Subsystems Report

Version <2.0.0>

**Document Control**

**Approval**

The Guidance Team and the customer shall approve this document.

**Document Change Control**

|  |  |
| --- | --- |
| Initial Release: | 3/2/2017 |
| Current Release: | 3/8/2017 |
| Indicator of Last Page in Document: | 11 |
| Date of Last Review: | 3/8/2017 |
| Date of Next Review: | 3/9/2017 |
| Target Date for Next Update: | 3/9/2017 |

**Distribution List**

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members:

Dr. Salamah

Customer:

Dr Jaime Acosta

Software Team Members:

Alfredo Garcia

Raul Hinostroza

Thomas George

Hector Cervantes

Eulalio Garcia

**Change Summary**

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0.0 | 3-2-17 | Alfredo Garcia | Uploaded template |
| 1.1.0 | 3/4/17 | Alfredo Garcia | Added 3 contracts to document: PCAP, PCAP Manager, Exterior Parser Controller |
| 1.1.6 | 3/6/17 | Alfredo Garcia | Added 2 contracts to document: Dissected Packet Manager, Project list |

Table of Contents

[Document Control ii](#_Toc445824963)

[Approval ii](#_Toc445824964)

[Document Change Control ii](#_Toc445824965)

[Distribution List ii](#_Toc445824966)

[Change Summary ii](#_Toc445824967)

[1. Introduction 1](#_Toc445824968)

[1.1. References 1](#_Toc445824969)

[2. Contracts 2](#_Toc445824970)

[3. Subsystem Cards 2](#_Toc445824971)

[4. Diagrams 2](#_Toc445824972)

[4.1. Collaboration graphs 2](#_Toc445824973)

# Introduction

* 1. << This section gives introductory information regarding the project and the subsystems of the software. Identify the project and define its scope. Describe the purpose of the subsystems report. Include definitions, acronyms and abbreviations. Indicate what the document contains and who the intended audience is. Describe the sections of the document>>

## References

<<References used to create this document>>

# Contracts

Ones in red still under reviewed by team 7 before putting cards on document.

|  |  |
| --- | --- |
| **Class Name:** GUI | |
| **Description:** | |
| **Contracts:**   1. View Controller  * Support Operations for Views   **Private Responsibilities:** | **Collaborations:**  Project(1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Project | |
| **Description:** This class represents the back-end logics supporting each tab in the project list. | |
| **Contracts:**   1. Manage Project  * Register itself with project list * Create a Project * Import Project * Close Project  1. Export  * Export Project  1. Save State  * Knows Project Current State   **Private Responsibilities:** | **Collaborations:**  Project List(1)  Project List(1)  Dissector Script Generator(1­)  Project State(1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Project State | |
| **Description:** | |
| **Contracts:**   * Knows Project name * Knows Project description * Knows Project path * Knows Protocol Decision Tree * Knows Layout Preference   **Private Responsibilities:** | **Collaborations:**  Layout Preference(1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Workspace | |
| **Description:** Class storing knowledge of systems relation to file structure, and available projects. | |
| **Contracts:**   1. Knows Name  * Knows Path  1. Knows associated Project list  * Associates with a project list.   **Private Responsibilities:**  Create workspace | ProjectList(2) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** PDGS Launcher | |
| **Description:** Bootstrap loader for the system. | |
| **Contracts:**   1. Loads Workspace  * Switch active Workspace * Launches Workspace   **Private Responsibilities:** | **Collaborations:**  Workspace (1)  Workspace (1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Project List | |
| **Description:** container for open projects | |
| **Contracts:**   1. Manage Project List  * Knows list of available Project * Remove Project  1. Initialize project space  * Register itself with workspace   **Private Responsibilities:** | **Collaborations:** |
| **Comments:** 1st contract used by project class | |

|  |  |
| --- | --- |
| **Class Name:** Temporary Format | |
| **Description**: This class handles XML version of a project | |
| **Contracts:**   1. Translates Project into XML:  * Stores Project into XML format * Knows protocol information   **Private Responsibilities:** | **Collaborations:**  Project (2) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Layout Preference | |
| **Description:** Class to handle preferences of layout for a user. | |
| **Contracts:**   1. Preference Settings  * Organize Views * Keep track of hidden views * Keep track of Builder Area * Knows view * Knows X coordinate * Knows Y Coordinate * Knows Width * Knows Height   **Private Responsibilities:** | **Collaborations:** |

|  |  |
| --- | --- |
| **Class Name:** Dissector Script Generator | |
| **Description:** Class responsible for generating LUA format scripts | |
| **Contracts:**   1. Translate to Lua  * Translates temporary format to LUA script   **Private Responsibilities:**  Knows Project Name  Knows Dissector Format  Knows Save Location | **Collaborations:**  Temporary Format(1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Error Manager | |
| **Description:** Manages all error messages | |
| **Contracts:**   * **Knows state of required field** * **Knows number of start block** * **Knows number of end block** * **Knows number of connectors**   **Private Responsibilities:** | **Collaborations:**  Protocol Decision Tree(3)  Protocol Decision Tree(2)  Field Block(2)  Decision Construct |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Reference List | |
| **Description:** Class representing a Reference Table, made of lists. | |
| **Contracts:**   1. List values  * Knows value   **Private Responsibilities:**  Knows name  Knows text description | **Collaborations:**  List(1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name: List** | |
| **Description: Structured data storage with a value and a description.** | |
| **Contracts:**   1. List Information  * Knows Value * Knows Description   **Private Responsibilities:** | **Collaborations:** |
| **Comments:** | |
| **Class Name:** Packet | |
| **Description:** A class that is a message segment that has been captured over a network and stored into a PCAP file. | |
| **Contracts:**   1. Knows packet metadata:  * Knows Value * Knows packet description * Knows name * Knows packet information  1. Register packet to packet container  * Register itself to packet list in PCAP manager   **Private Responsibilities:** | **Collaborations:**  PCAP Manager(2) |
| **Comments:**  This class holds the dissected packet information separated by the PCAP Manager, for a single packet. This information will be crucial for the dissected packet manager. The packet metadata is separated by the different information, within a packet of traffic data captured, that was captured in a PCAP file. The packet registers its existence to the PCAP manager so the PCAP manager can act as a container for the packet. | |
|  | |

|  |  |
| --- | --- |
| **Class Name:** PCAP | |
| **Description:** A save format for traffic gathered by T-shark. | |
| **Contracts:**   1. Knows PCAP information  * Knows PCAP Path * Knows PCAP Raw Data   **Private Responsibilities:** | **Collaborations:** |
| **Comments:**  PCAP class is a data store that stores PCAPs that are imported from the workspace. | |

|  |  |
| --- | --- |
| **Class Name:** Exterior Parser Controller | |
| **Description:**  This class get together a script and a PCAP and send it to T-shark for dissection. | |
| **Contracts:**   1. Retrieves T-shark results  * Know resultant PCAP   **Private Responsibilities:**   1. Send Luascript to T-shark 2. Send PCAP to T-shark | **Collaborations:**  PCAP (1) |
| **Comments:** This class sends a PCAP or packet and a Lua Script to Tshark and retrieves the resultant dissected PCAP file from Tshark. | |

|  |  |
| --- | --- |
| **Class Name:** Dissected Packet Manager | |
| **Description:** The class will show the accuracy of the dissector by comparing a dissected packet with the protocol the user has created. | |
| **Contracts:**   1. Manages dissected packet data  * Know dissected packet info * Knows dissected packet accuracy   **Private Responsibilities:**  Calculates comparison results | **Collaborations:**  Packet(1)  Packet(1), Temporary Format(1) |
| **Comments:** Takes in a single packet, parses out the information, and compares it to what is expected in a temporary format. Results are used to determine the accuracy of the packet to the dissector that has been created. | |

|  |  |
| --- | --- |
| **Class Name:** Exterior Parser Controller | |
| **Description:** Send Lua Script: PCAP to T shark and retrieves results | |
| **Contracts:**   1. Retrieves results    * Knows resultant PCAP   **Private Responsibilities:**   * Sends Lua Script * Sends PCAP to T Shark | **Collaborations:**  PCAP (1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** PCAP Manager | |
| **Description:** container for packets that have been separated | |
| **Contracts:**   1. Split PCAP    * Separates PCAP into packets 2. Knows Packet list    * Knows extracted packets    * Knows potential packet fields    * Keeps list of Packets   **Private Responsibilities:** | **Collaborations:**  PCAP (1),  Exterior Parser Controller (1)  PCAP (1),  Exterior Parser Controller (1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Protocol Decision Tree | |
| **Description:** Structure representing the architecture of a protocol | |
| **Contracts:**   * Knows protocol name * Knows protocol description * Knows blocks * Register itself with Project State * Knows protocol dependencies * Know start block * Describes required packet info   **Private Responsibilities:**   * Select a block for deletion | **Collaborations:**  Project State(1)  Protocol Decision Tree Construct(3) |
| **Comments:** A data structure, Contract 1: Associates itself with Project State. Contract 2: Information given by a start block | |

|  |  |
| --- | --- |
| **Class Name:** Protocol Dependency | |
| **Description:** Contains Dependency information for other OSI layers | |
| **Contracts:**   * Knows protocol name * Knows pattern * Register itself with Start Block   **Private Responsibilities:** | **Collaborations:**  Start Block(1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Protocol Decision Tree Construct | |
| **Description:** Super class to Decision Construct, Expression Construct, Block, and Connector Construct | |
| **Contracts:**   * Describes required packet info * Register on protocol decision tree   **Private Responsibilities:**   * Can delete itself | **Collaborations:**  Field Block(2)  Protocol Decision Tree() |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name: Decision Block** | |
| **Description:** Super Class is Protocol Decision Tree Construct | |
| **Contracts:**   1. Knows control flow strucure   Knows expression  Knows connector constructs  Can select connectors to delete  **Private Responsibilities:** | **Collaborations:**  Expression Construct(1)  Connector Construct(1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Expression Block | |
| **Description:**  Super Class: Protocol Decision Tree Construct | |
| **Contracts:**   * **Calculate expression** * **Knows operands** * **Knows operators**   **Private Responsibilities:** | **Collaborations:** |
| **Comments: Used by connector Block** | |

|  |  |
| --- | --- |
| **Class Name:** Connector Block | |
| **Description:** Superclasss is Protocol Decision Tree Construct | |
| **Contracts:**   1. Knows resultant of expression tree   Knows Result of expression construct  **Private Responsibilities:** | **Collaborations:**  Expression Construct(1) |
| **Comments:** | |

|  |  |
| --- | --- |
| **Class Name:** Start Block | |
| **Description:** Structure representing the architecture of a protocol | |
| **Contracts:**   * **Registers with Protocol Decision Tree** * **Knows Protocol Dependency**   **Private Responsibilities:** | **Collaborations:**  Protocol Decision Tree(2) |
| **Comments:** A data structure, Contract 1: Associates itself with Project State, Rename to blocks, rename al constructs to \_\_ block, create start block and end block subclasses of block | |

|  |  |
| --- | --- |
| **Class Name:** Field Block | |
| **Description:** Super Class: Protocol Decision Tree Construct | |
| **Contracts:**   * Knows abbreviation * Knows description * Knows reference list * Knows Data type * Knows required * Knows base * Knows mask * Knows value constraint * Knows Size * Knows it end block   **Private Responsibilities:** | **Collaborations:**  Reference List (1) |
| **Comments:** | |

# Subsystem Cards

|  |
| --- |
| **Subsystem Name:** |
| **Classes:** |
| **Collabortaion Graph:** |
| **Description:** |
| **Contracts:** |

# Diagrams

## Collaboration graphs

