# Not Just for Angry Birds, Practical Training with Mobile Devices

**Steven Borkman**
**Dignitas Technologies**
**Orlando, FL**
**sborkman@dignitastechnologies.com**

**Julio de la Cruz**
**US Army Research Laboratory**
**Simulation and Training Technology Center**
**Orlando, FL**
**julio.delacruz@us.army.mil**

## ABSTRACT

Targetry Range Automated Control and Recording (TRACR) is a common target controller used at live-fire Army target ranges. TRACR is used to control range targets (both lane based and maneuver). It provides the controller the ability to expose and conceal targets, set their properties, behaviors, etc. beyond individual targets. TRACR allows the controller to regulate the entire range, and the ability to create and play scenarios. Training typically consists of a single range tower governing the entire site. For lane based training, each lane has a trainer watching the student, but the trainer has no direct control over the events of the range and must constantly make necessary requests via radio to the controller at the tower. This creates a bottleneck at the tower and is inefficient from the trainer's perspective. Collaboratively with Army's Program Manager Training Devices and the Army Research Laboratory Simulation and Training Technology Center, the Team developed TRACR Ultra Lite (TÜL); an Android tablet based live training app which delivers all of the target control capabilities found in TRACR into the hands of the trainer. While using TÜL, the trainer can directly and efficiently control the trainee's exercise, allowing for a tailored training environment without overloading the tower operator. Along with lane based training, TÜL was also designed to meet the needs of maneuver range training and serve as a tool for site maintenance. This paper will discuss the abstract concept of enhancing live training with mobile capabilities. It will discuss how mobile technologies such as smartphones and tablets can be used to enhance the quality, efficiency, and safety of live training. Specific use cases of TÜL will be showcased and will describe how it was designed and developed to execute in unison with TRACR while meeting this criteria.

## ABOUT THE AUTHORS

**Steven Borkman** is a Senior Software Engineer at Dignitas Technologies with over 11 years of experience developing synthetic environment software for the simulation community. Mr. Borkman currently serves as the project lead for the High Fidelity Runtime Database Engine and TRACR Ultra Lite research projects. Mr. Borkman holds a Bachelor of Science in Computer Science from the University of Central Florida.

**Julio de la Cruz** is the Science and Technology Manager and Chief Engineer for Synthetic Natural Environment (SNE) and Simulation Technologies. Mr. de la Cruz has over 20 years' experience in government and private sector. As the SNE Chief Engineer for the Simulation and Training Technology Center Army Research Laboratory, Human Research and Engineering Directorate he oversees the research and development of emerging Database Generation tools and technologies that supports the Command and the Army current and future needs in Modeling and Simulation.

# Not Just for Angry Birds, Practical Training with Mobile Devices

**Steven Borkman**
**Dignitas Technologies**
**Orlando, FL**
**sborkman@dignitastechnologies.com**

**Julio de la Cruz**
**US Army Research Laboratory**
**Simulation and Training Technology Center**
**Orlando, FL**
**julio.delacruz@us.army.mil**

## INTRODUCTION

The smart devices market (smartphones and tablets) is exploding. According to a recent Cisco report (Cisco, 2012) the number of mobile devices will soon exceed the world's population, and will reach over 10 billion smart devices by 2016. This growth is being spearheaded by the biggest computing companies in the world, like Apple and Google.

There is a convergence of the demographic groups of active military personnel and smart device users with 54 percent of people between the ages of 18 to 24 using smartphones, and 62 percent of people age 25 to 34 (Nielsen, 2011). In short, the military is now composed of a generation of people whose use of mobile devices is second nature and who expect and rely on that type of computing power at their fingertips.

Capabilities are being introduced daily into the smart device market to entertain, educate, and provide the tools and capabilities that enhance our day to day routine. These capabilities that are so important to us in our civilian lives can also be leveraged to greatly enhance Warfighter training. Can they be used for something other than texting and Angry Birds?

The traditional computer paradigm, desktop computers plugged directly into a network and cumbersome, bulky laptops, fail to meet the portability and input interface requirements demanded by live training. Shoe-horning live training into a typical computer workstation environment causes training to make compromises that can reduce its quality, efficiency, and safety. However, smart devices excel where these traditional platforms struggle. So, how can the Army use the qualities found in smart devices to make live training better?

We are conducting an on-going effort, with the Program Manager for Training Devices (PM-TRADE) and the Army Research Laboratory's Simulation and Training Technology Center (ARL STTC) to leverage the capabilities found in modern Android Tablets to increase the safety and accuracy of live range fire training. This paper examines the concepts of enhancing live training while taking advantage of commercial off the shelf hardware such as mobile devices. We will outline current Targetry Range Automated Control and Recording (TRACR) interactions and our experience developing the new TRACR Ultra Lite (TÜL) app. We will also discuss our efforts to make Mobile technology an integral part of live-fire Army target range training and warfighter training in general.

## LIVE TRAINING USE CASES

### TRACR

TRACR is a common target controller for Live-Fire Exercises at Army target ranges (see Figure 1). TRACR is used to control both lane based and maneuver range targets, and supports the ability to control the training site, including exposing and concealing targets and setting target properties and behaviors. TRACR allows for the creation and play back of scenarios – scripted events that expose and conceal targets at a specified time.

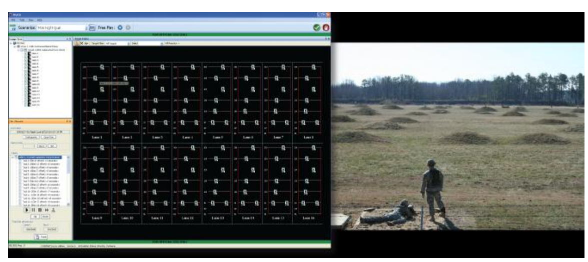

**Figure 1. Left image is the TRACR desktop application. Right image is a trainer and trainee at a lane based target range.**

A TRACR deployment typically consists of a single range tower governing the entire live-fire site. In the case of lane based training, each lane has a trainer watching the student, but the trainer has no direct control over the events of the range and must make necessary requests via radio to the controller at the tower. This creates a bottleneck at the tower and is inefficient from the trainer's perspective. Safety is also a concern in this arrangement, with the personnel closest to the live-fire having no direct control of the training activities around them.

A smart device solution delivering the control of a lane into the hands of a trainer would alleviate these issues. It would eliminate the communication bottleneck and result in safer training tailored to the individual trainee.

**ACE/OC Greenbook**

The difficulties faced in live training are not unique to TRACR. There is a general lack of portable computing, forcing trainers to, at times, rely on antiquated techniques. One example of this is the capture of training notes in the field by hand.

The Automated Crew Evaluation / Observer Controller (ACE/OC) component relies on in-the-field observers to jot down notes into a notebook (the Greenbook) and then following the training session, transfer the notes into a digital format. This is both inefficient and potentially inaccurate. Errors can be introduced in various ways between the time of the training session and the official recording.

Furthermore, using only a paper notebook and pen limits the quality and nature of the information that can be captured. Data such as photos, exact map locations using Global Positioning System (GPS), and exact times attached to digital notes would provide far greater detail and accuracy with less manual effort from observers. Capturing and relaying the information in real-time also allows for immediate After Action Review (AAR) feedback, and therefore better training for the student.

**Virtual Mortar Training**

Mortar crew training could also be enhanced with cutting edge mobile technology. The cost of training with live munitions can be prohibitive. Additionally, anytime live munitions are fired, there are implicit safety concerns. Because of these issues, the Army has developed virtual training capabilities to augment live mortar training.

Virtual mortar training is adequate for rehearsing the physical acts of positioning, aiming, and firing the mortar. The actual flyout and detonation point of the mortar can be modeled quite accurately in a virtual environment.

Where the system becomes inadequate is providing visual feedback to the observer of the impact point of the round. Right now it can be shown as an 'X' on a map, or even in a virtual reality database on a monitor. But this is not an accurate simulation of a firing exercise in the real world. The observer would watch for the visual artifacts of a detonation (smoke plume, debris) through their binoculars, and adjust the mortar based off of their observations. Again, it is another situation where the quality of training is impacted by the lack of mobility in the traditional computer paradigm.

## THE CASE FOR SMART DEVICES

The question then arises, how can training be moved away from the classroom, and directly embedded into live training environments where students could get better, safer, and more efficient training?

Smart devices are the future of computing. Their limitless mobility, in conjunction with their sensor capabilities and processing power, has the potential to revolutionize all aspects of computing in the military, including live training. The following sections enumerate the capabilities unique to smart devices and illustrate how they can be used to enhance Warfighter training.

**Portability and Form**

The quality that most separates smart devices from typical computers is their portability. For example, an iPhone 4s is 4.5 inches tall, 2.3 inches wide, 0.37 inches deep, and only 4.9 ounces. A device of this size is extremely portable and can fit into the pocket of standard issue fatigues.

The iPhone (or any smartphone device) can easily be carried and used with a single hand. Mounts are being manufactured which allow attaching phones to a Soldier's wrist (see Figure 2) or to the side of a gun.

**Figure 2. A wrist mount for a smartphone.**

According to an Army Times article (Gould 2011), a survey of Soldiers found that their preferred size is a 7 inch device (size of the Amazon Kindle Fire). A 7 inch device provides more real estate than a phone, but is still small enough to fit in a pocket.

In situations where two handed use is possible, a 10 inch tablet provides additional screen real estate for enhanced functionality or usability. A 10 inch iPad is 9.5 inches tall, 7.3 inches wide, 0.37 inches deep, with a weight of 1.46 pounds.

Of course, these devices must be able to survive the elements. Both device manufactures and aftermarket case manufactures are making great strides in ruggedization providing shock, dust, and water resistance.  The Panasonic Toughbook tablet is an excellent example of how manufacturers understand the need for ruggedization, especially for Military use.

**Interface**

Another major difference between smart devices and standard computers/laptops is their user interfaces. The standard computing interface is built around the keyboard and mouse, a layout that is hard to replicate in a mobile environment.

The multi-touch screen, found on the majority of modern smart devices, is an intuitive interface that allows a user to interact directly with the contents on the screen. Touch screens also enable single hand use and smaller form factors, removing the need for extra peripherals for user input. A standard set of practice and procedures have evolved since the advent of the multi-touch screen including pans, flings, and pinching. With the continued adoption of this technology by the public, these common interactions are becoming as familiar to the user base as typing on a keyboard or clicking a mouse. Both Android and iOS vendors provide best practices guides to developers to try to assure consistent user experiences with their devices.

**Processing Power**

Processing capabilities continue to excel in the smart device market. Modern smart devices come equipped with multi-core CPUs and dedicated Graphics Processors Units (GPUs).  For instance, the Samsung Galaxy S III, the newest smartphone from Samsung, is powered by a quad-core 1.4GHz Cortex-A9 processor. Tablets tend to be even more powerful than smartphones, allowing play of open GL based games on their large screens.

This processing power means apps targeted to Warfighter needs will not have to compromise on features or functionality. An app might provide a three-dimensional interface of an area based on satellite captured terrain data, or calculate routes through an urban environment that maximize cover and concealment from hostile locations.

**Connectivity**

Smart devices are able to interact with closed networks and the internet on the go. All smart phones, and some tablets, come with the capability to connect to the internet over cellular networks. In addition, they are capable of connecting to Wi-Fi networks and Bluetooth devices.

The real time sharing of information is critical to the Warfighter, especially in an operational context. Relaying updated locations of teammates or opposing forces could be essential to saving lives. A connected smart device could capture this intelligence and relay it to teammates and their commanding offices.

**Camera**

Almost all smart devices now come equipped with at least one on-board camera. Many devices now have an additional front-facing camera to go along with the rear-facing camera. The iPhone 4s, for instance, comes equipped with an 8-megapixel rear facing camera, with flash, that is capable of 1080p High Definition video recording, and a video graphics array quality front facing camera. When used in conjunction with the device’s GPS sensor, photos taken by the device can be geo-tagged with the exact location that the photo was taken.

A camera in the hands of the Warfighter could provide a key tool to the gathering and sharing of intelligence. For example, a camera-equipped Warfighter on patrol could capture a photo of a suspicious looking device, and have it checked for an IED or captured for reference by the next patrol.

**Sensors**

Smart devices come equipped with a suite of sensors which can be used to pinpoint the location, direction, and movement velocity of the device. This data can be accessed by applications to enhance the user’s experience, such as providing driving directions on a map, or a video game where the device is rotated in the user’s hands for input. The following is a list of sensors in the iPhone 4s (similar sensors are found in other devices): GPS and Global Navigation Satellite System location, digital compass, three-axis gyro, accelerometer, proximity, and ambient light sensors.

These sensors provide some of the most important capabilities to the Warfighter. GPS location tagging on captured intelligence could prove to be vital. In the previous photo taking example, the exact location of the item could be captured providing an explosives team the precise location of the suspicious item.

### Graphics/Augmented Reality

Multi-core processors, dedicated GPUs, and support for Open Graphics Library provide the capability of rendering 3D virtual environments. In addition to the hardware, Unreal and Unity provide third-party Software Development Kits (SDKs) to build virtual environments for mobile devices. Along with 3D graphics, these devices allow for augmented reality capabilities. Augmented reality, seen in apps such as Yelp (Figure 3), allow for the images from the camera to be overlaid with virtual information and displayed to the user.


**Figure 3. Yelp augmented reality capability (Monocle).**

## BUILDING SMART SOLUTIONS WITH SMART DEVICES

Accepting that smart devices are the next frontier, the question evolves from “can these devices be used?” to “how can these devices by used?” to meet the need of better quality, efficiency, and safety in training. The following sections re-examine the live training use cases presented earlier. We will outline possible smart device solutions for the ACE/OC and Mortar Training use cases, and present in detail the TÜL app developed for TRACR.

### ACE/OC Smart Device Solution

Portability, form factor, and interface make tablets an ideal computing device for observation and note taking on the field. A targeted capability that leverages GPS, maps, and a camera would increase the quality of data captured by observers compared with the current system of hand written notes. Exact geo-tagged and time-stamped locations could be captured for each event. The location of the team could be followed at all times on a map interface. All events and movement could be logged for AAR.

In addition, the chance for human error would be reduced. When an event is entered on the device, it is immediately saved, eliminating the inefficiency of re-entering data later. The data could automatically be transmitted over the network to a centralized server in real time.

### Mortar Training Smart Device Solution

Like ACE/OC, the quality of mortar training could be significantly enhanced by a smart device solution. An augmented reality solution could bridge the gap between live and virtual training. An observer would observe the training area through the camera display of their smart device. When a virtual detonation occurs, the camera feed would display virtual debris and smoke at the location of impact. This way the observer could continue to look through an “out the window” device, similar to their binoculars, and get feedback consistent with real world results. Using similar technology as demonstrated by Google Project Glass (Baldwin, 2012), it is not far-fetched to think that a device running Android will one day be built into a pair of binoculars, providing truly realistic virtual training.

## TÜL SMART DEVICE SOLUTION

TÜL was designed to harness the power of the mobile device and deliver the capabilities and control of TRACR directly to the trainer. Putting control in the hands of the person closest to the action increases the quality, efficiency, and safety of live range training. The trainer, with TÜL in his hands, can prepare a training scenario tailored for a particular trainee. The trainer can advance through the scenario at the trainee’s pace, and emphasize areas that are important to a particular training objective.

TÜL also provides the trainer the power to stop a site instantly. At a site with live ammunition, safety is paramount. Prior to TÜL, in an emergency situation, the tower operator was responsible for shutting down the training environment. The trainer, who is the closest person to the actual training being performed, had no access to safety override capabilities. Now, at the first sign of a hazardous situation, the trainer has the ability to shut down the entire site at the click of a button. This quick reaction to any safety concern can save lives.

TÜL runs on all modern Android tablets running Android OS 3.2 (or greater). It provides the user the ability to execute in both lane based and maneuver sites, to adjust all aspects of the target, and to control scenario playback. Details on TÜL development and the solution are outlined in sections below.

**Smart Device Operating System Considerations**

The two main operating system options for smart device solutions are Android and iOS. Early in our TÜL development process, we completed a survey to determine the optimal platform for the TÜL app. The survey was based on our teams' experience developing apps for both Android and iOS, analysis of the mobile platform environment, and customer input.

There are many considerations when comparing and contrasting Android and iOS platforms including: available hardware, cost, programming language, SDK, app delivery, and system openness. We also considered initiatives currently being conducted by PM TRADE as to their eventual goals for mobile platforms.

Although there are several advantages to each platform in many of the considered categories (such as hardware, cost, SDK, operating system) the main difference between them was the openness of the platform. iOS is a closed platform, meaning the operating system is supported only on Apple hardware. This limits availability of ruggedized army-centric hardware that can be developed in the future. App distribution must be set-up through the Apple App store or as a Developer Enterprise Program. Recent progress has been made in developing an Army App Marketplace (Army CIO, 2012) for the distribution of iOS capabilities, at the time of our analysis, the only available means of app delivery was through the standard iTunes store.

In contrast, Android is an open platform. Android releases are open source and available to be installed on a device by almost any hardware manufacturer. This opens the door for hardware manufacturers to develop specialized military-grade hardware. This might be specialized ruggedized devices that can handle the natural elements in the field, or devices with military-grade sensors such as GPS. App distribution in Android is also easier. Apps are compiled into application package distributable files. From there an app can be obtained through a marketplace, as a downloadable link from a website, or even included as an attachment in an email.

For TÜL we chose the Android operating system, primarily because of the openness of the system and the potential for greater adoption by the Army in the future.

**Smart Device Hardware Considerations**

There is an implicit tradeoff to be made with apps when determining size of the hardware, between look-and-feel/usability and portability. The larger format of a tablet (typically a 10 inch screen) provides much more screen real estate to deliver a better user experience than a phone's screen. The graphics can be nicer, the buttons larger, and more information can be displayed on a single screen. On the other hand, a sub-five inch phone offers far greater portability than a tablet. A phone is compact enough to easily fit in a pocket, and be held and used in one hand. A tablet, on the other hand, certainly cannot be carried in a pocket. It either needs to be carried, demanding the sole use of a single hand, or placed in something like a bag. Tablets also need at least two hands to use.

In the military world, equipment size and weight is very important. Today's Warfighter carries a lot of equipment. It would be hard to find any available real estate on their body for the size (and weight) of a tablet. Two handed use can also be difficult with all of the equipment that they are responsible for carrying. On the other hand, a phone (or smaller, phone-sized tablet) can be extremely portable, especially using a mounting device, as discussed earlier.

For TÜL, we chose a tablet platform, primarily because it will be used in a very specific situation at training ranges. The larger screen provides enough real estate to display the lane contents and target controls, while the added size of the tablet does not adversely affect the trainer enough to warrant a lesser user experience of a smaller screen.

To make it easier to support devices with different screen sizes Android introduced fragments as a part of the Honeycomb (3.0) operating system. Honeycomb was a rework of the standard Android operating system to meet the needs of tablet computing, including larger screens, multi-core processors, and hardware accelerated graphics. A fragment is a modular section of an Android application that can be invoked differently based on target device. This allows for users to re-use a significant amount of their application to meet the needs of both small screen and large screen devices. For example, if an application is developed using fragments, a tablet may organize all of their fragments on a single screen, whereas a phone may use the fragments serially, meaning the interaction with the first fragment may cause the display of a second fragment (see Figure 4). By developing TÜL with fragments, we minimize the effort needed to support different size screens in the future.

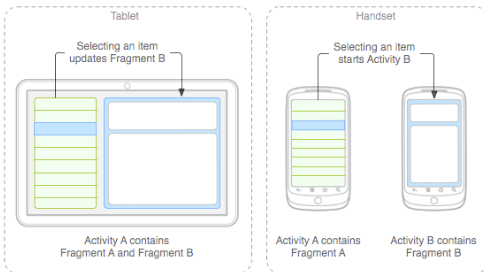


**Figure 4. Android apps utilizing fragments can support devices with different screen sizes (Android developers, 2012).**

**Architecture for Smart Applications**

With the predominance of two incompatible platforms (Android and iOS), and the necessity of meeting the unique look and feel requirements of different size devices, the development of multiple apps may be inevitable. Because of this, it is important to think in terms of cost mitigation, how can we re-use the most code and limit the amount of code rework needed for additional platforms.

A good practice in developing mobile apps is to separate the interface from the data. This is usually accomplished by structuring the app into a multi-layer architecture consisting of the interface to the user, the presentation layer, the application processing code, the business layer, and the data of the app, the data layer (Meier, Homer, and Hill, 2008).

Additionally, mobile apps typically fall into one of two categories. They are either a thin web-based client, where the presentation layer exists on the mobile device and the business and data layers reside on a network server (see Figure 5), or they are a rich client app where all of the layers exist inside the mobile app.

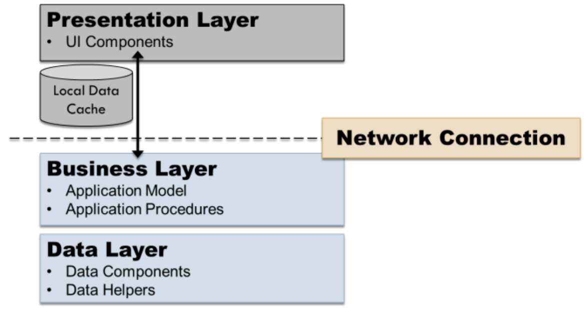


**Figure 5. Diagram of a thin client app.**

The thin web-based client app lends itself well to cross mobile platform apps. The core of the app's capabilities exist on a server, which any app can access. The developer just writes interface and networking specific code for each supported device.

The rich client app does not lend itself as well to cross platform development, but early design decisions can minimize code duplication between platforms.

iOS apps are written natively in Objective-C and Android apps in Java. Furthermore, there is no support for Objective-C in Android nor is there support for Java in iOS.

But there is some common ground, both platforms support code written in ANSI C/C++. Objective-C is a superset of C++ and Android supports native C/C++ through the Native Development Kit and Java Native Interface. By developing the majority of an app's business and data layers in C++ the same code can be used on both platforms.

However, there is really no way to completely avoid writing separate interface layers for each platform. It is the only way to have an app to look and feel natural to a user, and to integrate properly with the operating system. Fortunately, both platforms use standard image files for their graphics, allowing the app's artist to make the graphics once and allow them to be reused on each platform.

After careful consideration, we chose thin web-based client architecture for TÜL. The networked TRACR application serves as the business and data layers of the application. The TÜL app is stateless, and exists as an interface/controller for TRACR.

### The Interface

The main interface is broken into four panes (See Figure 6):

- <u>Site Contents (Pane 1)</u>: Visual representation of the targets on the site. Consists of a series of lanes for a lane based range, or a map for maneuver ranges
- <u>Objective Selection Pane (Pane 2)</u>: Allows the user to select which objective they want to focus on
- <u>Target Controls (Pane 3)</u>: An assortment of control widgets used to display and set target properties
- <u>Scenario Controls (Pane 4)</u>: Loads, plays, and stops scenarios

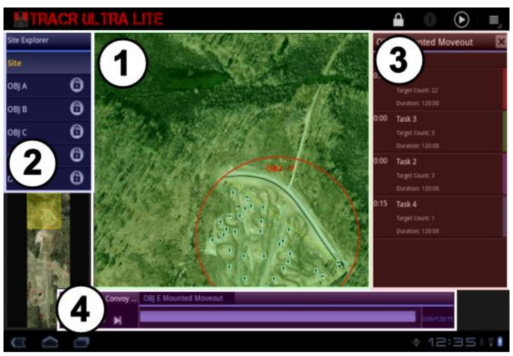


**Figure 6. Overview of TÜL user interface. The screen is divided into four panes for operation.**

The interface layer of TÜL is built on the Model-View-Controller (MVC) design pattern (see Figure 7). MVC explicitly separates the user input, the modeling of the external world, and the visual feedback into three separate layers (Burbeck, 1992). It is a sound foundation for any user interface, but becomes especially useful in thin client/mobile apps.

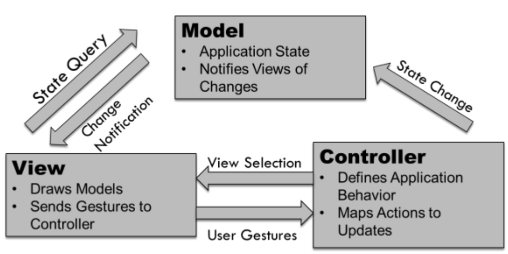


**Figure 7. Model - View - Controller diagram.**

In a thin client app, the user interface never directly alters the cached state of the model on the thin client. All of its control requests are sent to the business layer that resides on the networked server. The server executes the request, the model is changed, and its state is then broadcast out to the client. The client receives the model updates via the controller and the cached state is updated and reflected in its view. This makes the thin client stateless; it only exists to reflect the state of the server's contents.

As an example, we will follow how a target expose command is processed in TÜL/TRACR. The user, using the Android TÜL app, clicks on a button to request a target exposure. The TÜL controller receives the button click and sends a request for target exposure to TRACR over the network. TRACR receives and process the request within the business and data layers. The underlying business model of the target's state is updated and reflected in its next status broadcast message.

TÜL's model of the target receives the state update, changes its internal value to reflect the TRACR target's state, and notifies its observers that it has been updated. Both the visual target icon and the target control button are registered observers to the target model and are notified of the state change, which causes them to be redrawn to reflect their new state.

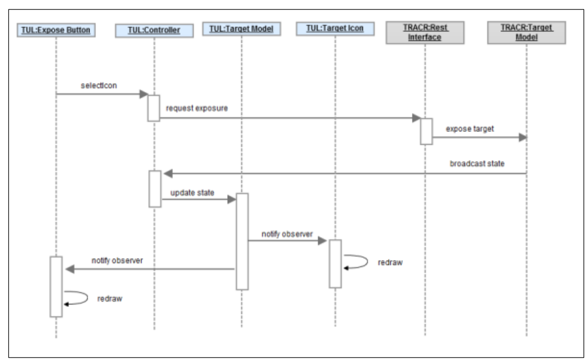


**Figure 8. Sequence diagram of communication between TRACR and TÜL.**

### TRACR Communication

TÜL interacts with TRACR over a standard Wi-Fi network, receives TRACR status via Multicast, and sends command requests using a Representational State Transfer (REST) interface. In short, REST is an architectural style for distributed systems that defines how clients and servers interact with each other. The largest implementation of a REST system is the World Wide Web.

On app startup, TÜL scans the network and connects to a TRACR instantiation, if available. The site configuration is read on TRACR from an SQL database and streamed over the Wi-Fi network to initialize TÜL's site model.

**Site Layout**

As discussed earlier, TRACR operates in two distinct sites: lane and maneuver. The interface for Lane based training is more of an abstract representation of the lane site compared to the view of the maneuver range. The lane interface (see Figure 9) shows a table of the training lanes at the site. A user can choose to view the entire site, a single lane, or filter the site view to show only the lanes they are interested in.


**Figure 9. TÜL running in lane mode. The center pane displays the lanes and targets in the site. The left pane allows the user to navigate to a lane of interest.**

The maneuver site interface (see Figure 10) utilizes a satellite map view similar to the typical Google Maps interface. Common map controls found on mobile devices are replicated including pinch to zoom and swipe panning. The targets are placed on the map at their real world locations, and are organized into objectives that are traversed by the trainees. Similar to the lane interface, a user can zoom into an objective using the objective pane on the left side.

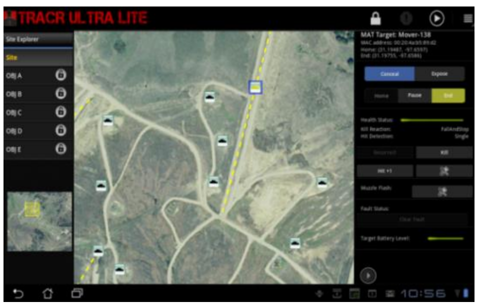


**Figure 10. TÜL in maneuver view. Targets are placed on the map at their real coordinates.**

**Target Controls**

The target visuals and target controls are consistent independent of the site mode. The target is represented in one of two states; either concealed (black) or exposed (red). A target can either be alive (able to be exposed) or dead (cannot be exposed), which is represented by a red 'X' overlay. The target visual also has overlays available to represent hit count, battery life, and fault state.

The control panel, in the right pane, is used to display the target's properties and to control the target. A user can select either a single target or multiple targets. The control panel is context sensitive, and has a different layout depending on whether a single or multiple targets are selected. The control panel is used to change the state of the target, including its exposure state, health, and response parameters.

Like the control panel, the target selection interface is uniform independent of site mode. The interface allows for single and multiple target selection. A single target can be selected by touching a target icon. When a target is selected, any previously selected target will be deselected and the new target will be the active target. The interface supports a "lasso" gesture for selection of multiple targets. To select multiple targets a user would touch and hold an empty area of the site until a visual response cue is displayed. From there the user can draw a path and "lasso" multiple targets, allowing them to command multiple targets with a single button press.

**Scenarios**

The individual selection and controlling of targets is applicable to running TRACR/TÜL in interactive mode, however most individual Warfighter training occurs in scenario playback mode. Scenarios are scripted behaviors which control the exposure and concealment of targets. A scenario executes within a timeline and each target event is triggered at a defined time. The scenario exists on the server, and is executed on the TRACR server. The TÜL app sends a REST request to the TRACR server to execute. Targets are manipulated by commands stored in the scenario and their status is communicated to TÜL via the typical scenario status message. TÜL has the capability to start, pause, and stop a scenario. Figure 11 illustrates scenario playback functionality. Controls are displayed on the bottom, and the right pane shows each scenario event, with a description.


**Figure 11. TÜL scenario view interface. The scenario playback controls are displayed on the bottom, and the right pane shows each scenario event, with a description.**

### Special Controls

Providing control to the trainer on the range will have a positive impact on range safety. The upper right corner of the TÜL app contains an emergency stop button. The emergency stop button has a safety trigger (an additional button has to be clicked to enable it) to prevent a user from accidentally shutting down the range. Once the button is active, it will send a signal to TRACR to completely shut down the site.

For a TÜL user to control a lane, they have to reserve a lane by "locking" it. Once a lane is locked, no other TÜL user can control the targets on a lane. The tower, however, still has the capability to override TÜL changes.

## FOUNDATIONS FOR GROWTH

At the time of writing this paper, TÜL is tentatively scheduled to be installed at a live-fire training site, such as Ft. Pickett or Ft. Eustis. Field testing will put TÜL directly in the hands of the trainers and will provide a tremendous amount of feedback for TÜL itself. It will also prove the applicability of mobile devices as training devices. Nothing can determine the effects of everyday use on a device, user adaptation to the devices, and general feasibility of the solution like putting it to use in the field in actual training exercises.

TÜL can also benefit from other available technologies on the market. The Tactical Terrain Analysis app, for iOS and Android, is a mobile app which provides tools for situational awareness to the user (Borkman, Hoffman, and Peele, 2010). Efforts are currently on-going to merge the Tactical Terrain Analysis app's capabilities into TÜL providing capabilities of line-of-sight analysis and GPS location. This will increase site safety by assuring that no individuals accidentally enter the line of another trainee's fire.

The mobile device, with its suite of sensors and capabilities, provides a rich foundation for future work. An augmented reality interface, virtual components overlaid on real images captured from the camera, can show the location of bullet hits and misses.

Independent of TÜL, the future of the mobile device market is bright. Recently Apple, a manufacturer of both smart devices and traditional computer platforms, announced that we are living in a "post-PC" world (Topolsky, 2011). International Data Corporation recently reported that 916 million smart devices were shipped in 2011, and the number is expected to reach 1.84 billion units by 2016 (International Data Corporation 2011). And right now the Army is making advances in the procurement of smart devices with plans of fielding such devices by mid-2012. Some apps have already been field tested, such as Nett Warrior (Bacon, 2011), to rave reviews from the Soldiers who have used them. While Angry Birds might be fun, there is so much more smart mobile devices can do for our Warfighter. It would be foolish not to leverage these devices, especially since they are capable of increasing the quality, efficiency, and safety of training.

## ACKNOWLEDGMENTS

The TRACR application was developed and is maintained by Riptide Software. Riptide has provided tremendous assistance to us on the TÜL development effort. Brian Kemper at U.S. Army Program Executive Office for Simulation Training, and Instrumentation and Julio de la Cruz at US Army Research Laboratory Simulation and Training Technology Center have also provided significant support to this project.

## REFERENCES

Activities: Fragments (n.d.) Retrieved June 10, 2012 from Android Developers reference site: *http://developer.android.com/guide/topics/fundamentals/fragments.html*

Android Design (n.d.). Retrieved June 10, 2012 from Android reference site: *http://developer.android.com/design/index.html*

Army CIO/G-6 (2012). Army launches apps marketplace prototype. Retrieved from http://www.army.mil/article/75966/Army_launches_apps_marketplace_prototype/

Bacon, L. (2011). Tablets, cellphones give Soldiers 'extra edge'. Retrieved from http://www.armytimes.com/news/2011/11/army-tablets-cellphones-give-soldiers-extra-edge-111511/

Baldwin, R (April 5, 2012). Google Glasses Face Serious Hurdles, Augmented-Reality Experts Say. Retrieved from http://www.wired.com/gadgetlab/2012/04/augmented-reality-experts-say-google-glasses-face-serious-hurdles

Borkman, S., Hoffman, M., & Peele, G. (2010). "Empowering Our Warfighter: Using iPhones for Situational Awareness" *I/ITSEC Proceedings*.

Burbeck, S. (1992) Applications Programming in Smalltalk-80: How to use Model-View-Controller (MVC). Retrieved from http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html

Cisco (2012). Cisco Visual Networking Index Forecast Projects 18-Fold Growth in Global Mobile Internet Data Traffic From 2011 to 2016 [Press Release]. Retrieved from http://newsroom.cisco.com/press-release-content?type=webcontent&articleId=668380

Gould, J. (2011). Army expects to field smartphones next year. Retrieved from http://www.armytimes.com/news/2011/12/army-expects-to-field-smartphones-next-year-122911/

Infinite Research (2012) Worldwide Tablet Computer Market Forecast. Retrieved from http://www.infiniteresearch.net/files/Infinite%20Research%20-%20Worldwide%20Tablet%20Market%20Forecast%20Brochure.pdf

International Data Corporation (IDC) (2012) Nearly 1 Billion Smart Connected Devices Shipped 2011 with Shipments Expected to Double by 2016 [Press Release]. Retrieved from http://www.idc.com/getdoc.jsp?containerId=prUS23398412

Meier, J.D., Homer, A., & Hill, D. (2008) Designing Mobile Applications. In *Microsoft Application Architecture Guide, Patterns & Practices, 2nd Edition* (Chapter 24). Retrieved from http://msdn.microsoft.com/en-us/library/ee658108.aspx

Nielsen (2011) Generation App: 62% of Mobile Users 25-34 own Smartphones [Press Release]. Retrieved from http://blog.nielsen.com/nielsenwire/online_mobile/generation-app-62-of-mobile-users-25-34-own-smartphones/.

Topolsky, J (March 3, 2011). Editorial: It's Apple's 'post-PC' world -- we're all just living in it. Retrieved from http://www.engadget.com/2011/03/03/editorial-its-apples-post-pc-world-were-all-just-living/