The Engineering Reality of Virtual Reality 2002, A. Woods, J. Merritt, S. Benton, M. Bolas, Editors, Proceedings SPIE Volume 4660, Stereoscopic Displays and Virtual Reality Systems IX, pages 445–457, 2002.

Usability Engineering: Domain Analysis Activities for Augmented Reality Systems

Joseph L. Gabbard[*a], J. Edward Swan II[**b], Deborah Hix[a], Marco Lanzagorta[c],
Mark Livingston[b], Dennis Brown[b], Simon Julier[b]
[a] Systems Research Center, Virginia Tech, Blacksburg VA, USA
[b] Virtual Reality Laboratory, Naval Research Laboratory, Washington DC, USA
[c] Center for Computational Science, Naval Research Laboratory, Washington DC, USA

ABSTRACT

This paper discusses our usability engineering process for the Battlefield Augmented Reality System (BARS). ***Usability engineering*** is a structured, iterative, stepwise development process. Like the related disciplines of software and systems engineering, usability engineering is a combination of management principals and techniques, formal and semi-formal evaluation techniques, and computerized tools. BARS is an outdoor augmented reality system that displays heads-up battlefield intelligence information to a dismounted warrior. The paper discusses our general usability engineering process. We originally developed the process in the context of virtual reality applications, but in this work we are adapting the procedures to an augmented reality system. The focus of this paper is our work is on ***domain analysis***, the first activity of the usability engineering process. We describe our plans for and our progress to date on our domain analysis for BARS. We give results in terms of a specific urban battlefield use case we have designed.

Keywords: Augmented reality, wearable computing, usability engineering, virtual reality, domain analysis, HCI

1. INTRODUCTION

In software engineering, it is both anecdotally and statistically documented that the later in the development lifecycle a bug is discovered, the more expensive it is to fix that bug. Many engineering/development teams make the mistake of not properly or adequately performing necessary software engineering activities in early development stages, mistakenly thinking they are saving time by doing so. But when bugs are found late in the development lifecycle, the total cost of the product typically becomes much more expensive than it would have been. This aspect of software engineering has been well known for decades [1].

An analogous situation applies with usability engineering. Usability problems can involve serious costs; common usability problems include:

- functionality that is missing (e.g., some user task the system does not support),
- poor user performance on a critical or common task (e.g., users are unable to perform tasks in a reasonable amount of time),
- catastrophic user error (e.g., accidental user corruption of a sensitive database),
- low user satisfaction (e.g., users do not like the system because it is so hard to use), and
- low user adoption of a new system (e.g., users will not use the system, again, because it is so hard to use).

Just like software bugs, the later in the development process such usability problems are discovered, the costlier it is to fix them. Usability engineering (see Section 2) is a process that helps avoid and mitigate usability problems, by involving users early and continually throughout an interactive system's development lifecycle.

This paper presents our general usability engineering process and how it is being applied to development of a *usable* ***augmented reality*** (AR) system called BARS (Battlefield Augmented Reality System). BARS, described in Section 3, is an outdoor AR system that displays heads-up battlefield intelligence information to a dismounted warrior. Specifically, this paper focuses on BARS ***domain analysis***, the first activity of the usability engineering process.

[*] jgabbard@vt.edu; phone (540) 231-3559; http://www.usability.vt.edu; Systems Research Center, 562 McBryde Hall (0251) Virginia Tech, Blacksburg , VA 24061
[**] swan@ait.nrl.navy.mil; phone (202) 404-4984; http://www.ait.nrl.navy.mil/vrlab; The Naval Research Laboratory, Code 5580 - Building 34 - Room 218A, 4555 Overlook Ave SW, Washington, DC 20375-5320

The Engineering Reality of Virtual Reality 2002, A. Woods, J. Merritt, S. Benton, M. Bolas, Editors,
Proceedings SPIE Volume 4660, Stereoscopic Displays and Virtual Reality Systems IX, pages 445–457, 2002.

2. A USABILITY ENGINEERING PROCESS FOR AUGMENTED REALITY APPLICATIONS

2.1. What is Usability?

Usability is a critically important property of any interactive system and is related to both user performance and user satisfaction. Specifically, usability is a combination of at least the following user-oriented characteristics [2]:

- Ease of learning
- Speed of user task performance
- User error rate
- Subjective user satisfaction
- User retention over time

Note that each of these characteristics is *quantifiable*; each can be measured. Thus, usability is quantifiable; it is not something that is merely subjective.

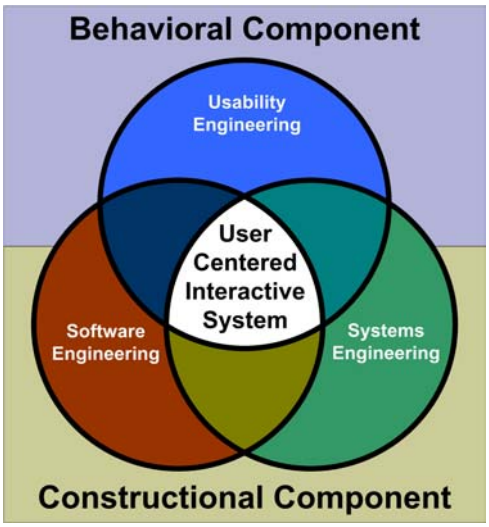


Figure 1: The engineering efforts required to produce a user centered interactive system.

An interactive system with high usability is both ***useful*** and ***usable***. *Useful* indicates that the system supports tasks users need to accomplish as part of some larger context. *Usable* indicates that users can utilize the system with a minimal and quantifiable amount of effort and training. An excellent indication that a system has succeeded in being both useful and usable, and thus has high usability, is when users choose to adopt a system on their own, with no organizational prompting. System usability is not something that happens by accident or good luck; it must be engineered into a product from the beginning of the development process. ***Usability engineering*** is a cost-effective, user-centered development process that ensures a high level of usability in an interactive system. It does this by involving users formally, early, and continually in the development lifecycle. Producing a *usable* interactive system requires complementary and parallel application of *systems engineering*, *software engineering*, and *usability engineering*, as shown in Figure 1. While systems and software engineering processes are routinely included in the development plans of interactive systems, usability engineering is often given little or no resources. While all three of these processes are necessary, this paper focuses only on usability engineering activities.

2.2. Complementary Engineering Components

As also shown in Figure 1, there are two distinct types of components involved in interactive system development: the ***behavioral component*** and the ***constructional component*** [2, 3] The *behavioral* component represents the view of the user and user interaction with the application, while the *constructional* component represents the view of the software developer and overall system.

In the behavioral component, *usability engineering* supports development of *user interaction* — the look and feel and behavior as a user interacts with an application. User interaction components include all icons, text, graphics, audio, video, and devices through which a user communicates with an interactive system, as well as navigation, layout, content, and so on. In the constructional component, *software engineering* and *systems engineering* support development of *software*, including that for both the user interface and the rest of the application (i.e., the non-user-interface, purely computational software), as well as other non-user-interface elements such as *hardware*.

Roles that support these two different components require different training, skills, and attitudes. Usability engineers do their work in the behavioral component, while software and systems engineers and related roles do their work in the constructional component. And while these roles are relatively well-defined and engineers are well-trained for software and systems development in the constructional component, these roles are much less well-defined and there are far fewer well-trained practitioners for user interaction development in the behavioral component.

Well-known techniques from software and systems engineering are appropriate for developing and evaluating user interface software. This kind of software evaluation can have many objectives, such as determining fidelity of a design

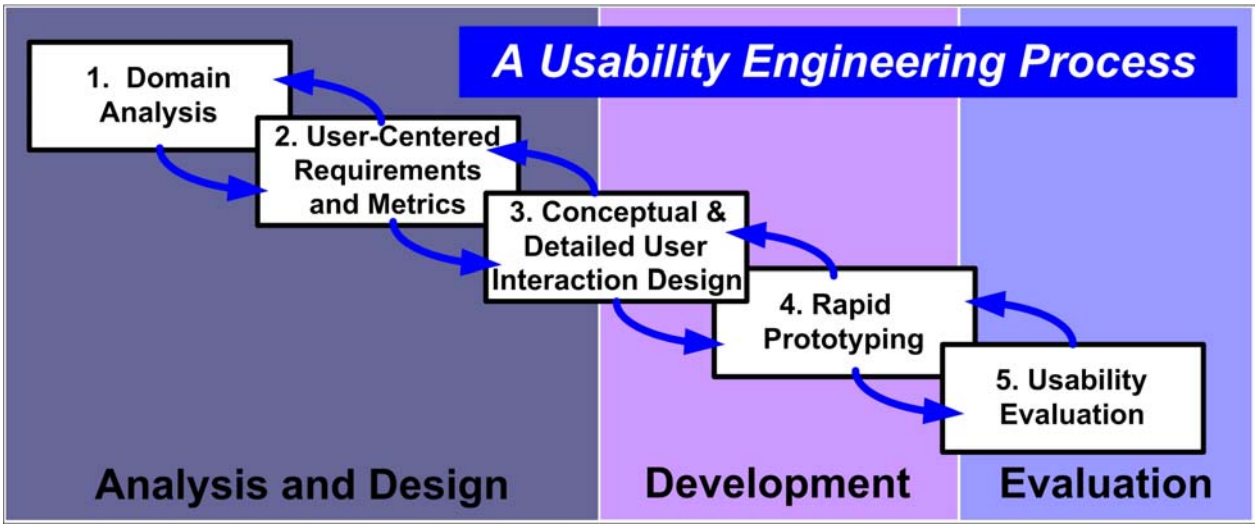


Figure 2: A usability engineering process and the associated activities.

to its implementation, reliability, reusability, and so on. Usability, however, is *not* one of these objectives, and usability engineering is a very different set of methods, which are described in Section 2.3. It is *not* the user interface software component that is engineered for usability, but rather the user interaction component (which just happens to be instantiated in software).

Both the behavioral and the constructional components are necessary for producing any interactive system, including AR systems, but the *component that ensures usability, and in which usability engineering is applied, is the behavioral component*. That is the context and focus of this paper.

2.3. Usability Engineering Process Diagram and Description

2.3.1. A Generic, Iterative Process

A high-level, generic usability engineering process is shown in Figure 2. That is, this version is applicable to developing any computer-human interface. In practice, we adapt the conceptual framework in Figure 2 to specific needs of a given project. Figure 2 thus represents a generalization of our experiences across a number of (mostly virtual reality) projects [3, 4].

The usability engineering process is depicted in Figure 2. While it spans the entire software development lifecycle including *analysis and design*, *development*, and *evaluation*, the usability engineering process specifically consists of the five distinct activities shown in Figure 2. Each of these activities is briefly described below. As indicated by the right-pointing arrows in Figure 2, these activities are nominally performed in the order given, beginning with *domain analysis* and proceeding through to *usability evaluation*. The left-pointing arrows indicate that each activity can iterate with the activity which proceeds it. However, our experiences have shown repeatedly that almost any activity can be followed by any other activity; and likewise we can iterate between any two activities. Thus the boxes should be considered completely connected, not just stepwise connected.

While the five activities within the usability engineering process can be applied to software lifecycles which are already underway, we have learned from experience that the most usable systems are created by leveraging the process early in the software lifecycle and employing all the activities listed in Figure 2.

2.3.2. Description of Usability Engineering Activities

1. Domain Analysis is the activity which answers two critical questions about a specific system domain:

- Who are the users?
- What tasks will they perform?

Domain analysis also reveals methods by which users attempt to complete tasks. Perhaps the most important result of domain analysis is that usability engineers gain an understanding of the user's point of view.

2. User-Centered Requirements and Metrics are quantitative ways of respectively specifying and measuring user interaction and user performance with the system. User-centered requirements ensure that requirements definition efforts reflect user goals and likely user activities. User-centered metrics are what allows usability engineers to assess the usability of a system. These metrics are based on measures related to the task domain.

3. Conceptual and Detailed User Interaction Design activities encompass designing a particular set of user interactions, based on tasks and users of those tasks. This activity ranges from conceptual, where large-scale usage and equipment requirements are discussed, to detailed, where specific user interactions are developed.

4. Rapid Prototyping is a quick and temporary way of implementing detailed user interaction designs for the system. Implementing designs is, of course, necessary before those designs can be evaluated. But it is important that initial versions of these designs be implemented rapidly and cheaply, because they are certain to change after usability evaluation. Thus, user interaction designs should not be committed to real system coding until later in the process, after various rounds of user-based evaluation. For example, if a particular set of user interactions involved different ways of drawing an object, initial designs would be created with a drawing package, as opposed to writing special-purpose rendering code.

5. Usability Evaluation is the activity of assessing and measuring the usability of a user interaction design. The purpose of evaluation is to drive successive iterations of user interaction design by pointing out interaction design flaws, as well as missed task and system requirements. Evaluation can also validate a good interaction design. Usability evaluation is itself a very broad and extremely important topic (see, for example [3]), but will not be further addressed here in the interest of space.

3. THE BARS APPLICATION

3.1. BARS Problem Domain

Many military planners believe that urban terrain is expected to be one of the most important environments that warfighters will face [5]. Because of increased urbanization, it is expected that many future military operations (such as peace-keeping or hostage rescue) will occur in cities. However, the urban terrain is also one of the most demanding environments. First, it is extremely complicated and inherently three-dimensional. Above street level, the infrastructure of buildings may serve many different purposes (such as hospitals or communication stations) and can harbor many types of risks (such as snipers or instability due to structural damage). These features are often distributed and interleaved over several floors of a multi-floor building. Below street level, there may be a complex network of sewers, tunnels and utility systems. Cities can be confusing (especially if street signs are damaged or missing) and coordinating multiple team members can be difficult. To ensure the safety of both civilian and military personnel, it has long been argued that environmental information must be delivered to the individual user in situ in that environment. However, technologies such as radios or handheld map displays are of limited use because the information is difficult to integrate or detach the user from the surrounding environment.

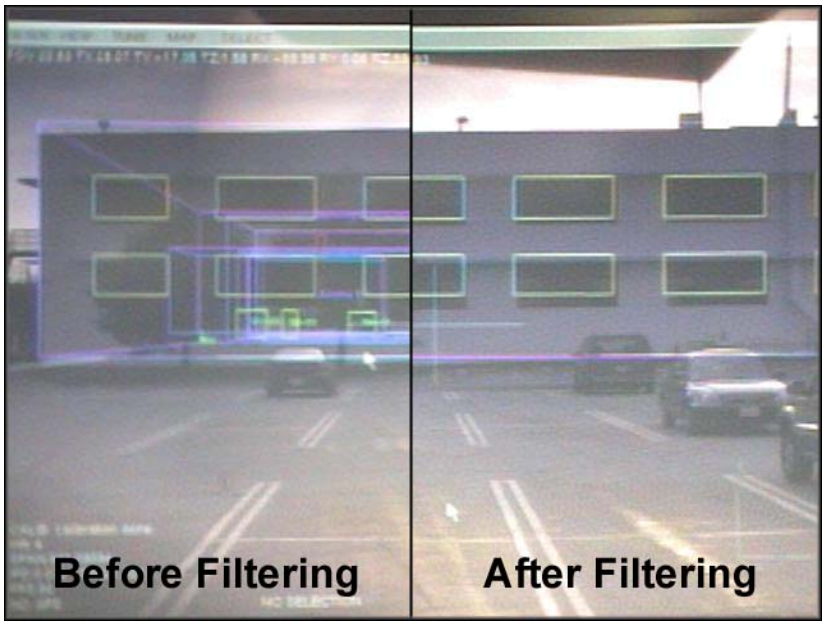


Figure 3: An augmented reality user's view – shown before and after information filtering.

BARS seeks to overcome these difficulties through the use of mobile augmented reality. Augmented reality is a display paradigm that mixes computer-generated graphics with a user's view of the real world (Figure 3). The user wears a see-through head-mounted display that the system tracks in six-degree-of-freedom space (position and orientation). Computer graphics are created and aligned from the user's perspective with the objects to be annotated. By providing direct, hands-free access to information, mobile augmented reality has the potential to recast the way information is presented and accessed.

Mobile augmented reality has many research challenges that are yet to be overcome. The technological challenges of hardware and software design are well-known [6, 7]. However, there are many issues related to the design of the user interface. Two problems are illustrated in Figure 3: information overload and the "Superman X-Ray Vision Problem" [8]. Information overload arises from providing the user with too much information. The "Superman X-Ray Vision Problem" encapsulates the fundamental advantages and disadvantages of mobile AR. With such a system, a user has "X-Ray" vision and can see data about objects that are not visible. However, the user loses occlusion cues, which are extremely important for perceiving depth. To overcome these difficulties, we are implementing an evaluation-based system development cycle, guided by the usability engineering process described in Section 2.

3.2. System Description

The hardware for the current implementation of BARS [9] is shown in Figure 4. BARS is built using the following commercial off-the-shelf products:

- an optical see-through head-mounted display,
- a position tracker (which tells where the user is located),
- an orientation tracker (which tells the direction where the user is looking),
- a wireless network,
- interaction devices (current a wrist mounted keyboard and wireless hand-held mouse), and
- a mobile computer with 3D graphics accelerated hardware.

The software architecture includes subsystems to manage the geometry database, distribute environmental information, and to filter the information for display. The geometry is acquired through traditional modeling approaches, such as photogrammetry or surveying. The geometry is hierarchically organized into objects, which are then assigned semantic meaning and given information attributes. These attributes can include whether the geometry represents physical objects (e.g. a building, building details such as doors and windows, or trees), moving objects like friendly or enemy forces,

Figure 4: Current prototype of the Battlefield Augmented Reality System (BARS).

invisible objects (e.g, .minefields or sewers), or completely virtual objects such as way points, lines of deconfliction, or target markers.

The primary limitations of the system are 1) the outdoor tracking systems, and 2) the see-through display device. It is well-known in the virtual reality field that tracking is a difficult problem, and an unprepared outdoor environment is a significant challenge. Orientation tracking relies on detecting the Earth's magnetic field, and sensors are confounded by the metal found in urban infrastructures and vehicles. Position tracking utilizes global positioning satellites (GPS), and works well only if the satellites can be seen via a direct line-of-sight, which can be difficult to maintain in an urban environment. Inter-reflections of the radio signals can also degrade performance. Current see-through display devices use liquid crystal displays, which limited by the brightness and field of view they can produce. Currently there is no liquid crystal-based display which can match the luminance values of a desert environment both at midnight and at high noon. Displays based on retinal laser scanning are planned for future implementations.

4. DOMAIN ANALYSIS PROCESS

Within the usability engineering process, domain analysis activities play a critical role in laying the groundwork for developing a user-centered system by clearly defining a context (both user- and task-based) within which user interaction will be designed. That is, an effective domain analysis ensures that system features and user performance are geared toward effective end-use within a specific usage context. As depicted in Figure 5, our domain analysis process typically consists of four (often overlapping) main activities: *use case development*, *user profiles*, *user needs analysis*,

and *user task analysis*. These four activities are described in more detail below. The results of these activities (in combination with other non-user-based analysis activities) include several types of *requirements*, also shown in Figure 5, which in turn guide user interaction design and may, in later stages of the development lifecycle, serve as an acceptance checklist.

Software engineers and project managers typically find requirements very useful for a number of reasons. One of their most immediate uses is to document features and functionality of the system, and to enumerate parts of the system that must be designed and built. It is this use of requirements that allows usability engineers to assert the interest of users by inserting user-centered requirements into a requirements definition process that traditionally focuses on technology and system performance issues. And as discussed above in Section 1, engineering usability at the requirements gathering stage, as opposed to later stages in the system development lifecycle, greatly reduces overall system cost. When possible, software engineers should be involved in as many of the domain analysis activities as resources permit to ensure that the technical limitations of development effort are respected as well as to gain a better understanding of the domain and the usability engineering process.

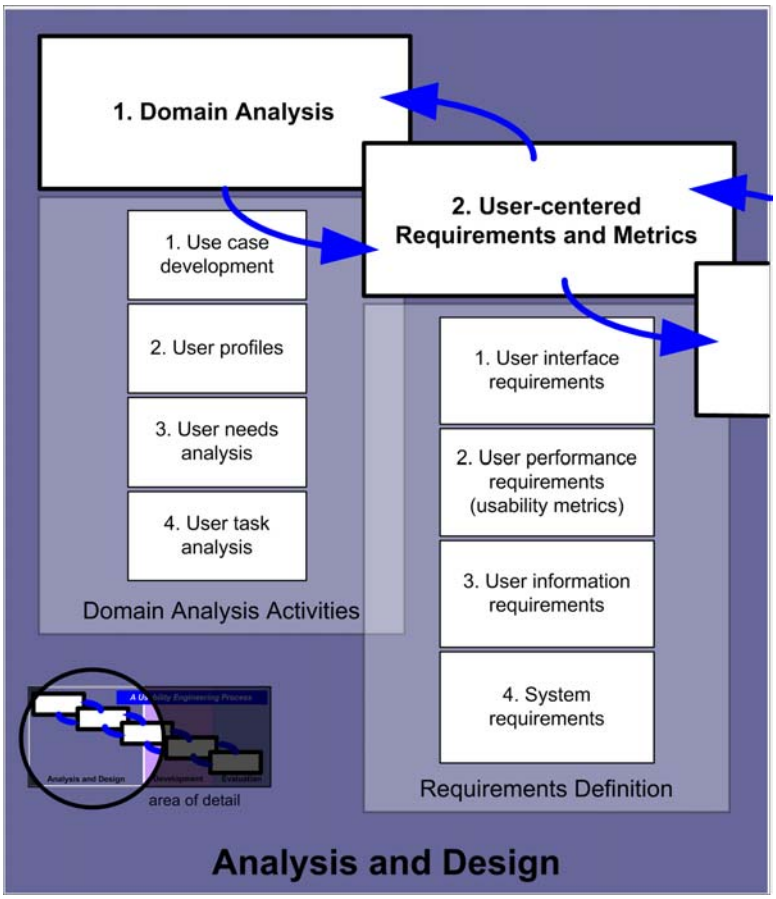


Figure 5: Activities of domain analysis as drivers for user-centered requirements definition.

The domain analysis activities we present help define specific user interface requirements as well as user performance requirements, or quantifiable usability metrics, that ensure that subsequent design and development efforts respect the interests of users. User information requirements also identified during domain analysis activities (and focused through the development of use cases) ensure that the final system provides useful and often time-critical insight to the user's task at hand. The most intuitive and usable interface in the world will not make a system useful, unless the core content of the system provides value to the end user. Finally, domain analysis activities may also shape system requirements, typically with respect to system components that affect user performance.

4.1. Role of Subject Matter Expert in Domain Analysis

A vital participant in the domain analysis process is the subject matter expert. This person (or persons) brings unique knowledge, skills, or expertise in the specific field for which the VE/AR application is created. Subject matter experts are typically very familiar with the types of users that will potentially employ the system, as well as the type of work (and specific tasks) that will be performed with the aid of the system. These experts also may have important knowledge about organizational and social practices and constructs with which the system must operate within — factors that may affect user acceptance and applicability. Optimally, the subject matter expert also has a good understanding of existing work-flow practices, and in particular, an understanding of the specific (existing) workflow and environment into which the technology will be inserted. To this end, a subject matter expert helps to ensure that the technology does not drastically change how users do their work, but instead, augments, aids and enhances how users perform work.

The knowledge a subject matter expert may contribute is often very different than that of a usability or software engineer. However, a successful domain analysis often employs all three types of experts, and in fact, often *requires* all

three types of experts. The *software engineer*, creating the constructional component, ensures that the products of a domain analysis can be implemented within the typical development constraints of time, money, personnel and technology. The *usability engineer*, working on the behavioral component, uses the *subject matter expert* to ensure (among other things) that the features and specifications under consideration are applicable to and designed specifically for the usage context. The usability engineer also acts as a facilitator of the domain analysis process, as well as ensures that the user and user performance are key drivers in any decision making process.

The BARS domain analysis was performed collaboratively by a team consisting of one subject matter expert, a handful of software engineers (performing various components of the domain analysis), and two usability engineers. The subject matter expert was an active duty Marine infantry officer with experience at the company commander level. As such, the subject matter expert was able to provide pertinent and specific guidance on numerous potential BARS uses.

4.2. Use Case Development

During a domain analysis, a subject matter expert is integrally involved in use case development, to set the overarching context for domain analysis activities. Use cases describe in detail specific usage contexts within which the system will be used, and for which the system should be designed. They typically describe the types of people, players, relationships and human dependencies involved (not necessarily just the system users but others who might be interacting in some way with those users), information content, needs and flow – i.e., how information is used and shared, the goals and specific tasks undertaken, as well as the environmental setting and context. A rich set of use cases form an essential context for further domain analysis activities (discussed below).

The use cases for BARS were developed over a series of several days involving the team described above. This team consulted military documents that describe protocol and tactics within an urban terrain to verify procedures (i.e., potential user tasks), as well as information (i.e., data) needs. Moreover, the team also consulted military documents that define specific terminology and symbology to ensure that the use cases were as accurate, thorough, representative, and concise as possible. Many times, information (such as terminology and symbology) captured during use case development is transitioned into the development effort — and eventually manifests itself in the user interface! This is, in fact, the desired outcome since a well conceived user-centered analysis should result in actual user interface and user interaction design (and implementation) decisions.

4.3. User Profiles

User profiles allow usability, software, and system engineers to focus their design efforts on a particular target population, and to effectively narrow the scope of the potential design space. User profiles provide a characterization of an interactive system's target population. The process of defining representative users in turn yields information that is useful in making design decisions. For example, a user profile for a specific system may identify:

- the amount of general computing experience a typical user of the system may have,
- the amount of VR/AR experience a user may have,
- the amount of training and experience a user may have within the usage context (e.g., urban warfare, in the case of BARS), as well as
- the type of social or organizational interaction and communication (formal and otherwise) a user may have with other users or other persons.

While user profiles are sometimes performed using surveys, we have found that for cutting-edge, emerging VE/AR applications, the actual end-use population may not be as easily definable or reachable as compared to, for example, applications designed to replace an existing in-house industry application (e.g., accounting system), and thus, have existing users in place. As such, the role of the subject matter expert and the development of pertinent use cases is paramount to creating a design that accurately targets a specific user profile.

Our subject matter expert was instrumental in identifying realistic use cases and subsequently what types of military personnel would likely be involved in those cases. Moreover, we were able to identify how these different users may use the system (i.e., what their goals may be, and how BARS may support military goals such as situational awareness, reconnaissance, comparing prepared intelligence to actual scene, and wayfinding), what type of background each user type may have, and what roles these individuals may play within the use case context (e.g., a squad leader role is very different than a demolition squad member).

4.4. User Needs Analysis

A user needs analysis further refines high-level user goals identified by user profiles by decomposing these goals within the context of the developed use cases. Moreover, the user needs analysis provides an assessment of what capabilities are required of the system to assist users in achieving these goals. The capabilities can then be further analyzed to identify specific user interaction requirements as well as information requirements.

While a user needs analysis can be derived from a combination of observational techniques such as surveys, interviews, and artifact analysis, we based the BARS user needs analysis on discussions and consultations with the subject matter expert. Within the context of the use cases, we identified specific user goals (for each user type) and then subsequently identified BARS system capabilities and information requirements needed to support these goals. When examining the capabilities and information requirements, we carefully assessed how these new capabilities may coexist with existing mechanisms for achieving user goals and closely examined how these new capabilities could be provided with minimal impact to the existing workflow. We then extended the analysis to identify technical user interaction requirements as well as information requirements. For example, for the user goal of wayfinding, we identified requirements such as "support labeling of buildings and streets", "support annotation of landmarks or other spatial locations", and "support interactive, top-down map views".

4.5. User Task Analysis

A user task analysis is one of the most commonly used and well-developed domain analysis activities employed by software and system engineers. Generally a user task analysis employs a set of methods for decomposing user tasks and understanding a set of procedures a user will undertake in pursuit of a particular goal. The basic approach is to first define a set of user tasks and respective task goals – these can be derived from other domain analysis activities and are scoped within the context of use cases. Each user task is then decomposed into a list of detailed subtasks so that associated information and user interaction requirements can be extracted to support these subtasks. The level of task refinement depends on the nature of the application, complexity of user interaction, and available usability engineering resources.

A user task analysis is useful for establishing user performance metrics by identifying what actions and results constitute successful user task completion, as well as assigning desired target values to each metric. For example, user performance metrics on a fine-grained positioning task may include time to complete the task, while the corresponding metric value for this metric may be 25 seconds. Often target values for desired user performance metrics have to be reassessed once representative users are employed during the evaluation phase. The user task analysis is also useful for noting potential errors or pitfalls in the task sequence, such as subtasks which may be extremely difficult or confusing – in these cases either the user task flow must be simplified or the user interface must be redesigned. In later stages of user interface development, the user task analysis can be used as a reality check on the interface's ability to exhaustively support all intended tasks. That is, the user task analysis can be used to identify user interaction components that may have been accidentally omitted or poorly designed.

We based the BARS user task analysis on extensive bibliographical research. We consulted both military history books [10, 11, 12, 13], and manuals of official combat doctrine issued by the US Department of Defense to Navy Seals, Marines, and Army personnel [14, 15, 16, 17]. We found that the two types of references provided complimentary information which fit into a natural hierarchy: the military history books provided a general task familiarization and context, while the combat doctrine manuals give specific tactics and rules of engagement. From the history books we constructed a list of user tasks, and then used the doctrine manuals to generate specific subtasks within each larger task. This process is described in more detail in Section 5.2 below.

5. DOMAIN ANALYSIS PRODUCTS

This section lists a series of products which have resulted from our domain analysis of BARS, and then discusses how these products yielded insights into BARS software and hardware requirements. As discussed in Sections 4.1 and 4.2 above, our first domain analysis product was a use case, which provided a context for additional domain analysis activities. We developed the use case with a subject matter expert over a period of several days. A representative diagram from the use case documentation appears in Figure 6. Briefly, the BARS use case is a military scenario, in which a platoon has the mission to infiltrate an enemy facility and destroy two tanks of non-lethal chemical agents. Figure 6 shows that the platoon consists of three squads; these squads have various positions they must occupy and

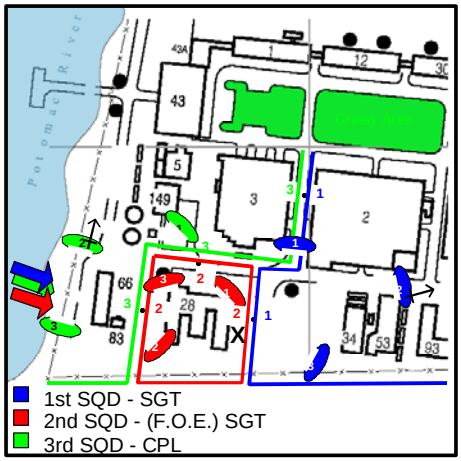
Background Description	Personnel Details	Diagram
Intelligence indicates that there are two tanks of non-lethal chemical agents that can be combined to create a deadly super gas. There are 13 men guarding the facilities (3*4)+1 leader. Given a 3:1 multiplier, we decide to send in 1 platoon reinforced (PLT+). A demolition or assault squad is included. Objectives: 1. Confirm presence of tanks. 2. Neutralize ability to develop deadly agent. 3. Delay rapid rebuilding of containers. Assault team will test for substance(s) and destroy containers. Scenario should take about 15 minutes.	PLT CMD – "Benotz" could use BARS to: • see blocking positions • see objective • convey info back to SQD Leader PLT SGT – "UmptyFratz" could use BARS to: • first to survey area (compare intelligence to actual) 1st Squad: blocking/security squad SGT Rock + Machine Gun Section 3 fire teams of 4 plus squad leader (13) 2nd Squad: FOCUS OF EFFORT (FOE) PLT CMD + SGT Stud + Demolition Squad (13+4+1 = 18) 3rd Squad: blocking/security squad PLT SGT + CPL Green + Machine Gun section (14)	

Figure 6: Sample text and diagram from BARS use case.

different duties at different stages of the mission. Figure 6 only gives a sense of the use case documentation — it has taken approximately 20 pages to capture the use case and associated analysis.

This use case sets the context for the following domain analysis products. Note that while this list of products is representative of AR systems, these particular products would not necessarily result from every domain analysis. These products are also what we have produced at the time of this writing — we expect to create additional domain analysis products the process continues.

5.1. User Information Requirements

A product we directly produced from analyzing the use case are information requirements for different BARS users. Figure 7 displays some of the user information requirements. The left-hand column of Figure 7 shows four different user tasks from the use case; for each task, the next two columns show information requirements for two potential BARS users, the squad leader and platoon commander. The complete list of use case user information requirements occupies several pages; Figure 7 is just a representative sample.

5.2. User Task Analysis

As described in Section 4.5 above, we performed a task analysis based on extensive bibliographical research. We consulted both military history books [10, 11, 12, 13] and military doctrine manuals [14, 15, 16, 17]. We considered both types of sources equally important and complimentary. Military history books provide a general overview of urban combat, describe associated problems, and outline tactics which have been successfully and unsuccessfully followed. They also provide a context in which we can discover the importance of information requirements such as situational awareness. Official training manuals are much more specific, and enumerate a variety of tactics and rules of engagement. However, because they give no historical context, it is very difficult to identify which tasks are the most important ones.

Description of Activity	**User Information Requirements**	
	Squad Leader	Platoon Commander
3rd Squad Lands, surveys area using BARS, compares reality to intelligence data.	• Intelligence data. • Geometric area model.	
3rd Squad secures landing and sets up blocking positions. Prepares for incoming squads.	• Planned blocking positions. • Fire teams locations.	• Where blocking positions are for 3rd Squad – not individuals, but squads. Can represent squad based on spatial position of three fire teams. • Sectors of responsibility. • Avenues of approach, and alternatives. • Landing point, alternate landing points. • Building numbers.
1st Squad lands and move toward their blocking positions. Note that 3rd Squad Section 1 expands their position by moving north east.		• Where blocking positions are for 1st Squad – not individuals, but squads. Can represent squad based on spatial position of three fire teams. Squad areas (sectors) of responsibility. • Avenues of approach, and alternatives. • Reported enemy movements perhaps vehicular (to relay to 1st Squad via radio).
1st Squad establishes final blocking position.		• Reported enemy movements perhaps vehicular (to relay to 1st Squad via radio). • Reportedly found strategic outpost – unmanned machine gun – observation point.

Figure 7: Some of the derived user information requirements.

Figure 8 shows a snapshot from the resulting task analysis document, which is currently 14 pages in length. From the military history books, we assembled a list of general tasks performed during armed conflict in urban areas. One of these is "movement outside buildings", which is displayed in Figure 8. Next, we used military doctrine manuals to create a detailed breakdown of the subtasks which each task requires. Figure 8 shows the high level task "moving through open areas", and two subsequent subtasks. In total we found several hundred different tasks. However, we realized that for a number of tasks AR and other visualization / cognitive systems are just not relevant — an example is the task of "grenade throwing". Based on this observation, we reduced the list to around 100 relevant tasks.

Task Analysis Components	**"Moving through open areas" – Representative User Subtasks**	
	Task 1. Identify open areas.	Task 2. Conduct movement inside building.
Perception	The Soldier must have a clear view of the area around him. Combination between reading a map and visual observation.	The soldier must be able to identify the buildings around him, their internal geometry and distribution, the access to them (windows, doors, holes), and their distribution in space.
Cognition	The soldier must be able to identify open from closed areas, maps can be difficult to understand in this sense.	The soldier must be able to identify the ways he can use the building's internal geometry and distribution to move (or advance) towards his destination.
AR support	The soldier using the AR system will have a 3D view of the terrain around him. He can invoke a 3D map that can manipulate at will. He will see which areas are open and which ones are interiors. He can even identify an open area like a park to an open area like the internal yard of a building.	The soldier using the AR system will have a 3D view of the terrain around him. He can invoke a 3D map that can manipulate at will. He can explore the interior of the 3D models of each building and understand more clearly their geometry and distribution.

Figure 8: A snapshot from the task analysis document.

System Features	User-centered Requirements
User interaction	• Allow the user to control the visual clutter by filtering.
	• Support hands-free user interaction (e.g., via voice or eye gaze).
	• Annotate the visual scene via voice.
	• Support user-acknowledgement of critical notifications.
Prepared information – intelligence and building/Site Plans	• Support planning and visualization of likely avenues of approach and landing points.
	• Display building layouts and building labels (when available).
	• Display planned objective site or set of objective sites.
	• Display planned points of reference and checkpoints.
	• Display landmarks and geographic locations of interest.
	• Provide navigational aids to the user.
Friendly personnel movement	• Visually represent the real-time, dynamic location of squads (e.g., fire team)
	• Support display of planned and real-time lines of de-confliction.
Intelligence feeds and enemy locations	• Display real-time notifications to individual BARS users.
	• Display the last known location of enemy forces (e.g., sniper).
	• Display objectives of tactical concern (e.g., strategic outpost).
Motion tracking, registration and display technology	• Support precise tracking of a users location and orientation.
	• Employ display technology that can be effective in both night and day lighting conditions.
	• Support both wire frame and shaded surface representations.
	• Support both monoscopic and stereoscopic computer graphics.
	• Display hidden and occluded buildings and objects.

Figure 9: Example of user-centered requirements.

Each task which could be enhanced by an AR system has two main components, perceptual and cognitive. The perceptual component of the task is the type of perceptual action that the soldier needs to perform, while the cognitive component describes the necessary mental processes. For each task we analyzed how an AR system could support both the perceptual and cognitive components. Figure 8 demonstrates this analysis.

5.3. User-Centered Requirements

We analyzed the use case documentation to produce user-centered requirements; Figure 9 shows a portion of the resulting list. This list is really the final outcome of domain analysis activities; this is what we can give the system engineers. However, this list is far from complete, because all of the steps in the usability engineering process will yield additional requirements.

We discovered that producing the user-centered requirements drove an important design decision on our part. We realized that our user-centered requirements identified a list of features which cannot be easily delivered by any current AR system. Therefore our development team has decided to take a step back and conduct some basic research and development behind these requirements. For example, the last item says that the system must be able to display the location of hidden and occluded objects (for example, an building located behind a visible building). How should such objects be depicted graphically? Some design ideas are dotted or blurred outlines, but until usability evaluations are conducted, all such designs are speculative.

In addition, we discovered important bounds on system requirements. For example, the use case showed us that while tracking has to be good enough to accurately position graphical indicators on buildings and streets, it does not have to be better than this. This bound is important, because extremely accurate tracking is extremely difficult.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have described a usability engineering process, which we have applied within the context of BARS, a mobile, outdoor, augmented-reality system for use in urban warfare. To the best of our knowledge, this work is the first time a systematic user-centered design process has been applied to an AR system. In particular, this paper focuses on the domain analysis phase of the usability engineering process. We discuss both the *process* of domain analysis, as well

as the *products* which domain analysis has yielded to date. The products of the process are helping to guide ongoing software and user interface development efforts. We plan to continue applying the usability engineering process to BARS development, and expect to be performing preliminary heuristic evaluation on forthcoming prototypes. As the prototypes mature, we expect to conduct usability evaluations employing representative users performing tasks derived from the domain analysis activities (i.e., use cases).

REFERENCES

1. F. Brooks, *The Mythical Man-Month, Anniversary Edition: Essays on Software Engineering*, Addison-Wesley, 1995.
2. D. Hix and H. Hartson, *Developing User Interfaces: Ensuring Usability through Product & Process.* New York, John Wiley and Sons, 1993.
3. J. Gabbard, D. Hix, J. Swan II, "User-Centered Design and Evaluation of Virtual Environments", *IEEE Computer Graphics and Applications*, 19(6), Nov / Dec 1999, 51-59.
4. D. Hix, J. Swan II, J. Gabbard, M. McGee, J. Durbin, T. King, "User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment", *Proceedings IEEE Virtual Reality '99*, IEEE Computer Society Press 1999, 96-103.
5. "A Concept for Future Military Operations on Urbanized Terrain", Concepts Division, Marine Corps Combat Development Command, July 1997.
6. R. Azuma, "A Survey of Augmented Reality", *Presence: Teleoperators and Virtual Environments*, 6(4), 1997, 355–385.
7. R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent Advances in Augmented Reality", *IEEE Computer Graphics and Applications*, 21(6), Nov 2001, 34–47.
8. A. Stedmon, R. Kalawsky, K. Hill, and C. Cook, "Old Theories, New Technologies: Cumulative Clutter Effects Using Augmented Reality," *IEE International Conference on Information Visualization '99*, London, UK, July 1999.
9. S. Julier, Y. Baillot, M. Lanzagorta, D. Brown, L. Rosenblum, "BARS: Battlefield Augmented Reality System", *NATO Symposium on Information Processing Techniques for Military Systems*, 9–11 October 2000, Istanbul, Turkey.
10. Col. M. Dewar, *War in the Streets: The story of urban combat from Calais to Khafi*, David & Charles, 1992.
11. A. Beevor, *Stalingrad: The fateful siege 1942-1943*, Viking Press, 1998.
12. N. Warr, *Phase Line Green: The Battle for Hue*, Naval Institute Press, 1997.
13. M. Bowden, *Black Hawk Down*, Atlantic Press, 1999.
14. *An Infantryman's Guide to Combat in Build-up Areas*, FM-90-10-1, US Army Field Manual, US Army, 1993.
15. *SWAT Team Manual*, US Army Field Manual, FM-1698, US Army, 1994.
16. *SEAL Combat Manual*, US Naval Special Warfare Center, US Navy, 1974.
17. *Hostage Rescue Manual*, L. Thompson, Greenhill Books, 2001.