OrMiS: A Tabletop Interface for Simulation-Based Training

Christophe Bortolaso[1], Matthew Oskamp[1], T.C. Nicholas Graham[1], Doug Brown[2]

[1]School of Computing
Queen's University
Kingston, ON, Canada

[2] Directorate of Land Synthetic Environments
CFB Kingston
Kingston, ON, Canada

(christophe.bortolaso, 7mao, nicholas.graham)@queensu.ca, dgbrown62@gmail.com

ABSTRACT
This paper presents the design of OrMiS, a tabletop application supporting simulation-based training. OrMiS is notable as one of the few practical tabletop applications supporting collaborative analysis, planning and interaction around digital maps. OrMiS was designed using an iterative process involving field observation and testing with domain experts. Our key design insights were that such a process is required to resolve the tension between simplicity and functionality, that information should be displayed close to the point of the user's touch, and that collaboration around maps cannot be adequately solved with a single form of zooming. OrMiS has been evaluated by domain experts and by officer candidates at a military university.

Author Keywords
Tabletop, Military, Simulation, Interaction Design, GIS.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous

INTRODUCTION
Maps have been used for thousands of years to depict topographic information. Digital tabletop surfaces naturally support collaborative analysis and planning around maps, enabling natural communication using pointing and gestures, and also allowing interactive zooming, searching and modification of the map's contents. Collaborative use of maps has been explored using multimodal interfaces [16] and digital tabletops in numerous domains such as urban planning [8], emergency response [10,18] and maritime command and control [21].

Despite this considerable interest, there are few detailed explorations of the design of tabletop applications supporting map-based tasks. To address this lack, this paper presents *OrMiS*, a multi-surface environment for simulation-based training. In simulation-based training, military officers use a map-based tool to carry out strategic manoeuvres and combat, enabling large-scale training exercises without the cost of field deployment.

ITS'13, October 6–9, 2013, St. Andrews, United Kingdom.
ACM 978-1-4503-2271-3/13/10.
http://dx.doi.org/10.1145/2512349.2512792

Figure 1. Using OrMiS

OrMiS (short for *Orchestrating Military Simulations*) was developed in collaboration with practitioners in military simulation. OrMiS is designed to replace traditional PC-based simulation tools while improving ease of learning and better facilitating collaborative work. The design was informed by observation of simulation-based training sessions involving dozens of participants. OrMiS has been evaluated by domain experts and by officer candidates at a military university.

As shown in Figure 1, OrMiS enables small groups of people to discuss unit dispositions, analyze terrain and develop strategy while gathered around a digital tabletop displaying a map. Units are moved by dragging them with a finger. Individuals can focus on one part of a map using a bi-focal lens [1]. A unit's past and future locations are shown through trace lines, and units' visibility and range are shown using overlays. Secondary displays show radar views. Multiple tables can be connected by a network, e.g. to support multi-team war-gaming.

Our design process involved iterations of field observations, design, implementation, and evaluation with expert users. From this process, we learned four important lessons for the design of map-based tabletop applications:

- *The design is particularly sensitive to the requirements of the domain.* Our initial understanding of the users' tasks was incorrect in small ways that critically impacted the eventual design of OrMiS. For example, the pace of activity was much slower and more deliberate than we expected, and planning activities were far more important than combat.
- *There is significant tension between simplicity and functionality.* This leads to the need for parsimonious

design, where features are added to the system only when truly needed. Current PC-based simulation tools have a profusion of features, and are difficult to learn and use. An example of this design tension is that in the specification of routes for units, we initially used a simple drag operation, which proved to inadequately support editing of routes in progress, and was replaced by a more complex waypoint-based interaction technique.

- *Visual feedback should be presented at the user's point of touch.* Users find it difficult to carry out touch actions in one location while attending to visual information in another location. The current PC-based applications rely on multiple windows to present data; for example, terrain elevation for a given point on the map is shown in its own window. Conversely, OrMiS shows the line of sight of a selected unit directly on the map, visually localizing the information to the point of touch.
- *There is more to collaboration than co-locating people around a table.* In OrMiS, users work largely independently, on occasion consulting to plan and coordinate their actions. Even while working independently, users retain awareness of other users' activities. OrMiS supports these activities using bi-focal lenses to support parallel, independent work, and radar views to allow retention of global context.

While similar lessons have been discussed in earlier HCI research, this paper is among the first to explore these issues together, in the context of a real tabletop application.

We first describe our application domain of simulation-based training and report the problems with current technology. We review previous map-based applications for digital surfaces. We then describe interesting choices in the design of OrMiS, and present feedback on OrMiS' usability. Finally, we conclude the paper by summarizing our lessons learned and the challenges in the design of map-based application on digital tabletop surfaces.

SIMULATION-BASED TRAINING

Western militaries make extensive use of simulations to train officers in planning and execution of military operations. An example training scenario might involve a

Figure 2. Interactors communicate with trainees by radio, and carry out operations using a PC-based simulation tool.

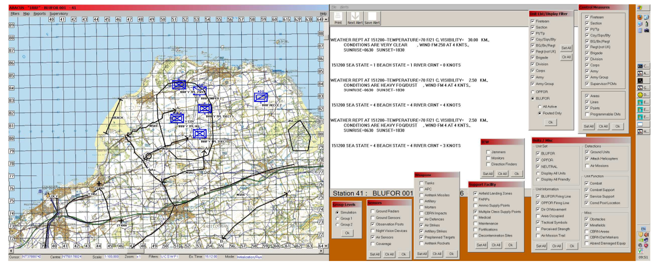

Figure 3. The ABACUS interface

military battalion (up to 1,000 troops) advancing toward and taking a military objective. In this scenario, the *trainees* are the officers working in the battalion command headquarters. The trainees communicate with officers in the field via radio, and view the status of the operation using feeds from unmanned aerial vehicles.

The "officers in the field" are actually role-played by retired military officers (called *interactors*) [19]. As shown in Figure 2, interactors sit in a room, using simulation software running on PCs to move around the virtual troops under their command. Interactors use radio (and other communications means) to speak to the trainees (who are located in a simulated headquarters). The interactors respond to orders and deliver situation reports as if they were actually in the operational theatre.

The simulation software allows interactors to mimic troop movement and combat engagement. Two popular simulation packages are ABACUS (cf. Figure 3) and JCATS [27]. As shown in Figure 3, simulation tool user displays are composed of a main area showing a map with a large set of accompanying controls. The units are displayed directly on the map using traditional military symbols. Interactors typically arrange the additional controls as a set of small windows on a secondary display. Controls allow users to set the orientation, heading, and rules of engagement of units, to organize the units' hierarchy, and to perform combat. Each interactor is in charge of a set of units, typically split according to the units' military hierarchy.

As we shall discuss below, there are two problems with the use of current simulation interfaces to support interactors. First, the tools are difficult for interactors to learn, requiring expensive training (and re-training) before each exercise. Second, current tools inadequately support collaboration among interactors. OrMiS addresses these problems through the use of a touch-based tabletop interface.

Field Observations and Task Analysis

We learned about the domain of simulation-based training by attending three simulation-based training exercises at the Army Simulation Center, part of the directorate of Land Synthetic Environments of the Canadian Forces, and through extensive discussions with military personnel and contractors responsible for designing and conducting simulation-based exercises.

Simulation tools superficially resemble real-time strategy games such as StarCraft, where players frenetically move units over a map while attempting to defeat other players. This resemblance is misleading. Simulated exercises are slow-paced, often running over several days. The simulations are literally real-time: it might take an hour for a simulated tank squadron to travel 20 km once the interactor has issued movement directions. Interactors frequently have a book or newspaper to keep them occupied during long periods of inactivity.

Most interactors are retired military officers, typically working about 20 days per year. The ABACUS and JCATS simulation tools are complex and hard to learn. Interactors require re-training each time they begin a new exercise, dramatically increasing the cost of running exercises. The simulation centre staff repeatedly emphasized their desire to have a simulation tool that is easy to learn.

During exercises, interactors are collocated in a large room (see Figure 2). Each interactor sits in front of a PC, and uses simulation software to control a set of units. In the back of the room, a set of screens displays the global map state. In the middle of the room (not shown in Figure 2), a large paper map is placed on a table (called a "bird table"), with small paper icons to represent units' positions. Interactors primarily use this table to collaboratively plan the simulation before it begins.

Activities

Interactors engage in three main activities:

1. *Assessment*: Interactors spend most of their time assessing of the state of the battle and analysing terrain. Interactors attempt to achieve their mission objectives while minimizing the risk for their troops. For example, determining a route that units will take to a target location depends on topography (avoiding impassable terrain), known enemy locations and line of sight of units (to avoid ambush). One interactor told us, "*We spend 99% of the time focused on the map*".
2. *Planning*: Once the situation is assessed, interactors plan how to carry out the orders they have received from the trainee headquarters. The majority of orders are related to unit movement. Planning is slow and deliberate. The interactors make great use of contour lines provided on the maps to determine visibility of units and plan safe routes through hostile terrain. Interactors rarely use the line of sight (LOS) tools provided by the simulation environments, reporting them as difficult to use.
3. *Giving Orders*: Once a plan has been created, interactors enter orders into the simulation tool. For example, a route for a unit is created by clicking and dragging a poly-line sequence of waypoints. Since it may take many minutes or even hours for units to follow a route, interactors are not under time pressure, and may take considerable time to plot the route correctly.

Collaboration

We observed that interactors continually coordinate their actions. For example, an attack might require troops controlled by two interactors to arrive at a given location at the same time, or one interactor might initiate an artillery barrage some minutes before another launches an attack. Collaboration is required to ensure that the simulation provides a high quality of the training, for example ensuring that it is neither too easy nor too difficult for the trainees to win a battle as it progresses.

Collaboration may be *explicit,* where one interactor calls across the room to another, or stands up from his desk and walks to the desk of another interactor. Sometimes, collaboration is *implicit,* where observation of other interactors' activities is sufficient to cue a coordinated action. Interactors typically have decades of military experience, and so awareness cues such as observing the path that another interactor has chosen can be sufficient to convey an over-all strategy.

In practice, interactors do call across the room to collaborate, but this limits them to simple discussions. Interactors rarely walk to another interactor's workstation, and when they do, the collaboration is limited by the fact that when away from their own desk, they can no longer interact with the simulation tool. Implicit collaboration is used on an ongoing basis, but an individual interactor's view tends to be focused on one part of the battlefield, and often filters other interactors' units. This limits the ability to easily follow other interactors' actions.

Any complex collaborative planning is performed at the bird table before the exercise begins. Since the bird table uses physical maps and tokens, it is rarely updated once the exercise is underway, limiting its usefulness.

Problem Statement

We identified two main problems with the existing simulation interface:

High learning curve: Both the ABACUS and JCATS simulation tools are tremendously difficult to learn. Several days are required prior to each exercise to train interactors in using the interface, even when those interactors have used them in previous exercises.

Weak support for collaborative tasks: The poor support for both explicit and implicit collaboration leads interactors to miss opportunities to plan and coordinate their activities.

As we shall see, our OrMiS tool addresses these issues by providing a digital tabletop interface for simulations. OrMiS' simple touch interface is easy to learn, and supports small-group collaboration through a shared map.

BACKGROUND: TABLETOP MAP INTERFACES

To inform OrMiS' design, we considered existing tabletop applications supporting analysis, manipulation and collaboration around maps. While there are few complete tabletop applications in this domain, there has been significant work on showing how the requirements of different map-related domains can be met using a digital tabletop interface. These domains include emergency response, utility grid management, geological exploration and military command and control.

Emergency Response

Emergency response involves detection and monitoring of emergencies, and deployment of resources to combat the emergency. Most emergency response systems are based on a large map of the area being monitored. Several proof-of-concept systems have been developed to explore the design of tabletop applications for emergency response. These include the Tangible Disaster Simulation System [10], which allows multiple users to create disasters on a map, the LIFE-SAVER project [15] for response to flood disasters, and the uEmergency project [18] for forest fire response. All these projects highlight the value of using digital surfaces in emergency management training, but are best viewed as technology demonstrations rather than comprehensive attempts to support the requirements of the application domain.

The MUTI project (Multi-User Tangible Tabletop Interface) aims to improve the efficiency of emergency response using a digital tabletop [14]. Users found MUTI to be useful and easy to use. The authors discuss design issues such as the size of the table, orientation problems, and the value of deictic referencing, but do not report the design of the application itself.

Utility Grid Management

The eGrid project [23] focuses on the management of the electrical grid of a city. eGrid provides personal windows, allowing users to concurrently work on different part of the same map. The system is limited to the annotation of maps, thus supporting a small part of the problem of managing an electrical grid.

Geological Exploration

The Skyhunter Exploration Project helps users in finding petroleum reservoirs [2]. This system was designed to demonstrate the concept of transferring a view of a map from a tabletop surface to a handheld device, and does not attempt to fully support geological exploration tasks.

Military Command and Control

While early work explored the use of multimodal interfaces to simplify military command and control [4], digital tabletops have recently received considerable attention in this domain. Command and control involves the assessment of battlefield conditions and the deployment of resources to meet military objectives.

For example, the Digital Sand Table is multi-touch tabletop system supporting command and control [24]. The authors observed well-known tabletop interaction issues such as the lack of user identification and ownership [22] and the orientation problem [6]. This system is limited to the display and annotation of static maps, therefore providing rudimentary support for command and control.

More recently, the ASPECTS system illustrated the use of tabletops for planning and executing naval operations [21]. ASPECTS enables real-time monitoring of ships' locations on the map. In addition to being one of the first tabletop systems for command and control in maritime operations, ASPECTS explored key issues in tabletop interaction such as orienting menus and windows to the user's perspective, and providing role-based interaction based on user identification. ASPECTS represents, to our knowledge, the only other tabletop-based application for monitoring and analysis of maps that has been designed based on deep understanding of the application domain. ASPECTS differs from OrMiS in that it is primarily focused on planning and monitoring, not on directing the activities of units.

Summary

These existing applications have served to show that digital tabletops are a natural vehicle for collaborative map-based applications. Small groups of users can share the representation of the map on the table, similar to the use of the physical bird table by interactors in simulation-based training. Movement of units can be naturally specified using touching and dragging. The applications have helped to demonstrate the importance of correctly orienting information and of tracking ownership of objects on the map, and have shown the potential of multi-display environments using hand-held tablets for private views.

With the notable exception of ASPECTS, however, these systems are best viewed as early-stage prototypes that were not based on a deep examination of the application domain. While the contributions of these systems are important, and without them our work would not have been possible, they do not fully answer the question of what design issues arise in map-based applications for tabletops. Through its basis on field observations and its iterative design and testing with professionals in simulation-based training, OrMiS helps to fill this gap.

DESIGN OF ORMIS

OrMiS allows small groups of interactors to collaboratively orchestrate a simulation-based training exercise. As shown in Figure 4, the interface displays a topographic map on an interactive tabletop. To accommodate larger numbers of interactors, multiple tables can be connected by a network.

The map is shown from a top-down perspective with 1 km gridlines. Military units are depicted using standard NATO symbols. Units can be moved around the battlefield by touching and dragging them to create routes. Routes show the path the units are following, and can be edited by manipulating their waypoints.

Combat begins when units move within range and visibility of each other, respecting the rules of engagement for each unit type. As shown in Figure 4, the direction of fire is shown using a yellow arrow. Visibility and attack range are

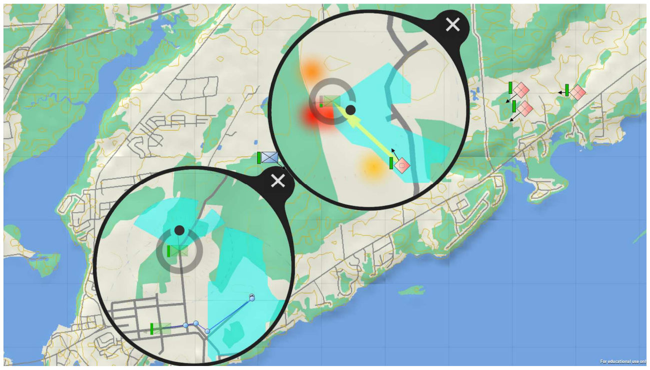

Figure 4. The OrMiS Interface

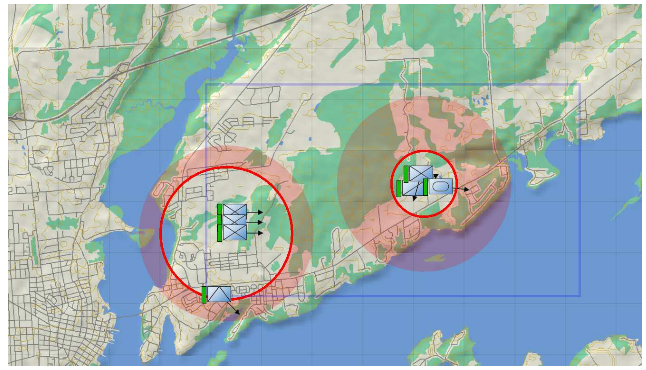

Figure 5. The OrMiS' radar view interface

displayed to the user by overlays on the map. The line of sight overlay is shown when a user selects a unit or sketches a route. The tool shows an arc where visible portions of the map are colored in. Users can change a unit's heading by selecting and rotating it.

Users can globally zoom the map by pinching, and can create zoomed local contexts using bifocal lenses. Lenses are created by a long touch on the open map and can be moved by dragging them with a single finger. The contents of lenses can be zoomed using a pinch gesture. Secondary displays show a radar view of the battlefield (Figure 5). The radar view displays the entire map with all units and routes. A blue rectangle represents the area of the table which is currently being viewed and red circles show where the lenses are positioned on the table. Interactors can use these radar views to retain spatial context, or to see where their colleagues are working without disturbing them.

Implementation

OrMiS runs on a PQ Labs G4S 55" multi-touch table. The software was implemented in C# using the Unity engine. OrMiS is fully compatible with Windows 7/8 and TUIO multi-touch inputs. The maps of OrMiS are generated using the InterMAPhics GIS [11]. Multiple surfaces are synchronized over a network using the Janus toolkit [20].

DESIGN CHOICES IN ORMIS

The goal of OrMiS was to achieve ease of learning and effective support for small-group collaboration. In addressing these goals, we repeatedly faced three design issues: balancing simplicity and functionality; finding ways to provide information at the point of the user's touch, and allowing people to work together in the presence of mixed-focus collaboration [25]. Throughout the design process, we found that deep understanding of the task domain was critical to making correct decisions. We now illustrate these issues through concrete examples from the design of OrMiS.

The Tension between Simplicity and Functionality

Interfaces designed for experts frequently provide rich and complex features at the expense of ease of learning and simplicity of operation. Examples of interfaces for experts include Adobe's Photoshop for graphical designers, or Microsoft's Visual Studio for software developers. As we have seen, tools such as ABACUS are also richly complex and intended for expert use. Interactors typically have several decades of military experience, which makes them expert at tactics, but they are largely not experts in computer use. The approximately 20 days per year that they work is not sufficient for them to retain expertise in simulation tools. As a consequence, there are significant training and re-training costs before each exercise, and the simulation tools are rarely used to their full potential. In the design of OrMiS, this led us to a tension between simplicity and functionality, where it was necessary to identify the features that were truly necessary, while keeping the interface as simple as possible. We followed a principle of *parsimonious design,* where by default, we adopted the simplest possible design of each feature, only adding complexity when it was proven necessary.

We illustrate this tension with the design of the route planning technique in OrMiS. As we have seen, route planning is used in almost every task during a simulation exercise, including attacking or defending an objective, rendezvousing with an ally, determining the best supply lines, and moving to a target location without being spotted by the enemy. We designed and tested three versions of the route planning interaction technique before finding a successful solution. We started with a simple free-form drawing technique, then moved to a free-form waypoint-based technique, and finally adopted a more complex polyline waypoint system. This design progression was informed by usability testing and by increased understanding of the context of use.

Solution A: The Free-Form Route Planning

Motivated by our goal of ease of use, our initial technique was based on free-form drawing. As shown in Figure 6.A, users simply drag their finger out from a unit to start drawing a route. The user's finger traces a free-form path, specifying the route that the unit will follow. The unit automatically starts following the path.

From early testing sessions, we observed that users had no problem learning and using this route planning technique.

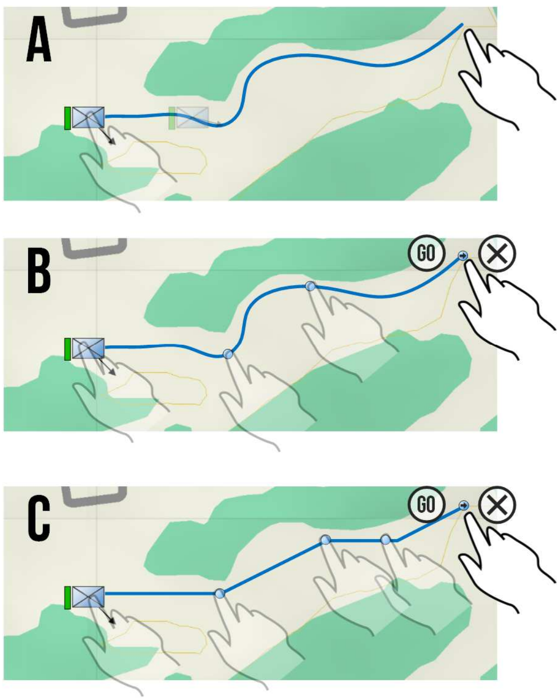

Figure 6. Illustration of the three versions of our route planning technique

However, as our tests adopted more realistic scenarios, we discovered significant shortcomings. Interactors spend a considerable amount of time defining routes. Every detail in the route is deliberately chosen, and specifying the right route requires gathering data about the terrain and other units' positions. We observed that interactors modify the route repeatedly. Our free-form approach did not provide for undoing the route, short of cancelling it and starting again, and therefore was frustrating and cumbersome to use in realistic scenarios.

Solution B: Free-Form Step-by-Step Route Planning

Our second version of the route planning technique, illustrated in Figure 6.B, introduced a limited form of undoing, making the technique more usable. The approach combined free-form drawing with a waypoint system. Users could lift their finger off the table while drawing a route. This introduced a waypoint. To extend the route, users would drag from the last waypoint. Tapping on the last waypoint displayed a small pie menu, allowing the user to delete the last route segment. Performing this action repeatedly allowed any number of route segments to be deleted.

Expert users found this technique more difficult to learn than the first one, as the waypoint interaction required a few minutes of instruction and training. More importantly, they reported that in a realistic simulation, they would prefer to be able to edit the route at any point, and not be restricted to undoing and redrawing only the last segment of the route. We found that some interactors preferred to initially draw approximate routes and then refine them. At this point, it became clear that it was necessary to introduce more functionality, despite the cost to simplicity.

Solution C: Polyline Route Planning

The third route planning technique, illustrated in Figure 6.C, is based on a polyline drawing system. As before, users drag the next step in the route with their fingers, release to create a waypoint, and then touch and drag from the last waypoint to create the next segment. Any waypoint can be dragged to modify the route, and waypoints can be deleted by tapping them and selecting deletion from a pop-up pie menu. Polylines are used instead of the earlier free-form drawing, as the semantics of moving a waypoint on a free-form line are not clear. This approach allows arbitrary editing of the route, better matching the interactors' workflow, but at significant cost in simplicity. Our experience, as reported later in the paper, shows that the full functionality of the polyline drawing is regularly used, and that the training time is still measured in minutes, making it clear that in this case giving up simplicity for functionality was justified.

Interesting Design Alternatives

In the early stages of our design, we considered and experimented with two other forms of input for route planning. We determined that both were not feasible with current technology.

Our initial input technique for routes was stylus-based, using Anoto technology. Anoto pens provide more precise input then touch and enable user identification. However, in practice, the pen needs to be held close to upright to function correctly, which many users found difficult. In addition, some users were confused by a parallax effect caused by the 1/8" distance between the tabletop surface and the projection surface. These problems don't exist with the PQ Labs touch technology. Finger-touch interaction also brings the benefits of traditional gestures for zooming and panning. It will be interesting to revisit this decision as stylus technology for large interactive surfaces improves.

We considered automation of the route planning process, such as in real-time strategy games where players specify only the target position, and the game determines the best route to move to that location. There are two significant barriers to such automation, and as a consequence, simulation tools such as JCATS and ABACUS do not use it. First, determining the best route is a complex problem involving consideration of terrain, relative unit strengths and state of repair, unit morale, and anticipated location and strength of the enemy. Often there is no single best answer, and therefore human input is required in the process. In addition, maps for a given theatre are frequently incorrect and incomplete, for example, missing the presence of a critical bridge, or failing to show small areas of impassable swamp. Small errors in the underlying maps can lead to very poor route choices, again implying the need for human control of the process.

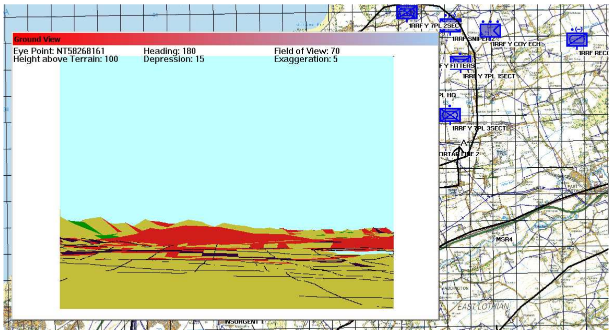

Figure 7. ABACUS Line of Sight Visualization interface

Figure 8. OrMiS Line of Sight Visualization interface

The Tension between Simplicity and Functionality

Our experience in designing an interaction technique for route planning illustrates the tension between simplicity and functionality. Our initial designs erred overly on the side of simplicity, and had to be extended in the face of user testing and deeper understanding of interactors' work practices. Nonetheless, our parsimonious design principles were effective. The final design can be learned in minutes, and its features are fully used in practice. This is in sharp contrast to the equivalent features in ABACUS and JCATS which require significant training.

Focusing Visual feedback at the user's point of touch

The ABACUS simulation tool (as shown in Figure 3) makes extensive use of secondary dialogues and views. Typically, one monitor is used to show the map, and a second monitor shows a profusion of menus and controls. Users frequently change focus between the main map and the alternative views on the secondary display.

For example, a frequent operation is to determine what part of the terrain can be seen from a particular location. For example, when establishing units in a defensive position, interactors need to examine sightlines to ensure that units can see all possible avenues of enemy attack. In ABACUS, users first select a line of sight tool from a menu, select a unit to use as a sight reference, and then select parameters (such as distance and direction of sight). A line of sight window pops up showing a 3-D view of what can be seen from that location (Figure 7).

In contrast, it is natural with touch-based systems to focus attention at the user's fingers [12]. This is because touch interaction is fluidly mixed with gesturing, and because when collaborating around a shared display, the target of controls on secondary displays can be unclear. Therefore, one of our major design goals was to remove the need for secondary controls, instead placing all relevant information at the point of the user's touch.

Figure 8 illustrates how OrMiS addresses the line of sight problem without requiring a secondary display. When a user touches a unit, the area that is visible to that unit is shown on the map as a pale blue transparent overlay. Similarly, as the user drags a route, the line of sight displays moves with the user's finger. The advantage of this approach is that it integrates line of sight directly into the route planning process, and provides rapid and dynamic exploration of terrain visibility compared the cumbersome process in ABACUS.

Another surprisingly useful example of localizing information around the user's touch is the terrain feature feedback system. When touching the map, a textual tag appears below user's finger, displaying the terrain type at the touch point. As we shall see, users found this feature useful in avoiding difficult terrain.

The design of the line of sight feature also illustrates the tension between simplicity and functionality. By default, line of sight is considered to be 360°. In practice, people are most likely to see objects that are in front of them. Some training exercises therefore require interactors to specify the direction in which units are looking. As shown in Figure 8, this functionality is provided in OrMiS by tapping a unit (selecting it), causing a circular handle to appear that allows the direction of view to be defined. In testing, this feature led to mode confusion, as the unit had to be de-selected to allow route specification. Experienced users learn to notice the selection mode of a unit, but such mode confusions introduce undesirable complexity.

Through these examples it can be seen that presenting visual feedback at the user's point of touch is important in direct manipulation interfaces, and can simplify traditional interfaces where secondary windows are required. However, there is a danger of interfaces becoming more complicated than simple touch interaction can easily support, leading for example to mode confusion or to information overload on the screen.

There is more to collaboration than co-locating people around a table

As we have discussed, interactors need to collaborate during exercises. Collaboration can be explicit (verbally negotiating when to launch an attack), or implicit (watching another interactor's route planning and deducing the underlying strategy). Interactors often use a bird table before the exercise begins to plan global strategy, and then communicate during the exercise by calling across the room and by watching other interactors' actions on their own display. It is difficult for interactors to communicate explicitly during an exercise without leaving their workstations, and in practice, they rarely do so.

In OrMiS, we aimed to improve collaboration by enabling small groups of interactors to work together around a digital tabletop. We found two fundamental issues with tabletop-based collaboration. First, interactors working together often need to view different parts of the map at different levels of detail. Second, the number of interactors supporting a training exercise often exceeds the number that can practically stand around a table. The first problem is solved through a combination of bi-focal lenses and radar views. The second is addressed by connecting multiple tables with a network. To our knowledge, OrMiS is the first touch-based application to combine these techniques as a comprehensive solution for supporting collaboration around map-based tasks.

A Combination of Overview+Detail and Focus+Context

Two common techniques have been proposed for supporting collaboration over multi-scale visual spaces such as maps. *Overview+detail* techniques use multiple windows to present zoomed and global views simultaneously [9,13], for example based on a radar view together with a detailed main view. *Focus+context* techniques provide zoomed regions (lenses) overlapping the main context [3,17]. The latter technique has been explored in detail on tabletop surfaces [2,5,7,26], showing that lenses can efficiently support different types of collaborative coupling [25].

OrMiS uses both techniques for different purposes. As shown in Figure 4, the table shows the complete map. The map can be zoomed using a standard pinch gesture. *Bi-focal lenses* can be created to provide focus. These lenses provide a circular area that can be zoomed independently of the map itself. When users are working on separate parts of the map, each can use a lens to focus on their work, while retaining global context through the base map. Lenses retain their position on the map even when the main map is zoomed.

Finally, radar views show a small representation of the entire map including units and important objectives (see Figure 5). This provides the global context view seen at the back of the interactors' room. The radar view supports coordination and mutual awareness between interactors, allowing interactors to precisely locate where a colleague is working in the map without disrupting their work.

Providing this variety of techniques has the potential to confuse users, once again requiring us to navigate the tension between simplicity and functionality. As we shall see, users in fact have no difficulty working with a combination of all techniques.

Networking Multiple Tables

To support larger groups of interactors, multiple tables can be connected by a network. For example, in a war-gaming scenario, units for the two sides of the conflict can be controlled on different tables. Unit visibility can be respected, so that for example, a "red" unit only appears on the blue team's table if at least one "blue" unit has line of site to the red unit. Alternatively, control of units may be split among tables based on military hierarchy or geographical proximity.

Interesting Design Alternatives

A significant known problem with bi-focal lenses is that they occlude part of the underlying map, which, as we have discussed, increases the difficulty of maintaining context. Fisheye lenses can reduce this problem by scaling the map content at the edge of the lens rather than hiding it [3]. The first version of OrMiS used fisheye lenses, but these were rapidly abandoned as the distortion in fisheyes proved unacceptable in military planning tasks.

EXPERIENCE

In our design process we solicited regular feedback from numerous domain experts, including senior military officers who perform simulation-based training and developers of a command and control application. In order to gain experience beyond this group, we invited six pairs of officer candidates from a military university to use our tabletop application to perform a realistic command and control scenario. All participants held the Basic Military Officer Qualification–Land qualification, requiring knowledge of the topographical standards used in military maps, as well as basic troop deployment strategies.

The scenario, shown in Figure 9, was designed in collaboration with senior military officers. In the scenario, one participant controls armoured units located at 1A, and the other controls infantry units located at 1B. Their first task was to rendezvous at position 2. They were then to move through hostile territory to the goal position 3, with the infantry flanking the armour in order to flush out any enemies located in the woods.

Overall, participants' comments about the system were highly positive, indicating that they found OrMiS easy to use. Participants showed no difficulty with even the most complex interaction techniques, including the route planning technique. One participant stated, *"I really liked the table, it was very intuitive"*.

Participants also expressed a desire for more advanced functionality. As one participant stated *"I really agree with the KISS* [Keep It Simple, Stupid] *principle, but as you get*

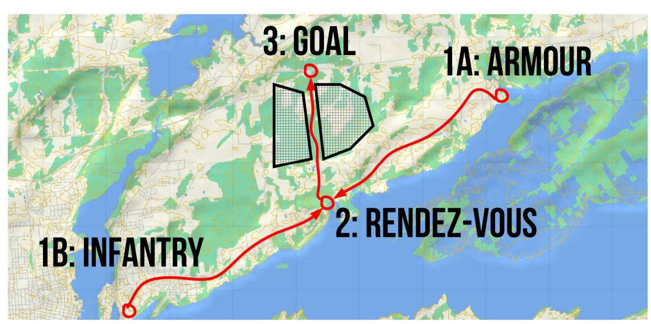

Figure 9. Scenario used during the study.

up into the higher levels they are going to need more specific stuff, like this is a rendezvous point and there is potential minefield close by". Several participants requested a means of annotating the map to visually demarcate areas and points of interests. Some participants also expressed the need to have more advanced control of the units, such as being able to create formations. These features will be added to future versions of OrMiS.

The localized feedback was also well received by the officer candidates. One stated, *"what I really liked was when I touched somewhere and it would tell me what it was."* The participants used the lenses for loosely coupled interactions, and used the table zoom when working in proximate areas. One participant said *"We preferred the lens in separate areas, but then zoomed the whole map when we were together"*.

Finally, we saw that the radar view was also used for high level planning. The radar view was not owned by any user, and thus served as reference for the global state of the map.

DISCUSSION AND LESSONS LEARNED

We now return to the four major design issues that we identified in the introduction to the paper.

There is significant tension between simplicity and functionality. Existing simulation tools are complex and feature rich, and are intended for expert users. Tabletop applications more typically provide simple interaction, at the cost of functionality. Given that interactors are typically domain experts but not experts in computer use, the simplicity of a tabletop interface is appropriate. For key features, we went through several iterations to find the appropriate balance between simplicity and functionality. This is illustrated by our examples of route planning, line of sight visualization and map viewing at mixed levels of detail. Continuous testing with domain experts was required to determine where our designs were too simple. The design was necessarily iterative as improvements in the system permitted us to use increasingly realistic scenarios. As we have seen, the resulting system is dramatically simpler than the existing PC-based tools, yet has proved capable of supporting realistic simulation-based training scenarios.

Visual feedback should be presented at the user's point of touch. As was shown in Figure 3, current simulation tools make extensive use of controls, menus and displays provided on a second monitor. Our second major design issue was finding ways of allowing users to manipulate and analyze the map while keeping their focus at the point where they are touching the table. To illustrate this principle, we showed the design a line of sight tool, and how it simplified the existing ABACUS similar feature, while providing faster and more dynamic interaction.

There is more to collaboration than co-locating people around a table. We discussed how interactors need to move between explicit communication for planning and coordinating actions and implicit collaboration based on awareness of each other's activities through the shared artefact of the map. We showed how this was possible by combining overview+detail and focus+context techniques. Despite the existence of many interaction techniques to support small group collaboration, experimental design was required to identify techniques allowing multiple people to simultaneously manipulate the map and to support switching between group and personal work.

The design of this style of application is particularly sensitive to the requirements of the domain. We learned from this experience that the design of map-based tabletop application is highly influenced by the specifics of the application domain (in our case, simulation-based training). Apparently minor aspects of the application domain can have a significant impact on the final design. We highlight four examples where this occurred:

- Interactors are domain experts, but do not work enough hours to become experts in simulation tools. Therefore, such a simulation tool should not be an "expert interface" with complex controls, but should be designed for ease of learning and use.
- Simulation exercises operate in real-time, and thus are very slow-paced. Interactors are not under time pressure, and therefore plan carefully and deliberately. The tool should therefore support finding the best decision rather than the fastest decision.
- When the military deploys to a new theatre, the available digital maps often have small but serious errors. This makes it unwise to rely on automated route planning.
- When splitting interactors between tables, one obvious approach is to separate interactors playing the friendly ("blue") team from those playing the enemy ("red") team. In fact, the interactors for the two teams often collaborate closely to ensure that the desired pacing and difficulty level of the scenario is achieved. A better split turns out to follow the hierarchy of the military organization.

These four design issues clearly indicate the importance of a parsimonious, iterative design process involving deep observation of domain experts and frequent usability testing using realistic scenarios. Our experience raises the question of how easy it is to share designs amongst tabletop applications that are in similar domains. For example, can our route planning technique or collaborative lenses be adapted for map-based applications in emergency response or geological exploration? The current lack of real tabletop applications makes such questions difficult to answer. The design presented in this paper therefore represents one crucial data-point in helping to understand the design issues in practical tabletop applications.

CONCLUSION

In this paper, we have presented and analyzed the design of *OrMiS,* a tabletop application supporting "interactors" who provide behind-the-scenes guidance for simulation-based training exercises. We showed how observation of interactors carrying out real simulation-based training informed the design of OrMiS. We discussed challenges in balancing functionality and simplicity, representing information close the user's visual focus, supporting small group collaboration. OrMiS was enthusiastically received by officer candidates carrying out a realistic scenario. Our next steps involve further assessing the practicality of using touch-based technology for simulation-based training, and researching the applicability of OrMiS to real (rather than simulated) military command and control.

ACKNOWLEDGEMENTS

This work has been supported by the SurfNet Strategic Network, the GRAND Networks of Centres of Excellence, and the Leif Transatlantic Exchange Program. We would like to thanks Kongsberg-Gallium and we gratefully acknowledge the contributions of Andrew Heard, Eric Ingle and Amir Sepasi to early versions of OrMiS.

REFERENCES

1. Apperley, M.D., Tzavaras, I., and Spence, R. A bifocal display technique for data presentation. *EUROGRAPHICS'82* (1982), 27–43.
2. Burns, C., Seyed, T., Bradley, K., et al. Multi-Surface Visualization of Fused Hydrocarbon Microseep and Reservoir Data. *GeoConvention*, (2012).
3. Carpendale, S., Ligh, J., and Pattison, E. Achieving higher magnification in context. *UIST'04*, ACM (2004), 71-80.
4. Cohen, P.R., Johnston, M., McGee, D., et al. QuickSet: Multimodal interaction for distributed applications. *MULTIMEDIA'97*, ACM (1997), 31–40.
5. Forlines, C. and Shen, C. DTLens: multi-user tabletop spatial data exploration. *UIST'05*, ACM (2005), 119-122.
6. Hancock, M.S., Carpendale, S., Vernier, F.D., and Wigdor, D. Rotation and Translation Mechanisms for Tabletop Interaction. *TABLETOP'06*, IEEE (2006), 79–88.
7. Ion, A., Chang, Y.-L.B., Haller, M., Hancock, M., Scott, S.D., and Chang, B. Canyon: Providing Location Awareness of Multiple Moving Objects in a Detail View on Large Displays. *CHI'13*, ACM (2013), 3149–3158.
8. Ishii, H. and Underkoffler, J. Augmented urban planning workbench: overlaying drawings, physical models and digital simulation. *ISMAR'02*, (2002), 203–211.
9. Javed, W., Ghani, S., and Elmqvist, N. Polyzoom. *CHI'12*, ACM (2012), 287.
10. Kobayashi, K., Narita, A., Hirano, M., et al. Collaborative simulation interface for planning disaster measures. *CHI EA'06*, ACM (2006), 977-982.
11. Kongsberg Gallium. *InterMAPhics. 2013*. http://www.kongsberg.com/en/kds/kongsberggallium/products/intermaphics/.
12. McLaughlin, A.C., Rogers, W.A., and Fisk, A.D. Using direct and indirect input devices: attention demands and age-related differences. *TOCHI 16*, 1 (2009), 1–15.
13. Nacenta, M.A., Pinelle, D., Stuckel, D., and Gutwin, C. The effects of interaction technique on coordination in tabletop groupware. *GI'07*, ACM (2007), 191-198.
14. Nayak, S., Zlatanova, S., Hofstra, H., Scholten, H., and Scotta, A. Multi-user tangible interfaces for effective decision-making in disaster management. *Remote Sensing and GIS Technologies for Monitoring and Prediction of Disasters*. Springer 2008, 243–266.
15. Nóbrega, R., Sabino, A., Rodrigues, A., and Correia, N. Visual Information Systems. *Web-Based Visual Information Search and Management*. Springer, 2008.
16. Oviatt, S. Multimodal Interactive Maps: Designing for Human Performance. *Human-Computer Interaction 12*, 1 (1997), 93–129.
17. Pindat, C., Pietriga, E., Chapuis, O., and Puech, C. JellyLens. *UIST'12*, ACM (2012), 261-270.
18. Qin, Y., Liu, J., Wu, C., and Shi, Y. uEmergency: a collaborative system for emergency management on very large tabletop. *ITS'12*, ACM (2012), 399.
19. Roman, P.A. and Brown, D. Games, Just How Serious Are They? *Interservice/Industry Training, Simulation & Education Conference*, (2008), 11 pages.
20. Savery, C. and Graham, T.C.N. Timelines: simplifying the programming of lag compensation for the next generation of networked games. *Multimedia Systems*, (2012), 1 – 17.
21. Scott, S.D., Allavena, A., Cerar, K., et al. Investigating Tabletop Interfaces to Support Collaborative Decision-Making in Maritime Operations. *ICCRTS'10*, (2010).
22. Scott, S.D., Sheelagh, C., and Inkpen, K.M. Territoriality in collaborative tabletop workspaces. *CSCW'04*, ACM (2004), 294-303.
23. Selim, E. and Maurer, F. EGrid: supporting the control room operation of a utility company with multi-touch tables. *ITS'10*, ACM (2010), 289-290.
24. Szymanski, R., Goldin, M., Palmer, N., Beckinger, R., Gilday, J., and Chase, T. Command and Control in a Multitouch Environment. *26th Army Science Conference*, (2008).
25. Tang, A., Tory, M., Po, B., Neumann, P., and Carpendale, S. Collaborative coupling over tabletop displays. *CHI'06*, ACM (2006), 1181-1190.
26. Von Zadow, U., Daiber, F., Schöning, J., and Krüger, A. GlobalData: multi-user interaction with geographic information systems on interactive surfaces. *ITS'10*, ACM (2010), 318.
27. *JCATS - Joint Conflict and Tactical Simulation.* http://www.jtepforguard.com/jcats.html.