Team 6: Mellow Yellow
CSPN-CSPL GUI

Design Notebook

CS 422
11/12/99

Nicholas Gunder
Kerry Hammil
Azrina Hussin
Ryoji Noda
Brock Rogers
Alex Velkov

Version 4.0

Abstract

This document contains the Design Notebook for the CSPN-CSPL GUI project. It contains:

1. A requirements analysis
2. 1 level 1 data flow diagram (DFD), 3 level 2 DFDs and 1 level 3 DFD
3. A context diagram
4. 4 Pspecs
5. A transaction analysis containing 6 transactions
6. A structure chart
7. A data dictionary

This document also describes our major design decisions. The most significant of these are our decision to treat the text editor as a third party application, so that it runs in a separate window from the GUI, and our decision to support Ghostview for PostScript viewing.

The CSPN-CSPL GUI provides an interface to the port team's CSPN executable (the port). Users of the GUI can specify command options to the port, invoke the port and receive feedback from the port that will be displayed in the GUI. The GUI will also have the ability to invoke the CGE executable as a third party application.

The Mellow Yellow team is responsible for the GUI project. The members of Mellow Yellow and their responsibilities are described in Appendix B along with the project schedule.

This document is organized into the following sections:

1. Introduction: Contains the statement of purpose for this document as well as a description of our approaches to traceability and design.
2. Requirements Analysis: Summarizes at a high level the requirements of the CSPN-CSPL GUI
3. Design Representation: Contains the DFDs, Pspecs, transaction analysis and structure chart
4. References: Contains the formal statement of 237 requirements for this project
5. Glossary: Cites documents that we used in the preparation of the design notebook
6. Appendices: Four appendices cover our data dictionary, RTM, schedule, and test cases

Table of Contents

Abstract …… ii
Table of Contents …… iii
List of Figures …… iii
List of Tables …… iii
1. Introduction …… 4
2. Requirements Analysis …… 6
3. Design Representation …… 8
4. References …… 19
5. Glossary …… 20
Appendix A: Data Dictionary …… 21
Appendix B: Project Schedule …… 28
Appendix C: Requirements Traceability …… 29
Appendix D: Identified Test Cases …… 34

List of Figures

Figure 1: Context Diagram …… 8
Figure 2: Level 1 DFD …… 9
Figure 3: Level 2 DFD …… 11
Figure 4: Level 2 DFD …… 11
Figure 5: Level 2 DFD …… 12
Figure 6: Level 3 DFD …… 13
Figure 7: Partitions of the System DFD …… 14
Figure 8: Structure Chart …… 17

List of Tables

Table 1: Design Decision Log Table …… 18
Table 2: Project Schedule …… 28
Table 3: Requirements Traceability Matrix …… 29

1. Introduction

This section describes the general goals of this document as well as our design and traceability approaches, and the organization of this document.

1.1 Project Purpose and Goals

This project summarizes the software requirements of the CSPN GUI project. It is a vital step in the process of designing the CSPN GUI. It addresses both the functional requirements of the product as well as the non-functional constraints. It also addresses miscellaneous conditions placed upon the software, such as the coding standard that our developers will follow and the development schedule we hope to meet.

The overall goal in the design is to contribute an effective, working and reliable interface to the CSPN project.

1.2 Design Approach

The design and engineering of the GUI is being done in two main parts and shall remain separated. The Functional GUI is being designed using a top-down SSA/SD process and will support all the functional windows API elements. The other element is the text editor and image display unit. These designs will include an Object Oriented SSA/SD process and will be designed with the assistance of the Microsoft Foundation Classes.

1.3 Traceability Approach

Our approach will trace our requirements using numbers assigned to them in our SRS. These numbers will identify each requirement in the Requirement Traceability Matrix. Requirements listed in other documents will also be assigned an ID number. The section that contains the requirement will be assigned the ID number. We will verify the requirements using Analogy, Analysis, Inspection, Demonstration and Test.

1.4 Background

The CSPN system model includes a GUI section that will interface the PORT and CGE elements into one central element that the user can access. It is determined that the GUI will hide many of the context elements that were included in earlier versions. It attempts to do so with a graphical approach that will be

easy and quick to use. All the features and options will be included along with two new elements: a text editor, and image viewer. The three elements will work together as separate executable files and will be extensible enough to allow the user to use other third party elements.

1.5 Organization of this Document

The design notebook is organized into the following sections:

- Introduction: A description of the goals of this document, project purpose, design, traceability approach and background.
- Requirement Analysis: A summarization and analysis of the major requirements specified in the requirement specification document.
- Design Representation: This is a design specification that includes: a context diagram, data flow diagrams, transaction analysis, transform analysis, design log, and references.
- Appendices: Miscellaneous topics, such as a data description of all the data that will be used in the project.

2. Requirements Analysis

This section briefly describes the requirements of the CSPN-CSPL GUI project.

2.1 Menus

The GUI shall contain menus to facilitate access to program options and functions. The menus shall be categorized according to the type of options their items perform. There shall be menus to provide options to handle file operators, editing operations, access to tools, window management operations, and help functions.

2.4 The Toolbar

The GUI shall contain a toolbar that provides one-click access to the most commonly used functions.

2.5 Window Information

The GUI shall use the MDI window format and shall provide standard Windows functions such as the ability to minimize, maximize, and close the window.

2.6 Help

The GUI shall provide context-sensitive help so that the user can press a key and see the appropriate help topic for the task he or she is currently performing. The help file shall be browsable and searchable. The GUI shall also provide tool tips for buttons and toolbar items.

2.7 Status Bar

The GUI shall display the current status in a status bar.

2.8 Dialog Boxes

There shall be display boxes that allow the user to graphically select files to open in the GUI, and to select filenames when saving files. There shall be a dialog box to allow the user to search and replace text in the text editor. There shall be a dialog box to allow the user to configure printing options. There shall be a dialog box that displays the current version information. There shall be a dialog box to configure system options. There shall be a dialog box to configure CSPN port options.

2.14 Text Editor

The GUI shall provide an ASCII text editor with context highlighting and line number display.

2.15 Image Viewer

The GUI shall provide an image viewer that can view BMP, GIF, and JPG files and that has the capability to save any image in any of these formats.

3. Design Representation

The context and DFD diagrams model all of the system elements. The communication between parts of the system is handled in the context diagram, while the DFD's show our functional breakup.

3.1 Context Diagram

This is a graphic of our Context Diagram. All the core elements are shown here for the GUI project, as well as the communication between the port, third party applications and user. Our DFD's were started with this model.

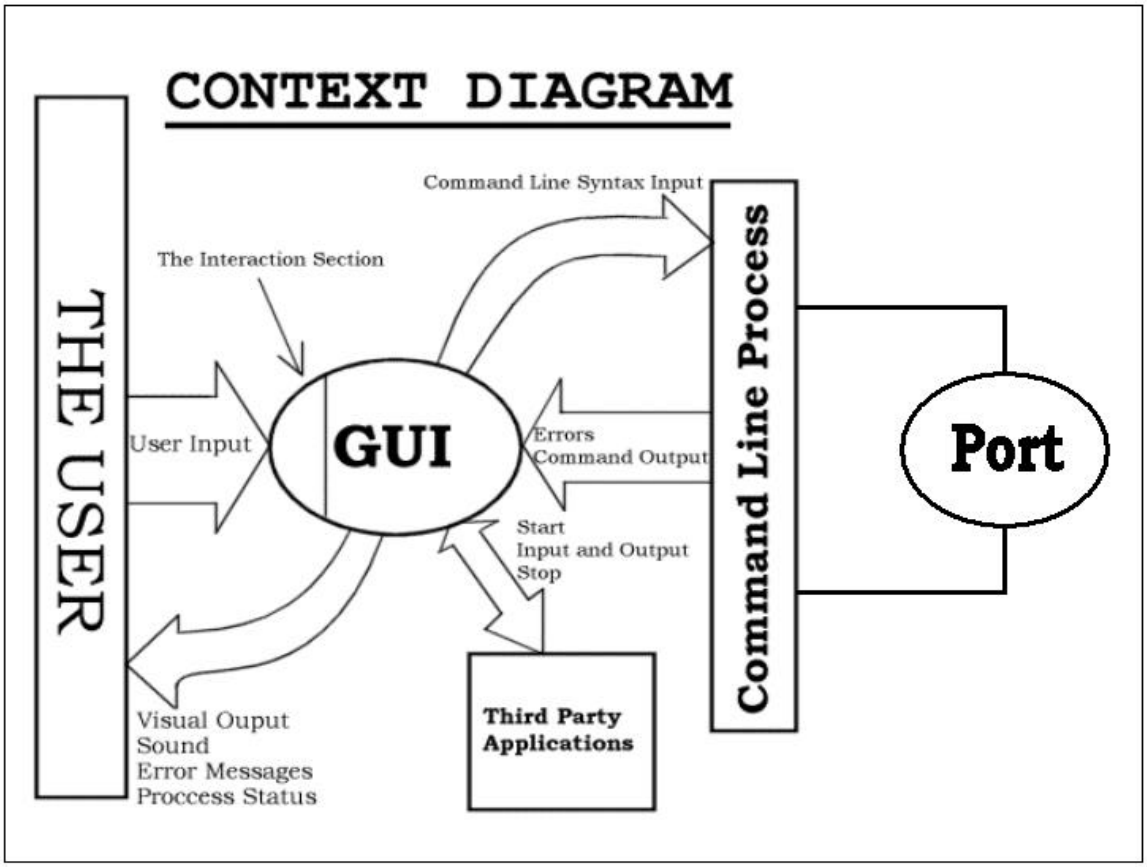

Figure 1: Context Diagram

3.2 Level 1 Data Flow Diagram and P-specs

Figure 2 shows the Level 1 Data Flow Diagram (DFD) for the CSPN-CSPL GUI. The external items depicted are:

1. The user who provides input to the system
2. The port (which represents the CSPN executable) to which the GUI sends commands
3. Third party applications which are invoked from the GUI

In this DFD the GUI itself is broken into several modules. The main control module is shown in bubble 1. It intercepts user input and sends those commands to the other GUI modules. Thus, it acts as the main control center of the program.

Bubble 2 shows the Port Translator, which organizes commands into a format that the CSPN port executable can read and sends those commands to it. Bubble 3 shows the Display Image module, which is responsible for decoding GIF, JPEG and BMP files and painting them on the screen. Finally, Bubble 4 depicts the GUI Processor which assists the command translator in transforming raw user commands into an organized format that can be used by the rest of the program.

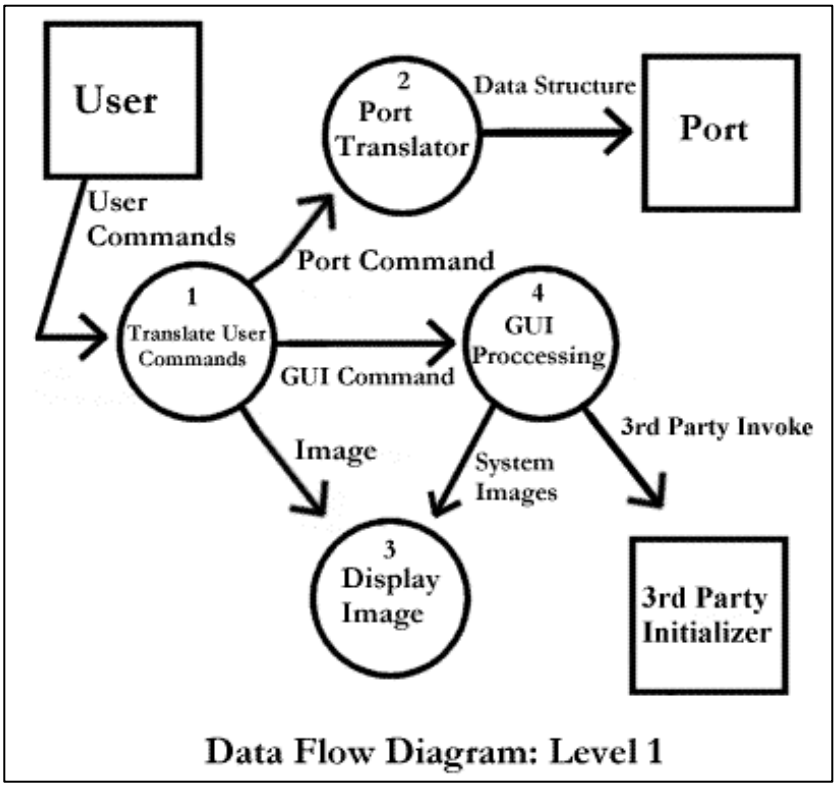

Figure 2: Level 1 DFD

GUI Element PSPECS:

1. Translate User Commands:

```
if user_command != valid_command
  then display error message
else if user command = port command
        Send_to (port_translator)
else if user command = GUI command
        Send_to (GUI_Proccessing)
else if user command = display image
        Send_to (Display Image)
```

2. Port Translator:

```
Parse user_command and compare to database
Start structure generator
Send structure to port
```

3. Display Image

```
Check header for file information
Invoke image viewer and display image in window
```

4. GUI Processing

```
If state_change
        Update window – invoke image view for system BMP
Else if input_command
        Translate command
        Update window – invoke image view for system BMP
Else if 3rd party command
        Initialize 3rd party application and wrap in a window
```

3.3 Level 2 and 3 Data Flow Diagrams

Figure 3 shows the Level 2 expansion of Bubble 1 from Figure 2. This includes the command translator module. User commands are first routed to an error checker, which notifies the user if he or she has specified invalid input. Valid commands are then entered into a command database, and are read out again by a command distributor. This distributor dispatches commands depending on whether they are meant for the GUI, the CSPN port or a third party application.

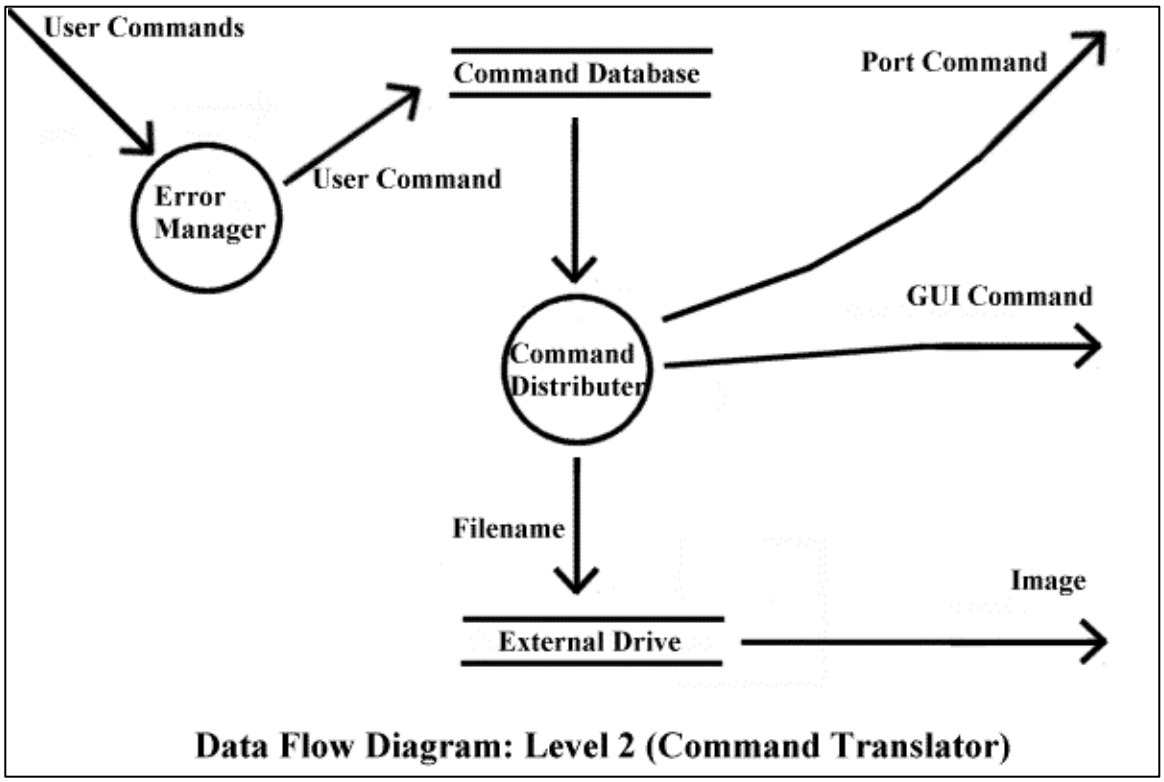

Figure 3: Level 2 DFD Part I

Figure 4 shows an expansion of the Port Translator from the Level 1 DFD. The translator executes in a linear fashion. It takes the port command given to it by the command translator, parses that command and builds a structure out of the data. This structure is recognized and understood by the port, allowing the GUI and the port to communicate effectively. This data structure is then sent to the port.

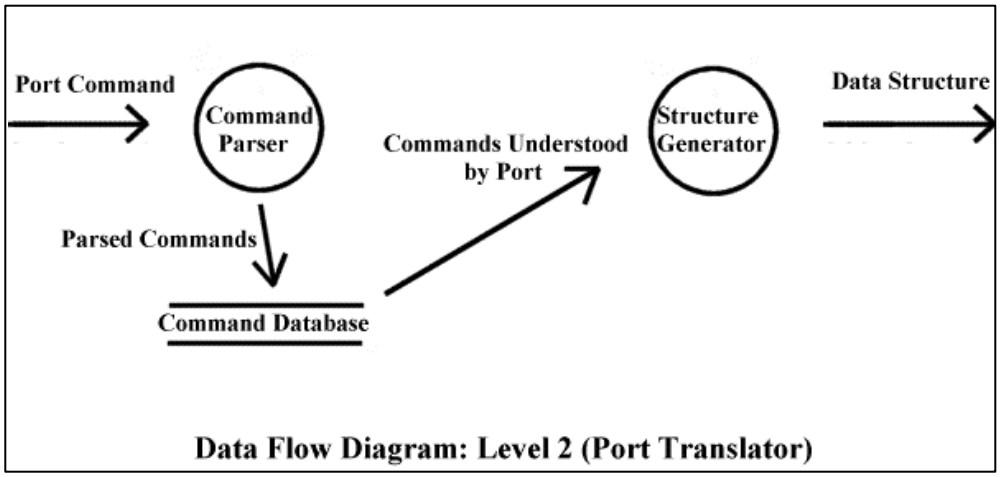

Figure 4: Level 2 DFD Part II

Figure 5 illustrates the GUI Processing module in more detail than Figure 2 showed. This element is responsible for determining when a screen update is necessary in the Input Translator bubble, and for performing the refresh in the Image Updater bubble. This section draws on the System Images store for elements like icon pictures.

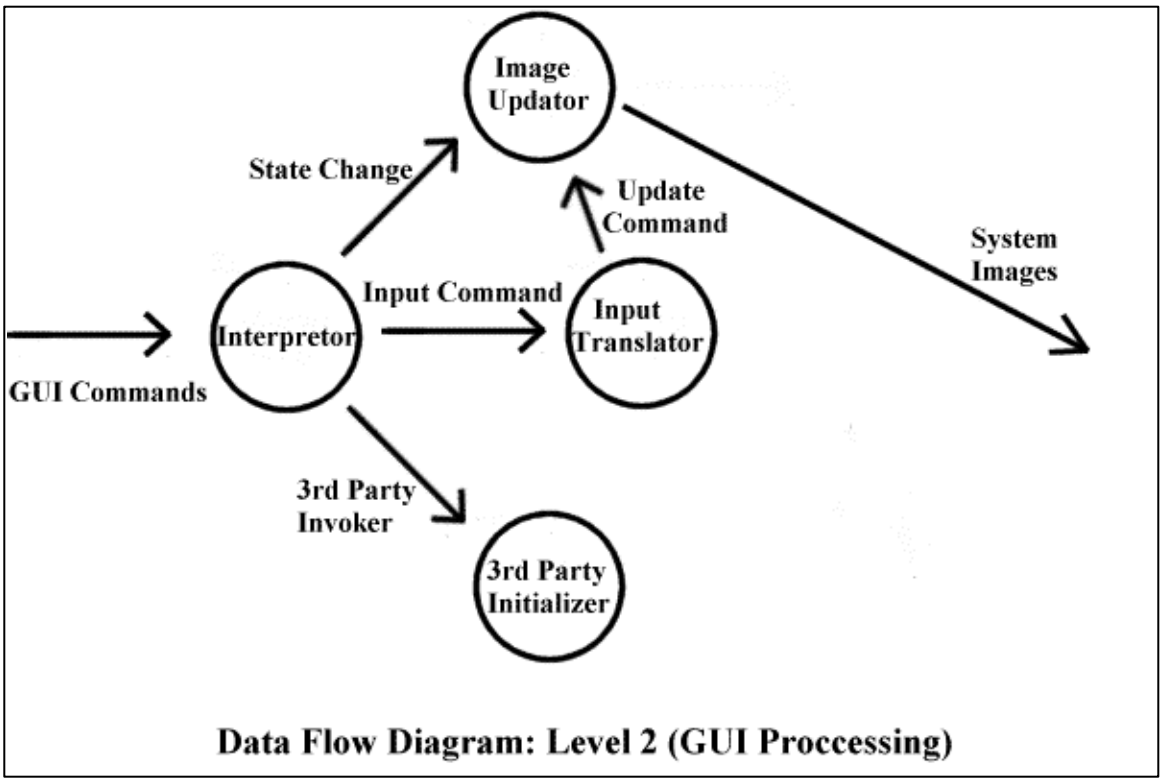

Figure 5: Level 2 DFD Part III

3.3.2 Level 3 Data Flow Diagram

The Level 3 DFD shows an update in the Level 2 DFD Part III (GUI Processing). First, the processor determines whether the GUI command is meant for a third party application or for the main GUI. If it is meant for a third party program, then it creates an invocation command for that application and executes it. It makes sure that any specified command line parameters are passed to the application. If the GUI command is meant for the CSPN-CSPL GUI itself then the translator executes the command, determines whether a screen refresh is necessary and performs that refresh.

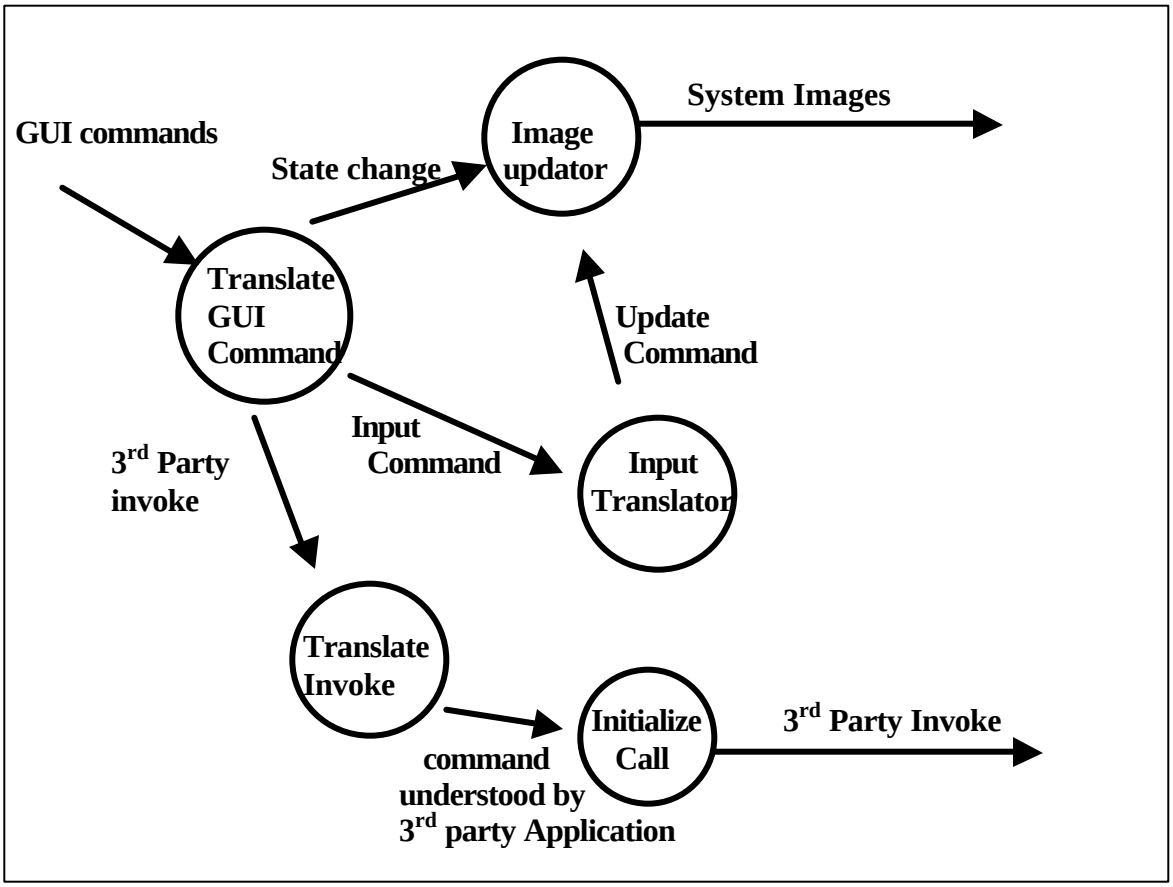

Figure 6 DFD: Level 3 (GUI Processing)

3.4 Transaction Analysis

- In transaction analysis, the complete system of Data Flow Diagram (DFD) is divided into parts. The purpose of transaction analysis is to separate the components of a large design into a network of cooperating subsystems.

3.4.1 Partitions of the System DFD

When the user enters a command, the user command translator module will translate the command to identify whether it should go to port translator module, GUI processing module, or image and sound generator module. The modules are shown in Figure 3.x. The user command translator module will direct the image display module to open the image for the user. If there is an error with the command, the user command translator will direct image and sound generator to display error message with associated sound to the user. Port translator module shall produce the output to the port as a data structure that is understood by the port. GUI processing module will determine if the command is associated with GUI processing such as text editor, state changes or third party application.

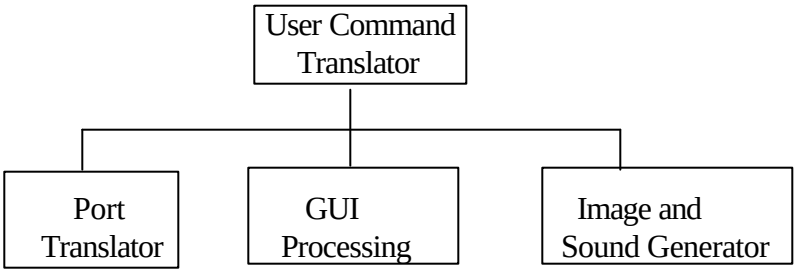

Figure 7 Partitions of the System DFD

3.4.2 Identified Transactions

The identified transactions below are examples of comprehensive transactions that account for all of the five items described earlier. The important parts of GUI are WIMP (window, icon, mouse, pointer), toolbar, status bar, context sensitive help, and message handling. Each example of transaction that is presented here consists of five important items that are:

- The **event** in the systems environment that causes the transaction to occur;
- The **stimulus** that is applied to the system to inform it about the event;
- The **activity** that is performed by the system as a result of the stimulus;
- The **response** that this generates in terms of output from then system;
- The **effect** that this has upon the environment.

3.4.2.1 CSPN-CSPL GUI Transaction Analysis

This transaction example shows that the port translator sends the user command from the user and sends the port response through the GUI processing module.

Event: Run CSPN port using option button

Stimulus: Open CSPN file that is known to be correct and click on the icon for running CSPN.

Activity: The port shall receive the command from port translator and GUI will receive the response back from the port.

Response: The option button will be on and off position as the mouse is clicked.

Effect: Current system status will be displayed on the status bar.

3.4.2.2 Text Editor and Image Viewer Transaction Analysis

This transaction example shows that the GUI processing module receives the user command to open file. This is also to verify that the interaction between mouse input and the system are successful.

Event: Open file using Menu option
Stimulus: Click File menu and choose to Open.
Activity: A dialog box shall pop up for the user to enter the filename and its format.
Response: The filename is received after the user enters an existing filename, its format, and click on the confirmation button.
Effect: The file shall be opened correctly.

3.4.2.3 Pointer Functionality Transaction Analysis

This transaction shows that the system communicate to the user through the pointer that is moved by the user. GUI processing module receives the user command from the located pointer and give the output to the user through the image and sound generator module.

Event: Test the pointer functionality on the toolbar
Stimulus: Place the mouse pointer on one of the icon on the toolbar.
Activity: The input shall be processed in GUI processing module and determine that a ToolTip shall appear describing the option's purpose.
Response: The ToolTip shall disappear as the mouse pointer is moved away from the option box.
Effect: GUI shall be able to display the correct ToolTip when the mouse pointer is placed on other option boxes.

3.4.2.4 Message Handling Transaction Analysis

This transaction shows that the system will give an error message when the user enter wrong command or to ask the user for confirmation.

Event: Test Message Handling in the Text Editor
Stimulus: Modify the existing text file, and save it with an existing filename.
Activity: GUI processing module will identify the filename, and send a message to image generator.
Response: A confirmation box shall appear to verify if the user wants to overwrite the existing file or to cancel the activity. If sounds are enabled, a certain sound shall be played.

Effect: The user response shall be received and processed in the user command translator module.

3.4.2.5 Keyboard Functionality Transaction Analysis

This transaction shows that the system will receive user command from keyboard input.

Event: Test Context Sensitive Help using keyboard accelerator
Stimulus: Press the keyboard equivalent for opening the help topic on a selected window (e.g. GUI/main window, Text Editor window, Image Viewer).
Activity: The help topic for the corresponding window shall be displayed.
Response: A specific help topic shall be displayed after it is chosen.
Effect: The user shall be able to open help topics in other windows using keyboard accelerator.

3.4.2.6 Invoking Third Party Transaction Analysis

This transaction is significantly modified from the existing system. There will be an icon presenting the third party invoker.

Event: Open third party application
Stimulus: Click on the third party icon in the toolbar.
Activity: User command translator will send the command to the GUI processing module and is processed as the third party invoker, and send the open command to the third party.
Response: If there is no error from the third party, the third party application shall be displayed, otherwise, an error message shall be displayed to the user.
Effect: The user shall be able to use the third party application outside the GUI main window.

3.5 Transform Analysis

Our transforms are based on the central GUI transform as seen by the context diagram (Figure 1). This transform is the basis for the information flow from one transform to another. All messages and information will be handled or distributed through this point. This central GUI object can be thought as a switchboard type system, where a central element (translate user commands transform, as seen in the level 1 DFD: Figure 2).

From this switch board transform we have three main transforms we can access: the port transform, image display transform and finally the GUI system transform. Here is a brief description of each:

Port Transform: This is the interpreter for communication between the GUI and port. It acts as a barrier or firewall where the GUI is able to interpret commands coming in and out of user to the port. It protects errors committed by the port or the user from crashing the GUI.

Image Display Transform: This is a separate executable file, and will be developed as a separate application. The GUI will treat this transform as a 3rd party application, and will check for errors committed in the application. The image display transform will assume the GUI is always functional, since a breakdown of the GUI will not allow the user to interact with the transform no matter if it remains operational or not.

GUI System Transform: The GUI transform acts as a maintenance crew. Updating the system to reflect changes made by the user or other transform will be its primary responsibility. Limited image commands will be sent to the image transform to display simple images for startup dialogs and things of this nature.

3.5.1 Structure Chart

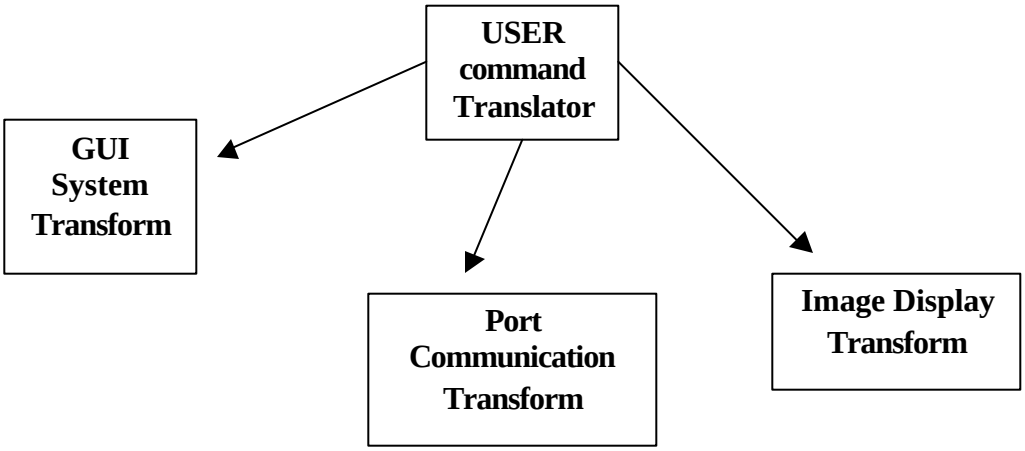

Figure 8 Structure Chart of Main Transforms

3.6 Design Decision Log

Table 1 shows a record of significant design decisions that have been made during the course of this project.

Date Introduced	Feature
9-6-1999	Base project elements are introduced from the Marketing Perceived User Needs document.
9-17-1999	System Model is derived from Marketing needs. SRS document completed and sent to development.
9-24-1999	Image viewer is decided to be a 3rd party implementation and is to remain separated from the GUI.
10-11-1999	SRS 3.0 is sent to development and is the foundation for all design criteria.
10-24-1999	With development ahead of schedule it is decided that improvements shall be made to the text editor. It will now included a last load file table and toolbar. Gostview shall now be supported.
10-25-1999	The development of the system GUI shall be done in API C Vs MFC C++. This will allow for a more robust system.

Figure 9 Design Decision Log Table

4. *References*

- IEEE Standard 1016-1987: Design Notebook Standard, 1987
- Pressman, Roger S.: Software Engineering A Practitioner's Approach, The McGraw-Hill Companies, Inc 1997
- Sheldon, F. T.: Design Notebook Guidelines and Standards, 1999
- Sheldon, F. T.: General Documentation Style Guidelines and Standards, 1999
- Sheldon, F. T.: Project 2 Marketing Perceived User Needs, 1999
- Sommerville, Ian: Software Engineering, Addison-Wesley 1995
- Petzold, Charles: Programming Windows 95, Microsoft Press 1996
- Johnsonbaugh & Kalin: Applications Programming in ANSI C, Prentice-Hall, Inc 1996

5. Glossary

ASCII – American Standard Code for Information Interchange. It defines a standard way for representing characters on computers.

API – Application Program Interface. It defines a standard way that programs work with pull-down menus, dialog boxes, and windows. Microsoft Windows, OS/2, and the Macintosh are examples of API in action.

BMP - .bmp file extension. Bitmap graphic file format

CSP – Communicating Sequential Processes (process algebra)

CSPN – CSP-to-Stochastic Petri Nets

CSPL – C-like Stochastic Petri Net Language

DFD - Data Flow Diagram: to provide an indication of how data are transformed as they move through the system, and to depict the functions that transform the data flow.

GIF (Graphics Interchange Format) - gif file extension. A special file format developed by CompuServe to store graphics that all computers can use. Up to 256 colors.

GUI – Graphical User Interface

JPEG (Joing Photgraphic Experts Group) - .jpg file extension. Up to 16 million colors. For storing digitized photographs, the JPEG standard offers sharper resolution but the older GIF standard works bette for plain line drawings or clip art.

PostScript – A page description language developed by Adobe Systems.

Pspec – Process Specification. Describes a program process in a functional manner

RTM – Requirements Traceability Matrix. This matrix will be used by the test team to verify each requirement individually.

SRS – Software Requirements Specification. A deliverable produced during the software engineering process.

SPNP - the product generated by our associated Port team, which carries out the actual work of parsing CSPL files.

Appendix A: Data Dictionary

These 29 data dictionary items describe data fields that will be used in the CSPN-CSPL GUI. Each dictionary entry contains the following information:

Field Name: The name of the field as it will be referred to in design documents
Field Description: A brief explanation of the purpose of this field
Data Type: The data type of the field (e.g. text, integer)
Field Size: How large the field is, measured in bytes
Usage Guidelines: Describes how the field is used by the program
Table: Indicates the table to which this data dictionary element belongs

GUI

Field Name: Icon Name
Field Description: This field identifies a graphics to be used.
Data Type: Text
File Size: 15
Usage Guidelines: Provide the name of the graphics file to be displayed in the information page.

Field Name: Image Display Area
Field Description: This field provides BMP, JPG, or GIF image files.
Data Type: Graphics image
Field Size: Undefined
Usage Guidelines: Provides Image displaying for BMP files. Decodes the image and displays it as a system image.
Table: ImageViewer Table

Field Name: How to Apply
Field Description: This field describes the method of obtaining the service or making application.
Data Type: Text
Field Size: Long
Usage Guidelines: Provide instructions on how to obtain or make application for the program.

Field Name: Last Update
Field Description: This field gives the date of the last update with information of the program.
Data Type: Date
File Size: Date

Usage Guidelines: Provide the date of the last update in the information area for this program. PROGRAM DESCRIPTION that will alert users to major changes in a program.

Table: Program

Field Name: Program Name

Field Description: This field provides the name or title of the program as defined by the team responsible for administering the program.

Data Type: Text

Field Size: 100

Usage Guidelines: Provide the program name or title. Use the name as defined by the team responsible for administering the program. It is the team's responsibility to develop internal procedures to ensure that its subordinate units do not assign the same PROGRAM NAME to different programs to other programs.

Table: Program

Field Name: Program Purpose

Field Description: This field describes why the program exists. It may include the origin and lineage of the program and related programs.

Data Type: Text

File Size: Long

Usage Guideline: The PROGRAM PURPOSE field provides a description of the function of the program. It should give the reason why the program was created.

Table: Program Detail

Field Name: Program Description

Field Description: This field describes the nature of the program and is intended to be a brief summary of the program. It gives user more information on the program.

Data Type: Text

Field Size: Long

Usage Guidelines: Provide a brief description of the individual program, which gives information relating to the general nature and scope of the program. Information to be included in the PROGRAM DESCRIPTION is likely to be found in existing program documentation or guidelines.

Table: Program Detail

Field Name: Program Web Address

Field Description: This field is the World Wide Web address for pages containing additional information on this program.

Data Type: Text

Field Size: 75

Usage Guidelines: This filed will be used to generate a hyperlink to team Web pages. Provide the address for pages on the World Wide Web that contain additional program information.

Table: Program

Field Name: Title

Field Description: This field lists the title of the program.

Data Type: Text

Field Size: 50

Usage Guidelines: Provide the title of the principal. Give the name of Instructor name and team name. Also give the team name.

Table: Program

Field Name: Version Number

Field Description: This field gives the version number of the program.

Data Type: Number

Field Size: Short

Usage Guidelines: Provide the version number of a program to alert users to major changes in the program.

Table: Program

Image Viewer

Field Name: Image Display Area

Field Description: This field provide BMP, JPG, or GIF image file.

Data Type: Graphics image

Field Size: 600x400

Usage Guidelines: Provide Image display BMP, JPG, and GIF file, decode the image and display it in a window on the screen.

Table: ImageViewer Table

Field Name: How to Apply

Field Description: This field describes the method of obtaining the service or making application.

Data Type: Text

Field Size: Long

Usage Guidelines: Provide instructions on how to obtain or make application for the program.

Field Name: Last Update

Field Description: This field gives the date of the last update with information of the program.

Data Type: Date

File Size: Date

Usage Guidelines: Provide the date of the last update in the information area for this program. PROGRAM DESCRIPTION that will alert users to major changes in a program.

Table: Program

Field Name: Program Name

Field Description: This field provides the name or title of the program as defined by the team responsible for administering the program.

Data Type: Text

Field Size: 100

Usage Guidelines: Provide the program name or title. Use the name as defined by the team responsible for administering the program. It is the team's responsibility to develop internal procedures to ensure that its subordinate units do not assign the same PROGRAM NAME to different programs to other programs.

Table: Program

Field Name: Program Purpose

Field Description: This field describes why the program exists. It may include the origin and lineage of the program and related programs.

Data Type: Text

File Size: Long

Usage Guideline: The PROGRAM PURPOSE field provides a description of the function of the program. It should give the reason why the program was created.

Table: Program Detail

Field Name: Program Description

Field Description: This field describes the nature of the program and is intended to be a brief summary of the program. It gives user more information on the program.

Data Type: Text

Field Size: Long

Usage Guidelines: Provide a brief description of the individual program, which gives information relating to the general nature and scope of the program. Information to be included in the PROGRAM DESCRIPTION is likely to be found in existing program documentation or guidelines.

Table: Program Detail

Field Name: Program Web Address

Field Description: This field is the World Wide Web address for pages containing additional information on this program.

Data Type: Text

Field Size: 75

Usage Guidelines: This filed will be used to generate a hyperlink to team Web pages. Provide the address for pages on the World Wide Web that contain additional program information.

Table: Program

Field Name: Title

Field Description: This field lists the title of the program.

Data Type: Text

Field Size: 50

Usage Guidelines: Provide the title of the principal. Give the name of Instructor name and team name. Also give the team name.

Table: Program

Field Name: Version Number

Field Description: This field gives the version number of the program.

Data Type: Number

Field Size: Short

Usage Guidelines: Provide the version number of a program to alert users to major changes in the program.

Table: Program

Text Editor

Field Name: Icon Name

Field Description: This field identifies a graphics to be used.

Data Type: Text

File Size: 15

Usage Guidelines: Provide the name of the graphics file to be displayed in the information page.

Field Name: How to Apply

Field Description: This field describes the method of obtaining the service or making application.

Data Type: Text

Field Size: Long

Usage Guidelines: Provide instructions on how to obtain or make application for the program.

Field Name: Last Update

Field Description: This field gives the date of the last update with information of the program.

Data Type: Date

File Size: Date

Usage Guidelines: Provide the date of the last update in the information area for this program. PROGRAM DESCRIPTION that will alert users to major changes in a program.

Table: Program

Field Name: Program Name

Field Description: This field provides the name or title of the program as defined by the team responsible for administering the program.

Data Type: Text

Field Size: 100

Usage Guidelines: Provide the program name or title. Use the name as defined by the team responsible for administering the program. It is the team's responsibility to develop internal procedures to ensure that its subordinate units do not assign the same PROGRAM NAME to different programs to other programs.

Table: Program

Field Name: Program Purpose

Field Description: This field describes why the program exists. It may include the origin and lineage of the program and related programs.

Data Type: Text

File Size: Long

Usage Guideline: The PROGRAM PURPOSE field provides a description of the function of the program. It should give the reason why the program was created.

Table: Program Detail

Field Name: Program Description

Field Description: This field describes the nature of the program and is intended to be a brief summary of the program. It gives user more information on the program.

Data Type: Text

Field Size: Long

Usage Guidelines: Provide a brief description of the individual program, which gives information relating to the general nature and scope of the program. Information to be included in the PROGRAM DESCRIPTION is likely to be found in existing program documentation or guidelines.

Table: Program Detail

Field Name: Program Web Address

Field Description: This field is the World Wide Web address for pages containing additional information on this program.

Data Type: Text

Field Size: 75

Usage Guidelines: This filed will be used to generate a hyperlink to team Web pages. Provide the address for pages on the World Wide Web that contain additional program information.

Table: Program

Field Name: Text Edit Area

Field Description: This field contains the ASCII text for CSPN.

Data Type: Text

Field Size: Infinity

Usage Guidelines: Provide the ASCII text display and provide context sensitive highlighting of CSPN keywords. And also provide the current line number and column position of the cursor for user to work comfortable.

Table: Text Editor Table

Field Name: Title

Field Description: This field lists the title of the program.

Data Type: Text

Field Size: 50

Usage Guidelines: Provide the title of the principal. Give the name of Instructor name and team name. Also give the team name.

Field Name: Version Number

Field Description: This field gives the version number of the program.

Data Type: Number

Field Size: Short

Usage Guidelines: Provide the version number of a program to alert users to major changes in the program.

Appendix B: Project Schedule

The Mellow Yellow team is responsible for the GUI project. The members of Mellow Yellow and their responsibilities are as follows:

- Kerry Hammil: Team leader and writer
- Alex Velkov: Writer
- Nick Gunder: GUI developer (interface to the port project and overall functionality of the GUI)
- Ryoji Noda: Architectural developer (text editor and image viewer)
- Brock Rogers: Graphical artist and GUI testing
- Azrina Hussin: Architectural testing

This table shows the proposed schedule for the completion of the various project deliverables

Start Date	*End Date*	*Deliverable/Milestone*	*Responsible Team Members*
9/6	**9/10**	Project Plan	
9/10	**9/17**	First Draft of SRS (Software Requirements Specification)	
9/10	**9/17**	Set up team Web site	Kerry Hammil
9/10	**9/17**	Set up team FTP site	Nick Gunder
9/10	**9/17**	Set up development environment on home computers	
9/17	**10/11**	Prototype of CSPN GUI	Nick Gunder and Ryoji Noda
9/17	**10/11**	Final Draft SRS	
10/13		PDR (Preliminary Design Review)	
10/25		CDR (Critical Design Review)	
11/1		Design Notebook Due	
11/29		Demonstration and Test Report Due	
12/6		User Manual Due	

Table 2: Project Schedule

Appendix C: Requirements Traceability Matrix

Note: DFD identifiers that refer to 3rd party are referring to the executable text editor that will be attached with the GUI system package.

System Level Requirement ID	Sub-System Level Requirement ID	DFD Identifiers	Module Name(s)	Verification Method
ME1	ME1	3+4	Menus	I
ME1	ME2	3+4	Menus	I, D
ME1	FM1	3+4	File Menu Items	I
ME1	FM2	3+4	File Menu Items	I
ME1	FM3	3+4	File Menu Items	I, D
ME1	FM4	4	File Menu Items	I, D, A
ME1	FM5	3rd Party	File Menu Items	I
ME1	FM6	3rd Party	File Menu Items	I, D
ME1	FM7	3rd Party	File Menu Items	I
ME1	FM8	3rd Party	File Menu Items	I, D
ME1	FM9	3rd Party	File Menu Items	I
ME1	FM10	4	File Menu Items	I, D
ME1	FM11	3rd Party	File Menu Items	I
ME1	FM12	3rd Party	File Menu Items	I, D
ME1	FM13	3rd Party	File Menu Items	I, D
ME1	ED1	4	Edit Menu Items	I
ME1	ED2	4	Edit Menu Items	I
ME1	ED3	4	Edit Menu Items	I
ME1	ED4	4	Edit Menu Items	I
ME1	ED5	4	Edit Menu Items	I
ME1	ED6	4	Edit Menu Items	I
ME1	ED7	4	Edit Menu Items	I
ME1	ED8	4	Edit Menu Items	I
ME1	ED9	4	Edit Menu Items	I, D
ME1	ED10	4	Edit Menu Items	I
ME1	ED11	4	Edit Menu Items	I, D
ME1	ED12	4	Edit Menu Items	I
ME1	ED13	4	Edit Menu Items	I
ME1	ED14	4	Edit Menu Items	I
ME1	TO1	4	Tool Menu Items	I
ME1	TO2	4	Tool Menu Items	I, D
ME1	TO3	4	Tool Menu Items	I
ME1	TO4	4	Tool Menu Items	I, D

System Level Requirement ID	Sub-System Level Requirement ID	DFD Identifiers	Module Name(s)	Verification Method
ME1	TO5	2	Tool Menu Items	A
ME1	TO6	4	Tool Menu Items	I
ME1	TO7	4	Tool Menu Items	I, D
ME1	TO8	4	Tool Menu Items	I, D
ME1	WI1	4	Window Menu Items	I
ME1	WI2	4	Window Menu Items	I, D
ME1	WI3	4	Window Menu Items	I, D
ME1	HE1	4	Help Menu Items	I
ME1	HE2	4	Help Menu Items	I, D
TB1	TB1	4	Toolbar	I
TB1	TB2	4	Toolbar	I
TB1	TB3	4+3rd Party	Toolbar	I, D
TB1	TB4	4	Toolbar	I
TB1	TB5	3rd Party	Toolbar	I, D
TB1	TB6	4	Toolbar	I
TB1	TB7	3rd Party	Toolbar	I
TB1	TB8	3rd Party	Toolbar	I, D
TB1	TB9	3rd Party	Toolbar	I
TB1	TB10	3rd Party	Toolbar	I, D
TB1	TB11	4+3rd Party	Toolbar	I
TB1	TB12	4+3rd Party	Toolbar	I
TB1	TB13	4	Toolbar	I
TB1	TB14	4	Toolbar	I
TB1	TB15	4+3rd Party	Toolbar	I
TB1	TB16	4	Toolbar	I
TB1	TB17	3rd Party	Toolbar	I
TB1	TB18	3rd Party	Toolbar	I
TB1	TB19	3rd Party	Toolbar	I, D
TB1	TB20	4	Toolbar	I, D
TB1	TB21	4	Toolbar	I
TB1	TB22	2	Toolbar	D, A
GW1	GW1	4	Window Options	I, D
GW1	GW2	4	Window Options	I
GW1	GW3	4	Window Options	I
GW1	GW4	4	Window Options	I, D
GW1	GW5	1	Window Options	I, D
CSH1	CSH1	1	Help Options	I, D
CSH1	CSH2	1	Help Options	I, D
CSH1	CSH3	1+3rd Party	Help Options	I, D

System Level Requirement ID	Sub-System Level Requirement ID	DFD Identifiers	Module Name(s)	Verification Method
CSH1	CSH4	1+3	Help Options	I, D
CSH1	CSH5	1+4	Help Options	I, D
SB1	SB1	4	Status Bar	I, D
DB1	OFD1	1	Dialog Boxes	I, D
DB1	OFD2	4+3rd Party	Dialog Boxes	I, D
DB1	OFD3	4+3rd Party	Dialog Boxes	I
DB1	OFD4	4+3rd Party	Dialog Boxes	I, D
DB1	OFD5	4+3rd Party	Dialog Boxes	I
DB1	OFD6	4	Dialog Boxes	I, D
DB1	OFD7	4+3rd Party	Dialog Boxes	I
DB1	OFD8	4+3rd Party	Dialog Boxes	I
DB1	OFD9	4+3rd Party	Dialog Boxes	I
DB1	SFD1	3rd Party	Displaying Files	I, D
DB1	SFD2	3rd Party	Displaying Files	I
DB1	SFD3	3rd Party	Displaying Files	I, D
DB1	SFD4	3rd Party	Displaying Files	I
DB1	SFD5	3rd Party	Displaying Files	I, D
DB1	SFD6	3rd Party	Displaying Files	I, D
DB1	SFD7	ND	Displaying Files	D, A
DB1	SFD8	3rd Party	Displaying Files	I
DB1	SFD9	3rd Party	Displaying Files	I
DB1	SFD10	3rd Party	Displaying Files	I, D
DB1	FTD1	3rd Party	Finding Text	I, D
DB1	FTD2	3rd Party	Finding Text	I, D
DB1	FTD3	3rd Party	Finding Text	I, D
DB1	FTD4	3rd Party	Finding Text	I
DB1	FTD5	3rd Party	Finding Text	I, D
DB1	FTD6	3rd Party	Finding Text	I, D
DB1	FTD7	3rd Party	Finding Text	I, D
DB1	FTD8	3rd Party	Finding Text	I
DB1	RTD1	3rd Party	Finding Text	I, D
DB1	RTD2	3rd Party	Finding Text	I, D
DB1	RTD3	3rd Party	Finding Text	I, D
DB1	RTD4	3rd Party	Finding Text	I, D
DB1	RTD5	3rd Party	Finding Text	I
DB1	PD1	3rd Party	Printing	I, D
DB1	PD2	3rd Party	Printing	I, D
DB1	PD3	3rd Party	Printing	I, D
DB1	PD4	3rd Party	Printing	I, D

System Level Requirement ID	Sub-System Level Requirement ID	DFD Identifiers	Module Name(s)	Verification Method
DB1	AD1	4	About	I
DB1	AD2	4	About	I
DB1	OD1	3rd Party	Options	I, D
DB1	OD2	3rd Party	Options	I, D
DB1	OD3	3rd Party	Options	I, D
DB1	OD4	3rd Party	Options	I, D
DB1	OD5	4	Options	I, D
DB1	OD6	4	Options	I, D
DB1	CP1	4	CSPN Port Options	I, D
DB1	CP2	4	CSPN Port Options	I, D
DB1	CP3	4	CSPN Port Options	I, D
DB1	CP4	4	CSPN Port Options	I, D
DB1	CP5	4	CSPN Port Options	I, D
DB1	CP6	4	CSPN Port Options	I, D
DB1	CP7	4	CSPN Port Options	I, D
DB1	CP8	4	CSPN Port Options	I, D
DB1	CP9	4	CSPN Port Options	I, D
DB1	CP10	4	CSPN Port Options	I, D
DB1	CP11	4	CSPN Port Options	I, D
DB1	CP12	4	CSPN Port Options	I, D
TEW1	TEW1	3rd Party	Text Editor Window	I, D
TEW1	TEW2	3rd Party	Text Editor Window	I
TEW1	TEW3	3rd Party	Text Editor Window	I
TEW1	TEW4	3rd Party	Text Editor Window	I, D
TEW1	TEW5	3rd Party	Text Editor Window	I
TEW1	TV1	3rd Party	Text Properties	I, D
TEW1	TV2	3rd Party	Text Properties	I
TEW1	TV3	3rd Party	Text Properties	I, D
TEW1	TV4	3rd Party	Text Properties	I, D
IVW1	IVW1	3	Image Viewer	I, D
IVW1	IVW2	3	Image Viewer	I
IVW1	IVW3	3	Image Viewer	I
IVW1	IVW4	3	Image Viewer	I
IVW1	IVW5	3	Image Viewer	I, D
IVW1	IVW6	3	Image Functions	I, D, A
IVW1	IVW7	Not Supported	Image Functions	D, A

Table 3: Requirements Traceability Matrix

KEY: T = by Test, A = by Analysis, I = Inspection, and An = by Analogy

Table 1 shows the Requirements Traceability Matrix (RTM). The hierarchy of system level requirements and their corresponding component level requirements are parts of the RTM. The component, labeled subsystem requirements, are said to be traceable to the system level which are their parent requirements. The subsystem level requirements are so called derived requirements.

The RTM provides one place where all requirements are listed, where they all located within the program, and finally how each is to be certified. One column is provided to mark if each of the requirements has been tested as a matter of tracking the progress and for use in the demonstration.

There are various verification methods to show that a particular requirement has been satisfied.
A= Analysis method is used to show that non-functional requirements have been met like response time.
I= Inspection method is used to show that something like using specific coding standards has been adhered to.
D= Demonstration method is used to show that by say running the program that it computes the correct output.
An= Analogy is used when all else fails. This method in some reasonably rigorous fashion shows or proves that a particular requirement(s) is met through some indirect means. For example, if the program displays the correct symbol then it shows that some other requirements that cannot be reasonably isolated, has also been satisfied.

Appendix D: Identified Test Cases

These test cases were first identified during the development of the Preliminary Design Review and the Critical Design Review. They are a sample of the full range of test cases that will eventually be developed for the GUI and described in the Test Report. As the development of the GUI program proceeds, the test team will run each of these cases on the compiled executable to verify that they work initially, and the cases will be rerun on each new major build to ensure that they continue to work. If a test case is found not to work, the test team will report the failure to the development team immediately, describing which test case failed and what behavior was produced when the test case was run.

D.1 Run CSPN port using option button

Purpose(s): To verify the functionality of the option button. To verify that the CSPN-CSPL GUI transaction is working correctly.
Method: Open a CSPN file that is known to be correct and run CSPN using option button provided on the toolbar.
Expected output: The output shall be the same as if the CSPN is run using keyboard accelerator and using mouse input in the menu option.

D.2 Open file using Menu option

Purpose(s): To verify that the menu selection under the title bar functions correctly. To verify that the system can receive input from the mouse. To verify that the system can receive string input from the user.
Method: Click File menu and choose to Open.
Expected output: A text file or an image file shall be displayed on a window in the main GUI. The output shall be the same as if the CSPN is run using keyboard accelerator.

D.3 Test the pointer functionality on the toolbar

Purpose(s): To verify the pointer functionality.
Method: Place the mouse pointer on one of the icon on the toolbar.
Expected output: The ToolTip shall appear describing the option's purpose or function. The ToolTip shall disappear as the mouse pointer is moved away from the option box.

D.4 Test Message Handling in the Text Editor

Purpose(s): To verify message-handling functionality. To verify the system can interact with the user and treat the user input and current system status as new input to the system.

Method: Modify the existing text file, and save it with an existing filename.

Expected output: A confirmation box shall appear to verify if the user wants to overwrite the existing file or to cancel the activity. If sounds are enabled, a certain sound shall be played.

D.5 Test Context Sensitive Help using keyboard accelerator

Purpose(s): To verify the system can handle context sensitive help. To verify the associated keyboard accelerator functions correctly.

Method: Press the keyboard equivalent for opening the help topic on a selected window (e.g. GUI/main window, Text Editor window, Image Viewer).

Expected output: The help topic for the corresponding window shall be displayed and a specific help topic shall be displayed after it is chosen. The output shall be the same as if the help topic is opened using mouse.

D.6 Open third party application

Purpose(s): To verify the system can invoke the third party application when the user enters the command. To verify the GUI can interact with the third party application when needed. To verify the GUI can send the message from the third party application when needed.

Method: Click on the third party icon in the toolbar.

Expected output: If there is no error from the third party, the third party application shall be displayed, otherwise, an error message shall be displayed to the user. The output shall be same as the command is entered using the keyboard accelerator and mouse selection in the main menu.