework, each technical detail of the system must be clearly defined and correlated with the actors that are involved in it and, furthermore, there must be a clear and understandable explanation of the interaction between the user and each technological component-function.

The basis of the Use Cases of a system under development lies upon the user requirements. According to Brooks, requirements always change, especially when designs incorporate rapidly-evolving technologies, requirements change rapidly every 2-3 years [7]. The more successful a design is, the more widely-adopted is, and the quicker its users expect more from it in terms of

functionalities and design requirements. So, the Use Cases need to be updated constantly during the project, in order to fit the updated user requirements.

Additionally, the Use Cases can be used in various steps of the project duration. The iterative evaluation and updated procedure for the evolution of the Use Cases during the project are presented and explained in this section, showing in which state of the process we are now, what has been done in the previous years and what has still to be done.

As it has initially identified by Rubin, Use Cases are applied in many steps of the projects lifecycle [8]. This stands also in VERITAS, where the Use Cases have and will be applied in various steps of the development process. In the Analysis domain, the research started with the definition of the problem scenarios, which defined in order to assist on the methodology for the capture of the user and the industrial needs. The problem scenarios have been reformed according to the outcomes of the user and industrial needs, as well as the benchmarking of the existing models, into the initial list of the Use Cases which has been prioritized, discussed and assessed by end-users and beneficiaries, as well as by internal and external to the project experts, iteratively until its finalization. Finally, the Use Cases have been reformed into pilot scenarios that will be used from the external developers during the pilot testing phase in order to assess the functionality, usability, usefulness, satisfaction and interaction of the tools developed. A more detailed representation of the participation of the Use Cases in the different project steps, that has been adopted and followed in VERITAS, is presented in the scheme that follows.

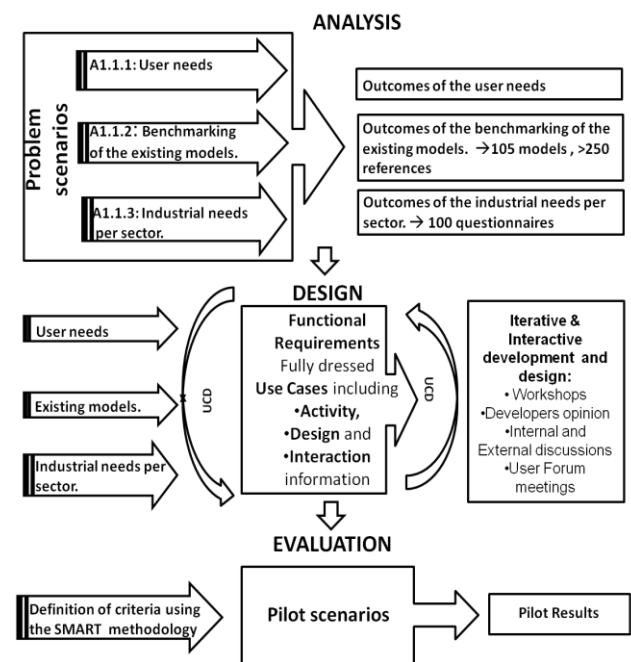


Figure 1: VERITAS Use Cases design framework

According to the figure above, the Use Cases are initially used in the Analysis of the problem of the project, in a more abstract format. Then they take their integrated format for the design process and finally are reformed into pilot scenarios in order to be used in the assessment and the evaluation of the developed systems. The outcome of this methodology at the moment consist of a set of problem scenarios, a set of 20 Use Cases and a set of preliminary pilot scenarios.

In these following sections, we will refer to all the development steps, by giving also concrete examples of the form of the Use Cases where possible.

4. ANALYSIS DOMAIN

The analysis domain has been mainly designed, discussed and developed in the beginning of the project when the main stakeholders' groups have been identified, the involved designers have been highlighted, as well as the beneficiaries groups and clusters have been reported; the industrial needs of designers and developers within the VERITAS application sectors has been identified, by pinpointing also in detail the similarities and the differences between them.

In the beginning of the analysis domain, and in order to have a clear vision of what is the problem that we are dealing with, some problem scenarios for the "before" situation have been drafted. Using the Use Cases, in the form of problem scenarios, at the analysis stage, can prevent the occurrence of costly error correction at later stages of the development cycle. In the following sections we will present one example of each application domain.

Below we present an indicative problem scenario for the automotive domain.

Problem scenario:

Paul has a company specialized in adjusting vehicles for use by people with disabilities. His team has extensive expertise in this field; however they still do depend heavily on personalization case by case since every user has a different set of needs. He therefore works together with driver rehabilitation specialists who perform comprehensive evaluations to identify the adaptive equipment most suited to the needs of the driver with disabilities. A complete evaluation includes vision screening and, in general, assesses:

- Muscle strength, flexibility, and range of motion
- Coordination and reaction time
- Judgment and decision making abilities
- Ability to drive with adaptive equipment

Upon completion of such an evaluation, the driver receives a report containing specific recommendations on driving requirements or restrictions, and a complete list of recommended vehicle modifications.

Equally, Paul's evaluators also consult on compatibility and transportation safety issues for passengers with disabilities. They assess the type of seating needed and the person's ability to exit and enter the vehicle. They provide advice on the purchase of modified vehicles and recommend appropriate wheelchair lifts or other equipment.

With the outcomes of the evaluation, Paul's team would start adjusting a car that seems to fit the needs of the user. While the vehicle is being modified, the user/driver must be available for fittings. This avoids additional waiting time for adjustments once the equipment is fully installed. Without proper fittings, problems might arise with the safe operation of the vehicle, and might results in the vehicle to be brought back for adjustments.

While Paul's team never ended in a situation that the car is not useful for the driver, it has happened many times that modifications still had to be carried out after the first presentation to the user of the completed vehicle. Paul's "ethical company bible" clearly states that no car leaves the premises without 100% satisfaction from the driver/user, thus iterations often happened, till the user/driver was fully satisfied.

The problem scenarios developed in this step are design-neutral and present the situation as it is trying to describe the problem of the users. Even if the problem scenarios have been developed during the early analysis of the project the team had a clear yet flexible idea of how technology might enhance current practice. The high level of abstraction of the problem scenarios has allowed the team to reform its visions without influencing the extraction of the user requirements or the extraction of the Use Cases list.

The outcomes of the Analysis domain have formed the basis for the update of the evolvement of the problem scenarios into the initial version of the Use Cases that is used in the Design domain that follows.

5. DESIGN DOMAIN

In the design domain, the protagonist is the functional requirements of the system-to-be. The functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. The systems functional requirements, either basic or exclusive are the features that characterize the system, making it useful and usable for the target groups and allow it to penetrate into the market [6].

The Use Cases are an effective tool for gathering the user requirements and presenting the functional requirements

of a system, when they are developed in a disciplined (systematic) and coherent manner, as part of a methodology that first creates a well defined domain-model.

For extraction of the Use Cases in the design domain the following realization steps too place.

1. Define a Use Cases Model (capable of supporting the variety of situations that VERITAS will support). The model is comprised of:
 - **Template** for the Use Cases descriptions (based on the fully-dressed format), in which the elements that are necessary according to VERITAS objectives should be inserted gradually. Important is to keep Use Cases very simple, in order to produce a Use Cases Index.
 - **Diagrams** accompanying each of the Use Cases textual descriptions (on the template). Diagrams are designed with the same capabilities of the template, i.e., providing the possibility of attaching details later. The diagrams are realized in a close to final step of the Use Cases, but are part of the Use Case model.
2. Define the Use Cases Index (comprised of a set of Use Cases titles). This step is achieved before the previous one, even if it comes second at the methodology. It corresponds to the Use Cases Index (comprised of a set of UC names) and it is the first level of detail to start writing Use Cases. The reason why this step comes second is the need to have a model before (from 1st step).
3. Discuss the initial list of Use Cases among the partners in order to come to a first common version, having in mind the following success criteria:
 - Containing all elements: such as UCs aim/scope, the trigger for the UC, the primary actor and possibly other stakeholders, all the interests of the stakeholders, preconditions, success and failure conditions.
 - The UC should contain a template with the necessary elements and diagrams, as well as the UC textual descriptions.
 - The UC should clearly show under what conditions the VERITAS functionalities are successful in relation to the problem/goal of the primary user.
 - The UC should clearly show what the minimal functionalities should be in relation to the problem/goal of the primary user for successful results.
4. Present the Use Cases in the 1st VERITAS Pan-European Workshop and User Forum with users (designers-developers) and with beneficiaries (elderly and disabled). Create a template for the Use Cases prioritization and a methodology to be followed. Prioritization of the Use Cases from both the users and the beneficiaries and notes of their comments.

5. Update of the Use Cases according to the 1st VERITAS Pan-European Workshop and Use Forum, correlate them to the different disability types and connect them to specific personas.
6. Discuss the Use Cases with the project partners and organize a review from external developers.
7. After gathering the commends from the internal and external developers the Use Cases have been updated and presented again to the users and the beneficiaries during the 2nd Pan-European Workshop and User Forum, where they have been discussed and commended.
8. After taking under consideration the commends from the 2nd Pan-European Workshop and User Forum of VERITAS, the Use Cases have been updated and discussed once more with the project partners. Their final format was structured and the same stands for their content.
9. Design of the UML diagrams accomplishing each of the UCs textual descriptions (on the template).

On the basis of the aforementioned framework the Use Cases of VERITAS were extracted and the extended list is presented below.

Category 1: Use Framework	
UC 1.1: User model generator.	
UC 1.2: Model platform.	
UC 1.3: Intelligent avatar editor.	
UC 1.4: Interaction adaptor.	
UC 1.5: Core simulation.	
UC 1.6: Multimodal interfaces.	
UC 1.7: Interaction manager and immersive simulation.	
Category 2.a: Automotive desktop design	Category 2.b: Automotive immersive design
UC 2.1.a: Car interior desktop design	UC 2.1.b: Car interior immersive design
UC 2.2.a: Motorcycle handling desktop design	
UC 2.3.a: ADAS/IVIS desktop design	UC 2.3.b: ADAS/IVIS immersive design
UC 2.4.a: ARAS/OBIS design	
Category 3.a: Smart living Spaces desktop design	Category 3.b: Smart living Spaces immersive design
UC 3.1.a: Home interior desktop design.	UC 3.1.b: Home interior immersive design.
UC 3.2.a: Domotics desktop design.	UC 3.2.b: Domotics immersive design.
Category 4.a: Workplaces desktop design	Category 4.b: Workplaces immersive design
UC 4.1.a: Office desktop design.	UC 4.1.b: Office immersive design.
UC 4.2.a: Collaborative tools desktop design.	
Category 5: Infotainment	
UC 5.1: Accessible metaverses design.	
UC 5.2: Collaborative games design.	
Category 6: Health Care	
UC 6.1: Remote Patient solutions design.	
UC 6.2: Mobile application design.	
UC 6.3: Health coach application design.	

Table 1: VERITAS project Use Cases list.

6. EVALUATION DOMAIN

The evaluation domain, as depicted in Figure 1 is the one that follows the design domain in which we are in the current stage of the project. In the evaluation domain the final pilot application scenarios that will be extracted from the Use Cases will derive.

In the current early phase of the project, user feedback may be obtained in rather informal settings, as defined also from the UCD methodology for example a participatory design session where designers-developers can be included in discussion and envisionment of activity scenarios, as it has already happened in VERITAS 1st and 2nd Pan-European Workshop and User Forum.

In the final test and evaluation of the VERITAS tools, a carefully designed summative evaluation will be carried out, with the goal of assessing how well the system performs with respect to its usability specifications. In that stage the pilot application scenarios will provide a realistic task context that will be analyzed in subtask so as to provide expected or desired usability outcomes. In that time the prototype will be robust enough to measure subtask times more detailed. Usability specifications developed in this way have two important roles in evaluation. First, they provide concrete usability objectives that can be serve as a management tool in system development and then the team's usability engineers are able to insist that redesign and improvement continue until they are met ([9], [10]). Second, the specifications tie the results of empirical evaluation directly to the usability issues raised during design. The pilot applications scenarios will be

7. CONCLUSIONS

On the basis of the aforementioned methodological framework the Use Cases of VERITAS were extracted to various forms, either descriptive in problem and application scenarios or specific for the development process using the fully dressed template. All the aforementioned have been realized keeping the user of the system in the centre of the development having as a top priority his/her needs and expectations.

To this end, the problem scenarios have been extracted in the beginning of the project so as to identify the core description of the situation as is for the under development system. The problem scenarios are concrete yet flexible, since their target is to assist developers envisage their system to be and not entrap them in non achievable solutions.

The development phase that followed required a more detailed description of the problem scenarios that also includes the correlation between the system and the users. Since the finalization of the Use Cases descriptions, the developers had identified the basic functionalities of the system, so it was possible to format the final Use Cases of the project following the fully dressed template format. The iterative procedure also promoted the iterative communication among developers, users and stakeholders, helping to make design activities more accessible to many sources of expertise.

Finally, the Use Cases will be used for the extraction of the final pilot application scenarios, when the prototype is robust enough to identify specific tasks to be objectively and subjectively measured, so as to evaluate the actual systems from the developers' perspective, as well as the developed application from the beneficiaries' perspective.

References

- [1] Report of the Inclusive Communications (INCOM) subgroup of the Communications Committee (COCOM) COCOM04-08, 2004.
- [2] Mary Beth Rosson and John M. Carroll, "Scenario-Based Design", Department of Computer Science and Center for Human-Computer Interaction Virginia Tech, Blacksburg VA, 2002
- [3] Ivan Jacobson, "Object Oriented Software Engineering: A Use Case Driven Approach", Book, 1992
- [4] Alistair Cockburn, "Use cases, ten years later", Paper, Originally published in STQE magazine, Mar/Apr 2002.
- [5] Bittner, K., Spence, L, (2006). Managing Iterative Software Development Projects. Addison-Wesley Professional, June 27.
- [6] A. Fantechi, S. Gnesi, G. Lami and A. Maccari Applications of linguistic techniques for use case analysis. In Requirements Engineering, Springer London Volume 8, Number 3 / 2003.
- [7] Brooks F, (1995) The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley, Reading, Massachusetts (Anniversary Edition, originally 1975).
- [8] D. M. Rubin, "Uses of Use Cases", Methodologies and Practices – White Paper, Softstar Research, Inc., 1998
- [9] Carroll JM, and Rosson MB, (1985) "Usability specifications as a tool in iterative development," Advances in Human-Computer Interaction, HR Hartson, ed., Ablex, Norwood, New Jersey., (New Orleans, USA), 225-231. ACM Press: NY.
- [10] Good, M., Spine, T.M., Whiteside, J., & George, P. 1986. User-derived impact analysis as a tool for usability engineering. In M. Mantei & P. Oberton (Eds.), Proceedings of Human Factors in Computing Systems: CHI '86 (pp. 241-246). New York: ACM.