ST. MARY'S UNIVERSITY OF SAN ANTONIO

# An Expert System for Network Router Configuration

## Software Requirements Specification

**Michael A. Perez**

**Spring 2015**

# 1. INTRODUCTION

## PURPOSE

This document presents formal requirements required for project development. Specifications for these requirements are documented here with enough detail to define system behavior. All requirements defined here can be traced back to targeted user needs and goals in the Vision document. This document overlaps greatly with both the Analysis and the Design phases and as such may contain UML diagrams from both. UML diagrams will be marked with their source phase.

## OVERVIEW

This document classifies requirements as one of functional, non-functional, or other. Functional requirements describe the behavior of the system and the software architecture. Non-functional requirements describe the characteristics of the system, such as design constraints, and describe the system's technical architecture. Classifying requirements as "Other" is a catch-all method to describe requirements that are neither functional nor non-functional. Document requirements fall under the "Other" classification.

## WRSPM REFERENCE MODEL

A reference model for using formal methods in developing requirements was described by [4]. Formal methods are mathematical techniques, Church's higher-order logic in this case, used by software engineers to solve problems in system verification and specification. This reference model defines the relationship between a system and its environment by the interaction of five artifacts: World, Requirements, Program, Machine, and Specifications. Simply, this WRSPM reference model states that requirements must be logically satisfied by the phenomena of the World artifact in which the system exists and the Specifications artifact which define the interface between the environment and the system. The World artifact defines all phenomena of the environment while the Machine artifact defines all phenomena of the system. The Requirements and Program artifacts specify the phenomena which are desirable of the environment and system, respectively. Using Church's higher-order logic, [4], state that W ^ S => R, or that the World and Specification artifacts logically satisfy the Requirement artifacts. Note that in the figure below the (S)pecification artifact is the interface between the Environment and the System.

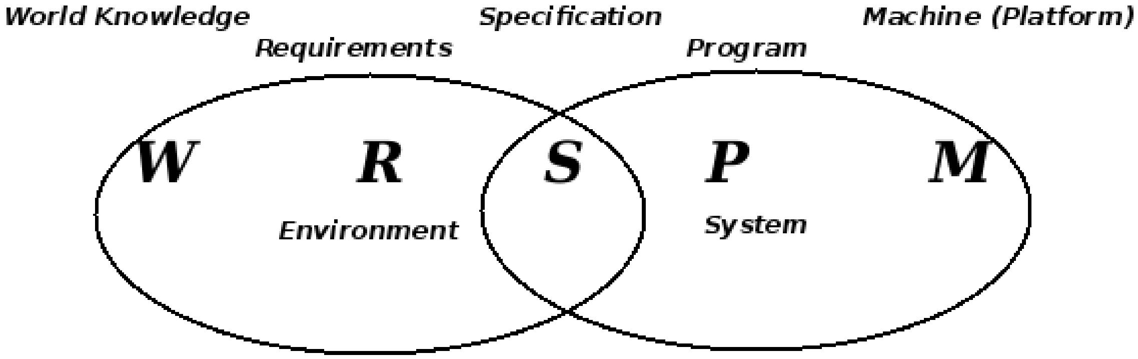

FIGURE 1: WRSPM REFERENCE MODEL

The phenomena of the World and Machine artifacts are known, non-functional requirements so we will discuss them in *section 2 Non-Functional Requirements*. Functional requirements are a product of the Requirement artifact, our the Software Product Features of the Vision document, and will be more formally discussed here in *section 3 Functional Requirements*. Phenomena of the the Specification artifact give our functional requirements verification and validity, so they, too, will be included in *section 3 Functional Requirements*. Phenomena of the Specification artifact will be derived from W, M, and R of our WRSPM model and from UML analysis diagrams as needed. The Program artifact is the software to be developed, an unknown at this point, and, as such, is not discussed in this document.

## 2. USE-CASE REALIZATIONS

To uncover any hidden or missed requirements from what we have already discussed in the Vision document, we will use UML analysis diagrams to realize our Use-cases.

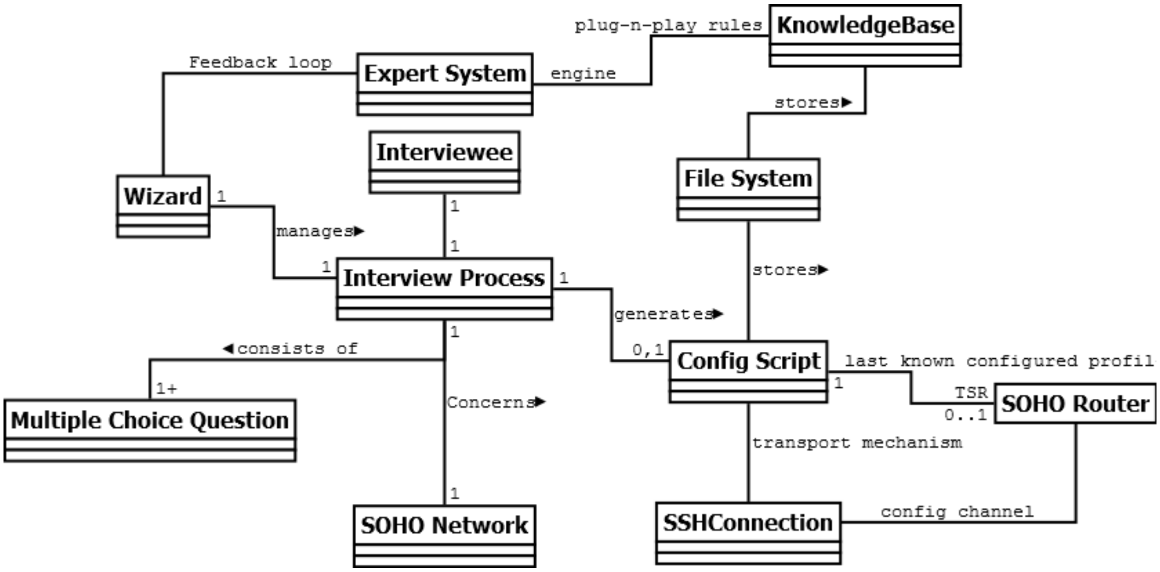

FIGURE 2: UML ANALYSIS CLASS DIAGRAM

## 3. NON-FUNCTIONAL REQUIREMENTS

This section defines non-functional requirements which specify overall system characteristics such as cost and reliability. Non-functional requirements drive the technical architecture of a system.

### WORLD (OR DOMAIN KNOWLEDGE)

(**W** - *R* - *S* - *P* - *M*)

"W restricts the actions that the environment can perform"

W1. The system user is not knowledgeable in network configurations.

W2. The system user can operate a computer Graphical User Interface (GUI)

W3. The system user will consider any response time over 7 seconds to be an indication of unresponsiveness.

W4. The system user has the SSH credentials of the targeted SOHO router (TSR)

W5. One TSR exists for each system user.

W6. The TSR is visible to the system user while operating the system.

**TABLE 1: WORLD ARTIFACTS OF THE WRSPM REFERENCE MODEL**

## MACHINE (PROGRAMMING PLATFORM)

(*W - R - S - P -* **M**)

"M restricts the actions that the system can perform"

M1. All pertinent networks use the TCP/IPv4 network stack.

M2. The operating system may be Microsoft Windows, Mac OS X, or Linux.

M3. A single targeted SOHO router (TSR) provides wireless and wired network connectivity between an ISP gateway router and a residential ISP customer's network.

M4. All pertinent networked devices use the TSR as their network gateway

M5. The TSR is running OpenWRT version Backfire 10.03.1 firmware

M6. The firmware is running an SSH server using SSH protocol version 2

M7. The firmware supports network configuration through the command line utility known as the Unified Configuration Interface (UCI)

M8. It uses the CLIPS inference engine developed by NASA

M9. The CLIPS inference engine accepts a flat file as a knowledge-base.

M10. The knowledge-base is TSR firmware specific

M11. The CLIPS inference engine is TSR firmware agnostic

M12. Configuration script data will be stored in flat files.

M13. Public/private key pairs for SSH login are stored using the RSA key format as a pair of files on the local file system.

M14. The system runs an SSH client using SSH protocol version 2

M15. The system and the TSR communicate using SSH tunnels only

**TABLE 2: MACHINE ARTIFACTS OF THE WRSPM REFERENCE MODEL**

# 4. Functional Requirements

This section defines the capabilities and functions that the System must be able to perform successfully. Functional requirements drive the application architecture of a system. Our functional requirements come from the Software Product Features from the vision document.

| Requirements<br>(*W* - ***R*** - *S* - *P* - *M*)<br>"R says which of all possible actions in W are desired" | |
|---|---|
| Traceability | Requirement |
| SPF01 | R1. The system shall prompt user with Yes/No question |
| SPF02 | R2. The system shall prompt user with Multiple Choice question |
| SPF03 | R3. The system shall start the wizard when the user says so |
| SPF04 | R4. The system shall cancel the wizard when the user says so |
| SPF05 | R5. The system shall determine if enough information from user has been collected |
| SPF06 | R6. The system shall package configuration into a repeatable script |
| SPF07 | R7. The system shall provide a configuration script viewer |
| SPF08 | R8. The system shall load a configuration file from disk |
| SPF09 | R9. The system shall provide a configuration script editor |
| SPF10 | R10. The system shall save configuration commands to a file that can later be executable on the target SOHO router (TSR) |
| SPF11 | R11. The system shall remotely apply a configuration script via SSH to the target SOHO router (TSR) that resides in the local LAN segment. |
| SPF12 | R12. The system shall determine the gateway IP address of the local LAN segment |
| SPF13 | R13. The system shall determine the IP address of the target SOHO router (TSR) |
| SPF14 | R14. The system shall login to the TSR via SSH with username/password credentials |

| | |
|---|---|
| SPF15 | R15. The system shall login to the TSR via SSH with public/private key pair credentials |
| SPF16 | R16. The system shall generate public/private key pairs for use in SSH logins |
| SPF17 | R17. The system shall load a knowledge-base to use for the wizard |
| SPF18 | R18. The system shall load a default knowledge-base when a user-defined knowledge-base is not provided |

**Table 3: Requirement artifacts of the WRSPM reference model**

We now define Specification artifacts and, because of Adequacy as defined previously ( W ^ S => R), we will use these specifications as our method of requirement verification and validity.

| | |
|---|---|
| SPECIFICATIONS<br>(*W* - *R* - **S** - *P* - *M*)<br>"S properly takes W into account in saying what is needed to obtain R" | |
| **Traceability** | **Specification to verify and validate the referenced requirements** |
| R1, R2, R5 | S1. The inference engine uses initial facts from the knowledge-base and user responses gathered up to this point to infer the next set of actions to take.<br>S2. When the user answers a question, the inference engine matches the user's response and existing assertions with the antecedents of unactivated rules.<br>S3. The knowledge-base stores each user question as part of the consequent of a rule<br>S4. The knowledge-base stores UCI configuration commands as part of the consequent of a rule<br>S5. When an activated rule's consequent includes a UCI command, the command is appended to the configuration script.<br>S6. When an activated rule's consequent includes a user question, the user is prompted with the question.<br>S7. When the list of activated rules' consequents do not include a user question, the wizard should complete the resulting configuration script should be saved as a flat file to disk. |

| | |
|---|---|
| R6, R7, R8, R9, R10 | S8. The configuration script is a flat file consisting of a collection of UCI commands, and other commands as needed, that are executable on the target SOHO router (TSR) running OpenWRT.<br>S9. The system saves the script file on the local filesystem after running the wizard through to completion<br>S10. The system loads the script file into a view for the user to review and edit after the user loads a previously saved configuration script.<br>S11. The script file view allows simple text editing of the configuration script file.<br>S12. The script file view allows saving the configuration script to a flat, text file using a filename of the user's choice. |
| R11-16 | S13. The system shall run an SSH client to connect to the SSH server running on the target SOHO router (TSR)<br>S14. The system will prompt the user for the SSH username and password of the target SOHO router (TSR) each time it connects and will not store the credentials in persistent data storage on the system.<br>S15. The system is able to generate its own private/public key pairs for use with SSH logins and distribute the public key to the target SOHO router (TSR)<br>S16. The system will treat default network gateway IP address as the target SOHO router (TSR) |
| R3-4 | S17. When the wizard is not running, a button should be marked in the system's GUI such that when clicked the wizard starts.<br>S18. When the wizard is actively running, a button should be marked in the system's GUI such that when clicked the wizard is interrupted and stops running. |
| R17-18 | S19. When the wizard is not running, the user is able to select a knowledge-base flat file from local disk for the wizard to use the next time it runs.<br>S20. If the user starts the wizard without choosing a knowledge-base file, a default knowledge-base file should be available to use. |

**Table 4: Specification artifacts of the WRSPM reference model**

# 5. Other Requirements

## Documentation Requirements

### User Manual

A key feature of this system is the accessibility to novice users. Therefore, a user manual is an important part of the project deliverables. The Normal and Alternative flows from the Use Cases will guide the creation of the user manual.

## Installation Guide

A guide will be written in which the installation will outlined.

## Labeling and Packaging

The system is developed and packaged as a single unit. Any ancillary systems (i.e. database, server software, etc.) are not provided.

## Licensing Installation

N/A