2009.1 SBIR Phase I Technical Proposal

“Smart-GUI” – An Automatic User Interface Configuration Management

TABLE OF CONTENTS

Black Sheep Networks

9 Prospect Hill

Tewksbury, MA 01876

Part 2 - Identification and Significance of the Problem

2.1 Identifying the Problem: Reducing the time to develop user interfaces and increasing the cross-system consistency with an automatic user interface configuration management system.

Tools to support and enforce quality and consistent graphical user interface (GUI) designs are lacking. Software engineers have testing and debugging tools. Similar tools are needed for designing user interfaces. These tools must be able to expand with constantly changing technology and employ the software concept of “configuration management” to user interfaces.

2.2 The Solution: Black Sheep Networks’ “Smart-GUI”

A smart “wrapper” to existing code will provide a common GUI. This “wrapper” will be easy to understand and work across all Operating Systems and platforms providing consistent user interfaces.

An automated and controlled approach will be created to maintaining user interface design consistency across all Operating System platforms regardless of the actual development language used. Elements of a user interface will be controlled, maintained and expandable. These elements will include color scheme, font, icon usage, icon design, menu/functionality placement, buttons, check boxes, etc. A testing capability will be provided to assure the design is appropriate to the intended purpose.

2.3 The Design of “Smart-GUI”

The stated problem is regarding ‘GUI’ design, and not actual coding or devices being used. Different code often produces different user interfaces… different colors and designs that only make things confusing to the operator or user.

Therefore, a “wrapper” providing a common GUI and GUI design is needed to be wrapped around actual code… whether the code is C, ADA, shell scripting, Visual Basic, etc… All these coding environments, and the Operating System they run on, will have the same GUI design.

Writing a new language or using custom/static libraries for ensuring a common GUI design is not the way to approach this problem as it will be difficult to update and keep current, especially with unique development environments where libraries are not compatible and challenging to update.

Separating the GUI development from the actual underlying or legacy code... yet wrapping the code within a GUI design is what is needed to ensure low impact to existing and future code regarding any GUI design changes. Each button, widget, window, or dial will be managed, and GUI standards enforced, with the “smart” GUI-wrapper design. Pressing a button will run existing compiled code already written in C, ADA, shell script, etc..

A GUI designer tool will also provide easy designing of standards-enforced GUI's. When system coding changes are required then this GUI-wrapper design will not affect actual code. Updates to operating systems will not affect GUI code and system code operability. This basis of this cross-platform consistency and GUI management will be accomplished through open-source Tcl/Tk language and windowing interface. The automated GUI generation will be accomplished through Visual TCL.

2.3.1 Smart-GUI : TCL/Tk

Tcl/Tk presents the a common look-and-feel, regardless of what platform it is run on. This look could be X-windows based, MS Windows style, or any other style desired.

2.3.2 Smart-GUI : Rapid Development Utilizing TCL/Tk

Implementing GUI designs, and integrating the Smart-GUI wrapper with existing code will be 5-10x faster with Tcl than with other languages. Once the GUI application wrapper is built in Tcl, it can also be evolved rapidly to meet changing needs. Existing code will not need to be recompiled or redesigned to support a GUI design change or to meet new user interface standards. The change will be with the Smart-GUI wrapper. Automated GUI development will be performed with a customized Visual TCL software.

2.3.3 Smart-GUI : Graphical User Interfaces with TCL/Tk

With its Tk toolkit, Tcl provides facilities for creating GUIs that are incredibly simple yet remarkably powerful. For example, the Tk canvas widget makes it easy to create displays with graphics, yet it also provides powerful facilities such as bindings and tags. The text widget provides sophisticated hypertext capabilities and more. No other toolkit has the same combination of simplicity and power. Tcl provides these same benefits on Linux, UNIX, Windows and Macintosh too. Tcl even runs on embedded platforms such as Windows Mobile (Windows CE), and Embedded Linux. Black Sheep Networks' "Smart-GUI" will ensure the GUI design is consistent across all platforms such as Embedded Systems, PC desktops, Aircraft Systems, Submarine Combat Displays, Medical Devices, PDA's, and more. Smart-GUI will provide this type of platform consistency that will not even require recompiling or recoding the user interface code.

2.3.4 Smart-GUI : Cross-Platform Applications

Black Sheep Networks' Smart-GUI with Tcl runs on Windows, Macintosh, and nearly every imaginable Unix platform. This makes it an outstanding tool for creating cross-platform applications. For example, the same Tcl script can run on Unix, Windows, Embedded Linux, Windows Mobile, and Macintosh and display a graphical user interface. Because it runs on all major platforms, Tcl provides an excellent management and integration tool for mixed environments, such as those with Windows desktops and Unix servers.

2.3.5 Smart-GUI : Extensible Applications with TCL/Tk

Smart-GUI will allow for creating powerful applications that can be scripted and extended by its users and modified in the field… yet retaining a consistent user interface and design that meets standards and requirements. The Tcl language is unmatched for this purpose. The Tcl interpreter was designed from the start to be embedded in a variety of applications. It is easy to incorporate Tcl into an application, and the Tcl interpreter melds naturally with the application, almost as if the Tcl language were designed exclusively for that particular application.

2.3.6 Smart-GUI : Flexible Integration with TCL/Tk

With Tcl it is easy to coordinate existing components and applications so that they work together effectively. For example, it is easy to use Tcl as a control language for special-purpose hardware and protocols, add a GUI or network interface to a legacy application, or integrate new Java applications with legacy code in C or C++. This makes Tcl a powerful tool in areas such as network management and factory automation.

2.3.7 Smart-GUI : Testing

Tcl is an ideal language to use for automated test to ensure the design is appropriate for the intended purpose. With Smart-GUI utilizing Tcl, a tester can easily invoke test functions, check the results, and report errors with the user interface design. Tcl's interpreted implementation allows tests to be created rapidly, and the tests can be saved as Tcl script files to reuse for regression testing. When testing a software application and how it will interface with the Smart-GUI wrapper, Tcl allows you to connect directly to lower-level APIs within the application, which provide much more precise and complete testing.

2.3.9 Smart-GUI : Consistency with Legacy Applications

No development platform has an easier to use GUI design than Tcl. Tcl provides a great way to add modern user interfaces to legacy applications.

2.4.0 Smart-GUI : Visual TCL – Automated GUI creation

Visual Tcl is a freely-available, high-quality application development environment for Linux, UNIX, Windows, Macintosh and AS400 platforms. Visual Tcl is written entirely in Tcl/Tk and generates pure Tcl/Tk code. As part of Smart-GUI, Visual Tcl will be customized to be integrated with user interface design standards which will allow for the creation and testing of a GUI that has a consistent look and feel that meets design guidelines.

Part 3 – Phase 1 Technical Objectives

<u>Objective 1</u>: Examine literature for both User Interface guidelines and software configuration management

> Contact the program manager and discuss the requirements, needs, and guidelines in full detail. How are the current user interface guidelines being enforced? What type of impact will configuration management have on the existing development of user interfaces? What will be needed to integrate the Smart-GUI "wrapper" design with the existing code base? Discover how previous user interface management and consistency plans may have failed and how to overcome these problems in a managed fashion.

<u>Objective 2</u>: Develop a plan for the Smart-GUI "wrapper" that will automate user interface configuration and generation

> Discover possible pitfalls with integrating existing code into a GUI interface wrapper. Will existing code have existing GUI interfaces? How will existing code need to be modified in order to interact with a user interface wrapper? How will new code be designed and managed with the Smart-GUI? What procedures are needed for testing a user interface's consistency and compliancy?

Accomplishment of all the technical objectives will allow Black Sheep Networks to proceed to Phase II, which includes the development of a working Smart-GUI, an automated user interface configuration management/generation system. Phase II will also include the assurance that Smart-GUI will generate consistent and usable interfaces for at least one military system.

Part 4 - Phase 1 Work Plan

The following tasks will occur in serial immediately upon SBIR award:

Task	Staff	Start	Finish	Task Details	Milestone
Research Guidelines	Kevin Caldwell Romande Carter	Week 1	Week 4	Research both User Interface Guidelines and software configuration management	Review of the problem and guidelines are complete
Smart-GUI Design	Kevin Caldwell Romande Carter	Week 5	Week 20	Full Detail and Design of Smart-GUI to automate user interface design and providing consistency on 'new' software code	Smart-GUI design is complete
GUI testing Design	Kevin Caldwell	Week 20	Week 24	Automated GUI testing software design to ensure user interface performs the intended purpose	Design for testing of user interface intended purpose is complete

Deliverables

a. Kickoff meeting within 30 days of contract start

b. Progress reports

c. Technical review within 6 months

Part 5 - Related Work

Black Sheep Networks has worked with SPAWAR, NAVSEA, and the IRS through larger defense companies such as Eagen McAllister and Raytheon. The successes of these agencies specific graphical user interfaces for security applications were only possible with Black Sheep Network's innovative technical abilities in integrating Information Security and Graphical Design utilizing TCL/Tk. Most recent work was designing a common, and consistent, user interface for a set of Information Security applications for NUWC/NAVSEA through sub contractor Raytheon.

Part 6 - Relation with Future R&D

The methodology developed by Black Sheep Networks for the Smart-GUI automated user interface generation and configuration management will be easily adaptable to any future modalities. The proposed methodology developed under Phase I can be modified to work in any type of user interface development environment.

Part 7 - Commercialization Strategy

Black Sheep Networks seeks to develop “Smart-GUI” which will be marketed to any industry requiring cross platform user interface design automation, generation, and consistency. For example, industries such as healthcare can benefit by providing an easy-to-use GUI generation system that will run on all platforms from PDA’s, PC’s, health equipment displays, and embedded user interfaces. By providing the same look and feel, this can alleviate confusion and time spent on relearning an interface.

Part 8 - Key personnel

Kevin Caldwell - Technical Investigator, Chief Consultant

Kevin Caldwell brings a vision of providing high quality Development and Design with a keen eye regarding Information Security for both corporations and the Government. He has over 12 years of start-up and corporate experience. Mr. Caldwell has held several roles with his career in Information Technology. His roles have ranged from a UNIX administrator, Manager, Director, and even a co-founder of a web-hosting company and an Information Security Consultancy. He has worked with several companies large and small from RSA Security, Nortel Networks, RiverDelta Networks, Motorola, Paper Exchange, Arris, Bandwidth Center, Armored Servers (http://www.armoredservers.com), and Black Sheep Networks (http://www.blacksheepnetworks.com). Government agencies include SPAWAR Space and Naval Warfare, NUWC, NAVSEA, and the Internal Revenue Service. Mr. Caldwell has also worked for over 6 years securing, armoring, and providing cross-domain Government and Military solutions. Mr. Caldwell brings expertise in TCL/Tk Graphical Design, UNIX security, network security, and Government Security as well as the business mindset needed in today’s development and security designs.

Technical Investigator Relevant and Recent Experience:

Nov 2001 – Present	**BLACK SHEEP NETWORKS INC.** Chief Consultant Tewksbury, Massachusetts http://www.blacksheepnetworks.com •Founder of data/information security consulting corporation •SECURITY Consulting includes:

Penetration Testing
Firewall and DMZ configuration
Vulnerability Analysis
Operating System Hardening
Cyber Forensics
System and Network Security Design
Disaster Recovery Planning
FISMA / Government DCID6/3 Protection Level accreditations

- UNIX Consulting includes:

  Graphical Interfaces with TCL/Tk
  Upgrades and new Installations with:
  Solaris, SGI IRIX, AIX, variations of Linux, and HPUX, TSOL
  Application installations such as:
  Rational ClearCase, Veritas VM (VXFS), Oracle,
  Apache, LDAP, NIS, DNS, Checkpoint FW-1
  CA eTrust Access Control, RSA ACE/Server (SecurID)

- Some recent clients that Kevin has performed services for: (LONG and SHORT term)

RAYTHEON [Portsmouth, RI] 2/2004 – Present

- Secret Clearance at Raytheon in Portsmouth, RI

[Tech Insertion TI08] 9/2007 – Present

- Designed next generation NTDS submarine interface gateguard with Solaris 10 and Trusted Extensions utilizing Containers, aka Zones, labeling and Mandatory Access Controls (MAC) to meet DCID6/3 PL4 accreditation.
- RedHat Linux RHAS 5 minimization and armoring

[Tech Insertion TI06] 9/2006 – 8/2007

- Designed system based on NTDS submarine interfaces with Trusted Solaris 8 to meet DCID6/3 PL4 and JDISCIS by ONI/SSO NAVY (Office of Naval Intelligence) and DITSCAP accreditation (STIG)
- Designed PKI authentication using keys instead of passwords for entire Weapon Control submarine network
- RedHat Linux RHAS 4 minimization down to 100mb filesystem
- Computer Associate’s eTrust UNIX Access Control implementation
- SUN SunScreen firewall on Trusted Solaris 8
- Implemented iSCSI SAN environment, along with GFS filesystem and RedHat Cluster Servers
- Designed unique Trusted Solaris 8 system to be defined as Controlled Interface, aka appliance, with no interaction or logins possible.
- Designed centralized shipboard virus scan gui in TCL/tk

- Designed development lab maintenance, cleanup, and usage admin GUI in TCL/Tk
- Assisted integrating TSOL functionality involving label downgrades into existing Weapons Control code at Raytheon

[Tech Insertion TI04] 2/2004 – 8/2006

- Designed a Centralized Password Accountability and Distribution software application developed in TCL/Tk for cross-platform independence. Implemented on Submarine Combat Systems, providing logging and auditing of user, group, and password schemes using encrypted network tunnels and an embedded SQL Database Engine.
- Provided Information Assurance for Submarine Combat Systems
- Hardening of HPUX 10.20 and 11.0, Trusted Solaris 8, RedHat Linux 7.3 and 9.0, and Windows 2000. Design and creation of OS Armoring scripts.
- W2K Armoring application designed with TCL/Tk to provide a GUI interface. Use of Freewrap to compile TCL/Tk code into an self-contained executable format.
- Usage of DoD STIG Security Guidelines and scripts
- Setup of Symantec Enterprise Security Manager (ESM)
- Setup and support of Symantec Intruder Alert (ITA)
- Design of Central Virus Scan for Linux/HPUX platforms
- Full Customization of Symantec ESM for security settings
- Deployed and Administered Trusted Solaris 8 environment
- Provided Trusted Solaris 8 (x86) Support

RAYTHEON ***Woburn, MA Location*** 10/2006 – 12/2006

- Jfires (Joint Fires) platform, Designed and implemented Trusted Solaris 8 on several SUN Fire v440 systems to meet special environment restrictions and requirements for NISPOM PL3 accredidtation by DIA.
- Full documentation for Design and meeting PL3 accreditation

Sunbelt Network Inc. 8/2004 – 11/2007

- Provided Oracle 10g installation, configuration and support on SUN E4500, and SUN Fire v880 servers.
- EMC Symetrics Storage devices configure and maintain

EMB Statistical Solutions, LLC 1/2004 – 3/2004

- Contract research organization providing data Management and statistical analysis of

Clinical research data
- Design and Setup of data storage system
- SUN Hardware and Solaris 10 OS Design

<u>SPAWAR (Space and Naval Warfare)</u> [Charleston, SC] 3/2003 – 2/04

- Secret Clearance at SPAWAR Systems Center [DoD]
- Lead INFOSEC UNIX engineer with Tiger Team
  Working with SPAWAR Information Warfare agency [DoD]
- Intrusion Detection and Policy Enforcement
- SPAWAR and IRS (Internal Revenue Service) Security assessments
- Designed Software Development Process for IRS UNIX LEM
- Setup/Maintained Configuration Management with ClearCase
- Migrate development code to ClearCase
- Established ClearCase vobs for four development projects
- Established backup procedures
- Established procedures for check-in/out, setview, etc.
- Established build and integration procedures
- Provided training as needed
- Setup development environment consisting of AIX, HPUX, BSDi, Solaris, Linux, and Tru64 operating systems
- Created Software Development portal for UNIX LEM Checker
- Managed development of UNIX System Security Scanner for IRS
- Design and Implement NTP securely with
  Access control and Authentication
- NTP and Sendmail Security Documentation
- ISS (Internet Security Scanner) Scans
- eTrust Access Control and Audit Deployment
- CVS and Configuration Management Deployment
- Setup Coding, BugTracking, and Testing Environment
- Documentation Writing and Management
  Standard Operating Procedures
  Technical Documentation

Romande Carter – UNIX Technical Engineer

Romande Carter has more than nine years of experience with UNIX systems administration and TCL/Tk programming . He has worked with companies large and small such as Nortel/Arris, Fleet Bank, and Evergreen Investments. His UNIX Administration and development background with financial companies has proven to be a tremendous value in regards with graphical user interface designs.

Part 9 - Facilities/Equipment

All proposed work will be completed at the Black Sheep Networks' office in Tewksbury, MA. Black Sheep Networks maintains a computer development datacenter and has the facilities and equipment to research and develop innovative solutions and technologies.

Part 10 - Subcontractor and Consultant Involvement

Black Sheep Networks will not need any subcontractor or consultant involvement on this project for Phase I.

Part 11 - Prior, Current or Pending Support of Similar Proposal or Award

Black Sheep Networks has no prior, current, or pending support for a similar proposal.