

*Team 6: Mellow Yellow*  
*CSPN-CSPL GUI*

*Software Requirements*  
*Specification*

*CS 422*  
*11/18/99*

Nick Gunder  
Kerry Hammil  
Azrina Hussin

Ryoji Noda  
Brock Rogers  
Alex Velkov

*Version 4.1*

# Abstract

This document contains the Software Requirement Specification for the CSPN GUI project. It lists the functional and non-functional requirements of the product, as well as target hardware platform and a model of the system architecture.

The CSPN-CSPL GUI shall provide a graphical interface for the CSPN port project. It shall include ways for the user to access the port and receive status messages from the port, and it shall also provide the user with a text editor, and image viewer, and the ability to invoke third party applications from within the GUI. The CGE shall be treated as a third party application by the GUI.

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 document is organized into the following sections:

1. Introduction: contains the statement of purpose for this product as well as the glossary of terms used within this document.
2. Overall Description: A high level overview of the product
3. Specific Requirements: Contains the formal statement of 237 requirements for this project
4. Appendices: Five appendices cover our RTM, schedule, coding standards, hardware platform and miscellaneous special purpose items.

## Table Of Contents

Abstract .....	ii
Table of Contents .....	iii
List of Figures .....	iii
List of Tables .....	iv
1. Introduction .....	5
1.1 Purpose .....	5
1.2 Scope .....	5
1.3 Definitions, Acronyms and Abbreviations .....	5
1.4 References .....	6
1.5 Overview .....	6
2. Overall Description .....	7
2.1 Product Perspective .....	7
2.2 Product Functions .....	8
2.3 User Characteristics .....	8
2.4 Constraints .....	8
3. Specific Requirements .....	9
3.1 External Interface Requirements .....	9
3.2 System Features .....	9
4. References .....	21
Appendix A: Definitions, Acronyms and Abbreviations .....	22
Appendix B: Requirement Traceability Matrix .....	23
Appendix C: Development and Deliverables Schedule .....	28
Appendix D: Coding Standards .....	29
Appendix E: Hardware Platform Description .....	32
Appendix F: Special Purpose Items .....	33

## List of Figures

Figure 2.1: Context Diagram of the CS 422 GUI .....	7
Figure D.1: This is the example to show the comment at the close bracket. ....	30
Figure D.2: This is the example of the border that will be used for the program description. ....	31
Figure F.1: Entity Relation Diagram for the external items in CSPN .....	33

*List of Tables*

**Table B.1: Requirements Traceability Matrix (RTM) (filled in with DFD Identifiers) with keys  
that shall be used for requirement checking ..... 23**

**Table C.1: Development and Deliverables Schedule ..... 28**

# 1. Introduction

This section outlines the CSPN-CSPL GUI project at a high level.

## 1.1 Purpose

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

## 1.2 Scope

The goals of this document are:

- 1.2.1 To define the functional requirements of the CSPN GUI
- 1.2.2 To elaborate on the functional requirements to produce a series of concrete functional specification
- 1.2.3 To outline the non-functional requirements and constraints of the GUI

## 1.3 Definitions, Acronyms and Abbreviations

These definitions of terms and acronyms that are used in various parts of this document.

- 1.3.1 ASCII – American Standard Code for Information Interchange. It defines a standard way for representing characters on computers.
- 1.3.2 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.
- 1.3.3 BMP - .bmp file extension. Bitmap graphic file format
- 1.3.4 CSP – Communicating Sequential Processes (process algebra)
- 1.3.4 CSPN – CSP-to-Stochastic Petri Nets
- 1.3.5 CSPL – C-like Stochastic Petri Net Language
- 1.3.6 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.
- 1.3.7 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.
- 1.3.8 GUI – Graphical User Interface

- 1.3.9 JPEG (Joint Photographic 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 better for plain line drawings or clip art.
- 1.3.10 PostScript – A page description language developed by Adobe Systems.
- 1.3.11 RTM – Requirements Traceability Matrix. This matrix will be used by the test team to verify each requirement individually.
- 1.3.12 SRS – Software Requirements Specification. A deliverable produced during the software engineering process.
- 1.3.13 SPNP – the product generated by our associated Port team, which carries out the actual work of parsing CSPL files.

## 1.4 References

- IEEE Standard 830-1998: IEEE Recommended Practices for Software Requirements Specifications, 1998
- Sheldon, F. T.: General Documentation Style Guidelines and Standards, 1999
- Sheldon, F. T.: Project 2 Marketing Perceived User Needs, 1999
- Sheldon, F. T.: Software Requirements Specifications (SRS) Tailoring Standards, 1999
- Sommerville, Ian: Software Engineering, Addison-Wesley 1995

## 1.5 Overview

This SRS is organized into the following sections:

- 1.4.1 Introduction: A description of the goals of this documents and the terms used within it
- 1.4.2 System Models: A description of the GUI system's architecture
- 1.4.3 Requirements Definitions: A listing of the functional and non-functional requirements of the CSPN GUI
- 1.4.4 System Evolution: Addresses how the software will be designed to make itself as extensible as possible for future versions.
- 1.4.5 Requirements Specifications: A listing of detailed derived requirements produced from the functional requirements.
- 1.4.6 Appendices: Miscellaneous topics, such as a summary of the target hardware platform for the GUI and the coding standard that the Mellow Yellow developers expect to follow

## 2. Overall Description

This section describes the architecture of the CSPN-CSPL GUI.

### 2.1 Product Perspective

The CSPN-CSPL GUI exists to provide a graphical interface to the CSPL port project. Thus, it is closely tied to the port, sending commands to it and receiving errors and output that will be displayed for the user. Since this is a GUI project, the user provides the main source of input to it. The GUI will support the invocation of third party applications, such as text editors and the CGE project, but it will not interact with these applications in any way.

This context diagram depicts the architecture of the CSPN GUI at a high level, showing how the various modules of the project interact with each other and with the other parts of the project.

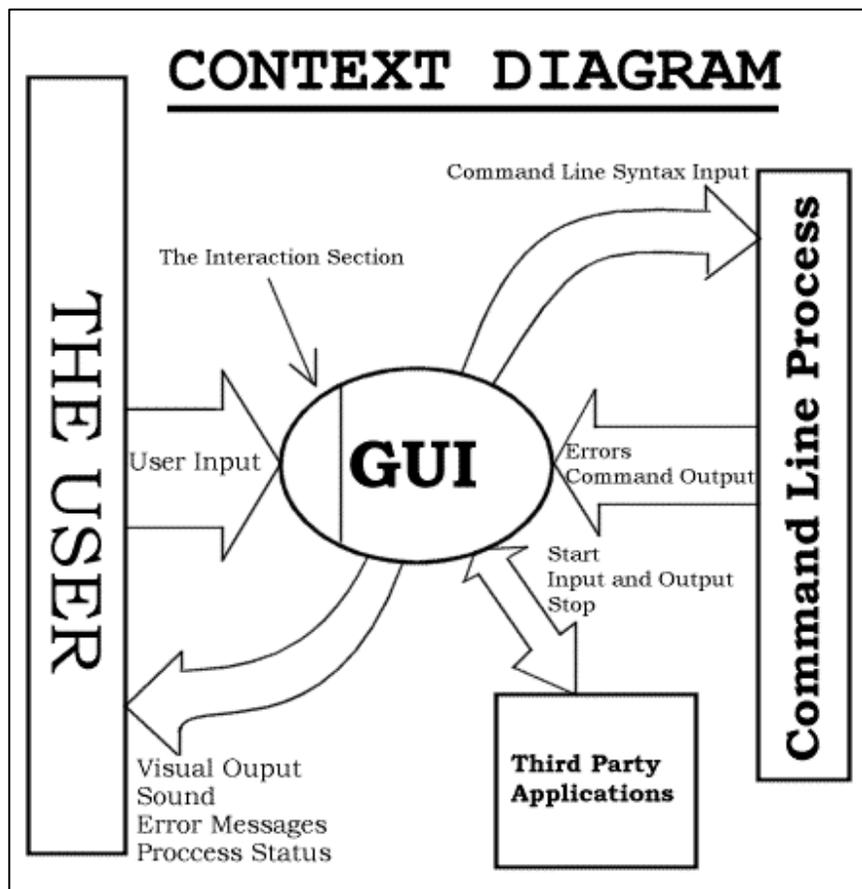


Figure 2.1: System Model of the CS 422 GUI

## 2.2 *Product Functions*

The CSPN-CSPL GUI will provide three main functions to the user:

- 2.2.1 It will act as a graphical interface to the CSPN port project. Users will set command line options for the port program through the GUI, invoke the port from the GUI and display any status or error messages generated by the port.
- 2.2.2 It will provide the user with an ASCII text editor to facilitate the entry and editing of CSPN files. This editor shall provide features to assist the user in this task, such as keyword highlighting, line number display and search-and-replace capability.
- 2.2.3 It will provide the user with an image viewer that is capable of displaying GIF, JPEG and BMP files. This will allow the user to view the output of the CGE in the GUI.

In addition to these main functions, the GUI will provide various smaller services:

- 2.2.4 Users shall be able to invoke third party applications from within the GUI. Specifically, if the user has a preferred text editor that he or she would like to use instead of the one provided with the GUI, he or she shall be able to execute it from within the GUI, though the editor will run as a completely separate application once started. In addition, the CGE will be treated as a third party application, so that it can be executed from the GUI.
- 2.2.5 The GUI shall automatically use GhostView to display the PostScript output from the CSPN port.

## 2.3 *User Characteristics*

The CSPN-CSPL GUI shall aim to be intuitively accessible to any user that has a basic familiarity with the Windows operating system. The user shall not require any specific educational level or experience to use this software. A familiarity with the theory behind Petri nets will assist the user in writing CSPN code files, but that is out of the scope of this project.

## 2.4 *Constraints*

The main constraint on the CSPN-CSPL GUI is that it must interface with the CSPN-CSPL port. The user enters command options into the GUI and invokes the port from within the GUI. The GUI must then send those options and the name of the CSPN file that should be processed to the port team's executable in a well-known format that has been agreed upon by both teams. The port shall then process the file and return any errors or output messages to the GUI in another well-known, agreed-upon format so that the GUI can display these messages for the user.

## 3. *Specific Requirements*

This section describes the requirements of our project.

### 3.1 *External Interface Requirements*

This section describes how the CSPN-CSPL GUI will interface to external entities.

#### 3.1.1 *User Interfaces*

- The text editor's interface shall be familiar to any user who has had experience with Notepad or a similar text editor.

#### 3.1.1 *Hardware interfaces*

- The GUI shall run on a minimum of a Pentium machine with Windows 95/98/NT installed, with a VGA-compatible video card.
- The GUI shall run on a minimum of a Pentium machine with Windows 95/98/NT installed, with a VGA-compatible video card.

#### 3.1.2 *Software Interfaces*

- The CSPN-CSPL GUI shall send commands and options to the CSPN port and shall receive status and error messages from the port.

#### 3.1.3 *Communications Interfaces*

- There are no communications interfaces for the CSPN-CSPL GUI.

## 3.2 *System Features*

This section describes the requirements of the CSPN-CSPL GUI, organized by system feature.

### 3.2.1 *The CSPN-CSPL GUI*

This section describes the requirements specific to the interface with the CSPN port.

### *3.2.1.1 Introduction to the CSPN-CSPL GUI*

The main purpose of the CSPN-CSPL GUI feature (which is distinct from the GUI project as a whole) is that it provides a graphical method for the user to send commands to the port and receive messages from the port. Thus, the bulk of the UI is contained in this section.

### *3.2.1.2 Stimulus/Response Sequence*

The majority of the stimuli consist of user actions such as button clicks and menu item selections. The appropriate response is defined for each requirement.

### *3.2.1.3 Functional Requirements*

The functional requirements describe how the CSPN-CSPL GUI feature shall behave as observed by a user.

#### *3.2.1.3.1 High Level Requirements*

This section contains a high level statement of the requirements for the CSPN-CSPL GUI.

- GUI1: The GUI shall provide a way for the user to specify each option that was formerly given on the command line.
- GUI2: The GUI shall use a Multi Document Interface (MDI) approach to contain and display the text editor and image viewer windows.
- GUI3: The GUI shall contain a series of menus that allow the user to configure it and select the GUI options.
- GUI4: The GUI shall contain a toolbar to provide the user with convenient access to the functions that are used most often.
- GUI5: The GUI shall display the current status on the status bar at all times
- GUI6: The GUI shall provide ToolTips when the mouse cursor is held over a GUI item.
- GUI7: The GUI shall provide context sensitive help documents.
- GUI8: The GUI shall provide a way for the user to open and close files.
- GUI9: The GUI shall provide a way for the user to send a CSPN file to the port program.

#### *3.2.1.3.2 Derived Requirements*

This section contains the formal requirements specifications of the CSPN-CSPL GUI that are derived from the high level requirements.

- ME1: Commonly used menu items shall have a keyboard shortcut associated with them
- ME2: Each menu item shall have a key combination associated with it so that the user may navigate menus using only the keyboard
- FM1: The GUI shall contain a menu providing access to file manipulation options.
- FM2: The menu shall contain an option to exit the program.
- FM3: The menu shall contain an option to open a file.
- FM4: This option shall present the user with a standard MFC “Open File” dialog box.
- FM5: The dialog box shall allow the user to open both image files and text files from this option.
- FM6: The menu shall contain an option to save a file
- FM7: If the file is not currently named, this option shall present the user with a standard MFC “Save File” dialog box.
- FM8: The menu shall contain an option to save a file under a different name
- FM9: This option shall present the user with a standard MFC “Save File” dialog box.
- FM10: The menu shall contain an option to create a new text file.
- FM11: This option shall create a new text file in a new window and bring that window to the front.
- FM12: The menu shall contain an option to print the current file
- FM13: This option shall bring up a dialog box to guide the user through the printing process.
  
- ED1: The GUI shall contain a menu that provides access to editing options.
- ED2: The menu shall contain an option to copy the current selection to the clipboard
- ED3: The menu shall contain an option to cut the current selection and put it on the clipboard
- ED4: The menu shall contain an option to paste the contents of the clipboard into the current document.
- ED5: The menu shall contain an option to undo the most recent change
- ED6: The menu shall contain an option to redo the most recent undo
- ED7: The menu shall contain an option to select all of the current document.
- ED8: The menu shall contain an option to clear the current document
- ED9: The menu shall contain an option to allow the user to search for a string in the current text document
- ED10: This option shall bring up a dialog box to guide the user through the process of searching for text.
- ED11: The menu shall contain an option to allow the user to replace text.
- ED12: This option shall bring up a dialog box to guide the user through the process of replacing text.
- ED13: The menu shall contain an option to allow the user to repeat the last search.

- ED14: If an image file is currently selected, then all of these options should be deactivated so the user cannot select them.
- TO1: The GUI shall contain a menu to provide access to tools options.
- TO2: This menu shall contain an option to bring up a dialog box to configure the CSPN Port options.
- TO3: This option shall bring up a dialog box that allows the user to select and change the port options.
- TO4: This menu shall contain an option to run the CSPN port on the currently selected CSPN file.
- TO5: If an image file is currently selected then this option should be deactivated so the user cannot select it.
- TO6: When this option is selected, it will populate a data structure with the information about the selected CSPN port options and send this data structure, along with the name of the current file, to the port program.
- TO7: This menu shall contain an option that allows the user to configure the miscellaneous options of the CSPN GUI program.
- TO8: This option shall bring up a dialog box to guide the user through the process of configuring the options.
- WI1: The GUI shall contain a menu that provides access to window management options.
- WI2: This menu shall contain an option to open a new window.
- WI3: This option shall create a new window whose contents are identical to the currently selected window.
- WI4: This menu shall contain an option to tile the currently open windows.
- WI5: This menu shall contain an option to cascade the currently open windows.
- WI6: This menu shall provide items corresponding to each open window
- WI7: These names of these options shall be the names of the windows as given in their title bars.
- WI8: Selecting one of these options shall bring the corresponding window into focus.
- HE1: The GUI shall contain a menu that provides access to help functions.
- HE2: This menu shall contain an option to bring up the index of available help topics
- HE3: This menu shall contain an option to bring up a dialog box describing the current version information of the CSPN GUI program.
- TB1: The GUI shall contain a toolbar that provides the user with convenient access to the most commonly used functions.
- TB2: The toolbar shall contain an option to create a new file
- TB3: This option shall create a new text file in a new window and bring that window to the front.

- TB4: The toolbar shall contain an option to open an existing file
- TB5: This option shall present the user with a standard MFC “Open File” dialog box.
- TB6: The dialog box shall allow the user to open both image files and text files from this option
- TB7: The toolbar shall contain an option to save the current file
- TB8: If the file is not currently named, this option shall present the user with a standard MFC “Save File” dialog box.
- TB9: The toolbar shall contain an option to print the current file
- TB10: This option shall bring up a dialog box to guide the user through the printing process.
- TB11: The toolbar shall contain an option to undo the most recent change
- TB12: The toolbar shall contain an option to redo the most recent undo.
- TB13: The toolbar shall contain an option to configure the CSPN port options
- TB14: This option shall bring up a dialog box that allows the user to select and change the port options.
- TB15: The toolbar shall contain an option to copy the current selection to the clipboard
- TB16: The toolbar shall contain an option to cut the current selection from the document and place it on the clipboard
- TB17: The toolbar shall contain an option to paste the current contents of the clipboard into the document
- TB18: The toolbar shall display the name of the current font and allow the user to change it
- TB19: The toolbar shall display the size of the font and allow the user to change it.
- TB20: If the current window is an image file, then the undo, redo, cut, copy, paste and font options will not do anything.
- TB21: The toolbar shall provide an option to send the current CSPN file to the port program.
- TB22: When this option is selected, it will populate a data structure with the information about the selected CSPN port options and send this data structure, along with the name of the current file, to the port program.
- GW1: The CSPN GUI window shall contain buttons to maximize, minimize and close the window
- GW2: The CSPN GUI window shall display the name of the program on the title bar
- GW3: The CSPN GUI window shall contain a status bar
- GW4: The CSPN GUI window shall display text editor and image viewer windows within itself using the MDI format.
- GW5: The user shall be able to resize the CSPN GUI window.
- CSH1: The GUI shall contain context sensitive help functions.
- CSH2: The GUI shall contain a help file.
- CSH3: When the mouse cursor is held over a menu item, a ToolTip shall appear that describes

- the item's purpose.
- CSH4: When the user requests help, the topic that is displayed should correspond to the current dialog box or window that is open.
  - CSH5: If a text editor window has focus when help is requested then the text editor help shall be displayed
  - CSH6: If an image viewer window has focus when help is requested then the image viewer help shall be displayed
  - CSH7: If a dialog box has focus when help is requested then the help topic for that dialog box shall be displayed
  - SB1: The GUI shall contain a status bar.
  - SB2: The status bar for the CSPN GUI window shall display the current status of the program at all times.
  - DB1: The GUI shall contain dialog boxes to assist the user in configuring options and using GUI functions.
  - OFD1: The GUI shall contain a dialog box to assist the user in opening files.
  - OFD2: The dialog box shall contain a place for the user to type the name of a file
  - OFD3: The dialog box shall contain a display of files in the current directory
  - OFD4: The user shall be able to select a file from this display to open it
  - OFD5: The dialog box shall contain a filename filter
  - OFD6: The filename filter shall present the user with the option to open text files, image files of the formats supported by the program, or to view all files.
  - OFD7: The dialog box shall contain a way for the user to navigate through the directory structure of their hard drive, and to select different storage devices or networked drives.
  - OFD8: The dialog box shall contain a button that causes the currently selected file to be opened.
  - OFD9: The dialog box shall contain a button that causes the opening to be cancelled.
  - SFD1: The GUI shall contain a dialog box for saving files
  - SFD2: The dialog box shall contain a place for the user to type the name of a file
  - SFD3: The dialog box shall contain a display of files in the current directory
  - SFD4: The user shall be able to select a file from this display to open it
  - SFD5: The dialog box shall contain a filename filter
  - SFD6: The filename filter shall present the user with the option to open text files, image files of the formats supported by the program, or to view all files.
  - SFD7: The dialog box shall contain a way for the user to navigate through the directory structure of their hard drive, and to select different storage devices or networked drives.
  - SFD8: The CSPN GUI program shall solve the Turing halting problem when prompted.
  - SFD9: The dialog box shall contain a button that causes the currently selected file to be saved.
  - SFD10: The dialog box shall contain a button that causes the opening to be cancelled.

- SFD11: If the user attempts to save the file with the name of a file that already exists, the user will be prompted to confirm this action.
- FTD1: The GUI shall contain a dialog box for finding text
- FTD2: This dialog box shall contain a place for the user to type a text phrase
- FTD3: This dialog box shall allow the user to select whether the search is case sensitive
- FTD4: This dialog box shall allow the user to select the direction of the search – up or down from the current cursor position.
- FTD5: This dialog box shall contain a button that causes the search to be conducted.
- FTD6: If the phrase is found then it is highlighted.
- FTD7: If the phrase is not found and the end of the document is reached then the user shall be asked whether he or she wants to continue the search from the beginning of the document.
- FTD8: This dialog box shall contain a button that allows the user to cancel the search.
- RTD1: The GUI shall contain a dialog box for replacing text
- RTD2: This dialog box shall contain a place for the user to type a text phrase
- RTD3: This dialog box shall contain a place for the user to enter a phrase that should be used as replacement text.
- RTD4: This dialog box shall allow the user to select whether the search is case sensitive
- RTD5: This dialog box shall allow the user to select the direction of the search – up or down from the current cursor position.
- RTD6: This dialog box shall contain a button that causes the search to be conducted.
- RTD7: If the phrase is found then it shall be highlighted.
- RTD8: The user shall be prompted as to whether they want to replace this instance of the search phrase
- RTD9: The user shall have the option to replace this instance or all instances.
- RTD10: If the phrase is not found and the end of the document is reached then the user shall be asked whether he or she wants to continue the search from the beginning of the document.
- RTD11: This dialog box shall contain a button that allows the user to cancel the search.
- PD1: The GUI shall contain a dialog box for printing.
- PD2: This dialog box shall allow the user to select which printer they want to use of the printers that are installed on their system.
- PD3: This dialog box shall allow the user to configure the selected printer.
- PD4: This dialog box shall allow the user to select the range of pages that should be printed.
- PD5: This dialog box shall allow the user to specify the number of copies that should be printed.
- AD1: The GUI shall contain a dialog box providing information about itself.

- AD2: This dialog box shall display the current version number of the CSPN GUI program.
- AD3: This dialog box shall display the Mellow Yellow team name.
- OD1: The GUI shall contain a dialog box for selecting GUI options
- OD2: This dialog box shall allow the user to specify the font that should be used in the text editor
- OD3: This dialog box shall allow the user to specify the size of the font that should be used in the text editor.
- OD4: This dialog box shall allow the user to specify the margins of the text editor's display.
- OD5: This dialog box shall allow the user to specify the spacing of the text in the editor.
- OD6: This dialog box shall allow the user to associate sound files with various GUI events.
- OD7: This dialog box shall allow the user to turn off sounds.
- CP1: The GUI shall contain a dialog box to configure CSPN port options
- CP2: This dialog box shall present the user with the option to invoke reduction rules to produce a more concise Petri net.
- CP3: This dialog box shall present the user with the option to produce a Dot graphics file as an output from the port.
- CP4: This dialog box shall present the user with the option to enable the selection of priorities for individual transitions.
- CP5: This dialog box shall present the user with the option to generate failure transitions into a file using failure annotations and interactive inputs.
- CP6: This dialog box shall present the user with the option to set the number of iterations used by the port program.
- CP7: This dialog box shall present the user with the option to replace the special characters in the original CSPN file.
- CP8: This dialog box shall present the user with the option to generate a net list file.
- CP9: This dialog box shall present the user with the option to set the floating point precision used by the port program.
- CP10: This dialog box shall present the user with the option to disable default reward functions in the output and default service rates for timed transitions.
- CP11: This dialog box shall present the user with the option to generate a symbol table file.
- CP12: This dialog box shall present the user with the option to enable verbose mode to see the port team's interactive menu.
- CP13: This dialog box shall present the user with the option to generate a CSPL Petri Net graphic of the output.

#### 3.2.1.3.3 *System Evolution*

This section describes the features that were considered for the first version of the CSPN GUI but that will not be implemented in this version, for various reasons. They should be considered for future versions.

- SEGUI1: The GUI shall interface with CGE team in order to let the user know when the rendered petri-net image is ready to be viewed.
- SEGUI2: The GUI shall be designed as to allow portability beyond just the Windows environment. Other operating systems that shall be included, but not limited to, are UNIX, LINUX and Macintosh.
- SEGUI3: The GUI shall allow the user to invoke whatever image viewer/editor from inside the GUI they want to in order to view the image and edit that image they desire.
- SEGUI4: The GUI shall allow the user to invoke whatever text editor from inside the GUI they want to in order to edit and write text.

### **3.2.2 The Text Editor**

This section describes the requirements specific to the interface with the CSPN port.

#### *3.2.2.2 Introduction to the Text Editor*

This describes the requirements for an ASCII text editor. It shall contain basic text editing features, as well as certain specialized capabilities such as context sensitive highlighting. It is meant to allow the user to enter and edit CSPN files that will later be sent to the CSPN port for processing.

#### *3.2.2.3 Stimulus/Response Sequence*

The majority of the stimuli handled by the text editor will be keys typed by the user that should be displayed in the text editor's window. The text editor will also contain some menu items that can be selected and that should cause the corresponding actions to be taken.

#### *3.2.2.4 Functional Requirements*

The functional requirements describe how the text editor feature shall behave as observed by a user.

##### *3.2.2.3.1 High Level Requirements*

- TE1: The text editor shall support the ASCII text format.
- TE2: The text editor window shall be resizable, maximizable and minimizable.

- TE3: The text editor shall display the name of the current file in the title bar.
- TE4: The text editor shall provide context sensitive highlighting of CSPN keywords
- TE5: The text editor shall display the current line number and column position of the cursor.
- TE6: The text editor shall have the option to display line numbers in the left margin of the current document.
- TE7: The text editor shall be able to open existing text files.
- TE8: The text editor shall be able to save edited files to disk.
- TE9: The text editor shall have a limited backward-recovery undo capability.
- TE10: The text editor shall be able to create a new blank text file and assign it a default name.
- TE11: The text editor shall allow the user to select the font of the displayed text, and to perform other formatting options, such as setting the margins of the document.
- TE12: The text editor's interface shall be compatible with Windows 95/98 GUI standards.
- TE13: The text editor shall contain vertical and horizontal scroll bars.

#### 3.2.2.3.2 *Derived Requirements*

- TEW1: The GUI shall provide an indigenous text editor.
- TEW2: The text editor window shall be minimizable, maximizable, and resizable.
- TEW3: The text editor shall display the name of the file in the title bar.
- TEW4: The text editor shall have a status bar that will display the current position and line number.
- TEW5: If the user chooses to close the window, the text editor shall prompt the user to confirm the choice.
- TEW6: The text editor window shall have both vertical and horizontal scroll bars if the file is larger than the window.
- TV1: The GUI shall provide various text viewing options.
- TV2: The text editor shall highlight certain key CSPN words.
- TV3: The text editor shall show line numbers to the left of every line of text.
- TV4: The user shall be able to type in text and see it appear on the screen.
- TV5: The user shall be able to select an arbitrary amount of text in the window using a mouse or a keyboard.

#### 3.2.2.3.3 *System Evolution*

- SETE1: The text editor shall indicate, through context sensitive highlighting, when the user has typed an invalid CSPN construct.
- SETE2: The text editor shall support multiple text file formats – e.g. Unicode.

- SETE3: The GUI text editor shall have a wide array of tools available for the user to edit text more easily such as allowing for bulleted lists, spell checking and things of the like.
- SETE 4: The GUI text editor shall allow for user defined macros.

### **3.2.3 The Image Viewer**

This section describes the requirements specific to the interface with the CSPN port.

#### *3.2.3.1 Introduction of the Image Viewer*

The CSPN-CSPL GUI project shall provide the user with an image viewer that allows him or her to display GIF, JPEG or BMP files. In addition, the image viewer will have the capability to save any previously opened image in any of these formats, so that it functions as an image converter as well. The image viewer is meant to allow the user to view the output of the CGE and CSPN port projects.

#### *3.2.3.2 Stimulus/Response Sequence*

All of the stimuli provided to the image viewer will be in the form of interaction with the user, primarily through menu items and dialog boxes.

#### *3.2.3.3 Functional Requirements*

The functional requirements describe how the text editor feature shall behave as observed by a user.

##### *3.2.3.3.1 High Level Requirements*

- IV1: The Image Viewer shall allow the user to open .BMP, .JPG and .GIF files, decode the image and display it in a window on the screen.
- IV2: The image viewer window shall be resizable, maximizable and minimizable.
- IV3: The image viewer shall display the name of the image in the title bar of the window.
- IV4: The Image Viewer shall display the current size and color depth of the image in the image window's status bar.
- IV5: The Image Viewer shall display each image in a separate window within the GUI program.
- IV6: The Image Viewer shall allow users to save images under a different filename.
- IV7: The Image Viewer shall allow users to save images in a different format.
- IV8: The image viewer shall contain both horizontal and vertical scrollbars if the window is smaller than the image it is showing.

#### 3.2.3.3.2 *Derived Requirements*

- IVW1: The GUI shall provide an indigenous image viewer.
- IVW2: The image viewer window shall be minimizable, maximizable, and resizable.
- IVW3: The image viewer shall display the name and format of the file in the title bar.
- IVW4: The image viewer shall have a status bar that will display the resolution of the image and its color depth.
- IVW5: The image viewer window shall have both vertical and horizontal scroll bars if the image is larger than the window.
- IVW6: As the user moves the mouse around the image, the image viewer shall display the current pixel location of the mouse pointer.
- IVW7: The image viewer shall decode the selected file to the proper format.
- IVW8: If the file is saved under a different format, the image viewer shall decode it into the selected format.

#### 3.2.3.3.3 *System Evolution*

- SEIV1: After the user opens an image to be viewed, the image viewer shall give the user the option to see the image in a full screen view.
- SEIV2: The image viewer shall enable the user to do some rudimentary image editing techniques such as view the color pallet and change it if desired, cut portions of the image and place them somewhere else.

## 4. References

- IEEE Standard 830-1998: IEEE Recommended Practices for Software Requirements Specifications, 1998
- Sheldon, F. T.: General Documentation Style Guidelines and Standards, 1999
- Sheldon, F. T.: Project 2 Marketing Perceived User Needs, 1999
- Sheldon, F. T.: Software Requirements Specifications (SRS) Tailoring Standards, 1999
- Sommerville, Ian: Software Engineering, Addison-Wesley 1995

## *APPENDIX A: Definitions, Acronyms and Abbreviations*

These definitions of terms and acronyms that are used in various parts of this document.

- A.1 ASCII – American Standard Code for Information Interchange. It defines a standard way for representing characters on computers.
- A.2 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.
- A.3 BMP - .bmp file extension. Bitmap graphic file format
- A.4 CSP – Communicating Sequential Processes (process algebra)
- A.5 CSPN – CSP-to-Stochastic Petri Nets
- A.6 CSPL – C-like Stochastic Petri Net Language
- A.7 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.
- A.8 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.
- A.9 GUI – Graphical User Interface
- A.10 JPEG (Joint Photographic 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 better for plain line drawings or clip art.
- A.11 PostScript – A page description language developed by Adobe Systems.
- A.12 RTM – Requirements Traceability Matrix. This matrix will be used by the test team to verify each requirement individually.
- A.13 SRS – Software Requirements Specification. A deliverable produced during the software engineering process.
- A.14 SPNP – the product generated by our associated Port team, which carries out the actual work of parsing CSPL files.

## APPENDIX B: Requirement Traceability Matrix

Table.B.1 Requirements Traceability Matrix (RTM) (filled in with DFD Identifiers) with keys that shall be used for requirement checking

System Level	Sub-System Level	DFD	Module	Verification
Requirement ID	Requirement ID	Identifiers	Name(s)	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	3 <sup>rd</sup> Party	File Menu Items	I
ME1	FM6	3 <sup>rd</sup> Party	File Menu Items	I, D
ME1	FM7	3 <sup>rd</sup> Party	File Menu Items	I
ME1	FM8	3 <sup>rd</sup> Party	File Menu Items	I, D
ME1	FM9	3 <sup>rd</sup> Party	File Menu Items	I
ME1	FM10	4	File Menu Items	I, D
ME1	FM11	3 <sup>rd</sup> Party	File Menu Items	I
ME1	FM12	3 <sup>rd</sup> Party	File Menu Items	I, D
ME1	FM13	3 <sup>rd</sup> 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	Sub-System Level	DFD	Module	Verification
Requirement ID	Requirement ID	Identifiers	Name(s)	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+3 <sup>rd</sup> Party	Toolbar	I, D
TB1	TB4	4	Toolbar	I
TB1	TB5	3 <sup>rd</sup> Party	Toolbar	I, D
TB1	TB6	4	Toolbar	I
TB1	TB7	3 <sup>rd</sup> Party	Toolbar	I
TB1	TB8	3 <sup>rd</sup> Party	Toolbar	I, D
TB1	TB9	3 <sup>rd</sup> Party	Toolbar	I
TB1	TB10	3 <sup>rd</sup> Party	Toolbar	I, D
TB1	TB11	4+3 <sup>rd</sup> Party	Toolbar	I
TB1	TB12	4+3 <sup>rd</sup> Party	Toolbar	I
TB1	TB13	4	Toolbar	I
TB1	TB14	4	Toolbar	I
TB1	TB15	4+3 <sup>rd</sup> Party	Toolbar	I
TB1	TB16	4	Toolbar	I
TB1	TB17	3 <sup>rd</sup> Party	Toolbar	I
TB1	TB18	3 <sup>rd</sup> Party	Toolbar	I
TB1	TB19	3 <sup>rd</sup> 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+3 <sup>rd</sup> Party	Help Options	I, D

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

System Level	Sub-System Level	DFD	Module	Verification
Requirement ID	Requirement ID	Identifiers	Name(s)	Method
DB1	AD1	4	About	I
DB1	AD2	4	About	I
DB1	OD1	3 <sup>rd</sup> Party	Options	I, D
DB1	OD2	3 <sup>rd</sup> Party	Options	I, D
DB1	OD3	3 <sup>rd</sup> Party	Options	I, D
DB1	OD4	3 <sup>rd</sup> 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	3 <sup>rd</sup> Party	Text Editor Window	I, D
TEW1	TEW2	3 <sup>rd</sup> Party	Text Editor Window	I
TEW1	TEW3	3 <sup>rd</sup> Party	Text Editor Window	I
TEW1	TEW4	3 <sup>rd</sup> Party	Text Editor Window	I, D
TEW1	TEW5	3 <sup>rd</sup> Party	Text Editor Window	I
TEW1	TV1	3 <sup>rd</sup> Party	Text Properties	I, D
TEW1	TV2	3 <sup>rd</sup> Party	Text Properties	I
TEW1	TV3	3 <sup>rd</sup> Party	Text Properties	I, D
TEW1	TV4	3 <sup>rd</sup> 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

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

Table B.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 are located within the program, and how each is to be certified. One column is provided to mark whether each of the requirements has been tested as a manner 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 C: Development and Deliverables Schedule*

This schedule is presented as a list of dates and events. Where indicated, specific group members are responsible for the deliverable or milestone. We will further subdivide these events and assign individual portions to team members as the requirements for these deliverables become clearer.

Table C.1: Development and Deliverables Schedule for CSPN GUI Project, Team Mellow Yellow

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

## APPENDIX D: Coding Standards

Appendix D provides the format that the Mellow Yellow team will use when writing code. This format shall define standards with the goal of improving consistency, readability and ease of code walkthroughs.

### D.1 Coding Format and Description

In order to increase the readability of our code and make it easier for team members to follow each other's progress, we will follow a standard coding style. We will use C++ programming language. Each function will be formatted and documented as follows:

```
Func_val func (...argument(s)...)
{
    //*****//
    Name of modifier:
    Create Date:
    Update Date:
    Description:
        Input:
        Output:
        Algorithm:
    //*****//
    variable_declarations          // comments
    ...
    // comments
    statements
    ...
    statements
} // func
```

D.1.1 Below are the descriptions of each of the item specified:

D.1.2 Func\_val - this is the function return value. For example,

D.1.3 int – the function will return integer value.

D.1.4 bool – the function will return boolean value.

D.1.5 void – the function will not return any value.

D.1.6 Argument(s) – this is the function parameter variable, where it has the same type as the variable(s) from the function call. Function call is the statement in the program that call a particular function.

D.1.7 Name of modifier – this is the person who modifies the program.

D.1.8 Created date – this is the date when the program is first written.

D.1.9 Update date – the date when the program is last modified.

D.1.10 Description – description contains the input, output and the algorithm of the program.

D.1.11 Input – this is where the type of or the appropriate program input is specified. There shall be an example of the input that should be accepted by the function. The input might be from the function call or the user input.

- D.1.12 Output – this is where the type of or the appropriate program output is specified. The output might go to another function or might be visible to the user as a message. There shall be an example of the output that should be produced by the function.
- D.1.13 Algorithm – the flow of the function will be described in plain English in the algorithm section.
- D.1.13 variable\_declarations – each variable that will be used only in the function will be declared. Each of the variable shall be documented, so each team member will know what it means. Although C++ allow the programmer to declare the variable in the middle of the program, we will constantly declare all of the variables above all of the statements so that it is easy to find.
- D.1.14 statement – the instruction lines.
- D.1.15 comment – any statement that is to describe or to comment the program, so that the other team members will understand what the function does. In C++, the comment will begin with “//”.

## *D.2 Specific Programming Rules*

### *D.2.1 Comments*

These are format that will be used for the comments:

- D.2.1.1 Each variable will be commented. The comment will be at the side of the variable.
- D.2.1.2 For the statements, the comment will be at the top of the particular statement. This way, other group members will get the idea of what the statement is doing. Not all statements will have a comment.
- D.2.1.3 The end of each function with the function name, so that it is distinguishable.
- D.2.1.4 The end of each loop and condition statement, such as while and if-then statements that need the open and close bracket ({ and } for C++), will be commented. The example is shown in figure C.1.

```
For ( count = 0; count < 10; count++) {  
    cout << "Mellow Yellow is happy" << endl;  
    cout << "CS422 is fun!!" << endl;  
    jump += high;  
} // for
```

Figure D.1 This is the example to show the comment at the close bracket.

```

//*****//
// Name of modifier: //
//
// Create Date: //
// Update Date: //
// Description: //
//     Input: //
//     Output: //
//     Algorithm: //
//*****//

```

Figure D.2 This is the example of the border that will be used for the program description.

The description will be within the first line of comments. The first character after the ‘//’ sign will be start with two spaces. This is for the description clarity and neatness. The example is shown in figure C.2.

### *D.3 The Code and the Indentation*

- D.3.1 There will be indentation for each nested statement. C++ will indent the nested statement automatically.
- D.3.2 For the statement that is not indented automatically, we will use four spaces to the left of the first character in the above statement.
- D.3.3 The identifier and the function name in the code will not be too long but will be meaningful enough to make sure it is more understandable. For example, grpMem for group members, str for string, and melYel for Mellow Yellow.

## *APPENDIX E: Hardware Platform Description*

The minimum hardware platform that can be used for our CSPN GUI program is specified below:

- First Generation Intel Pentium Processor or later.
- 8MB RAM (Random Access Memory).
- VGA Videocard with 256 colors – for image display and visual output.
- 16 bits soundcard – for user interface sound. For example, error message sound.

## APPENDIX F: Special Purpose Items

This section elaborates upon the external items to the CSPN-CSPL GUI that were introduced in the context diagram.

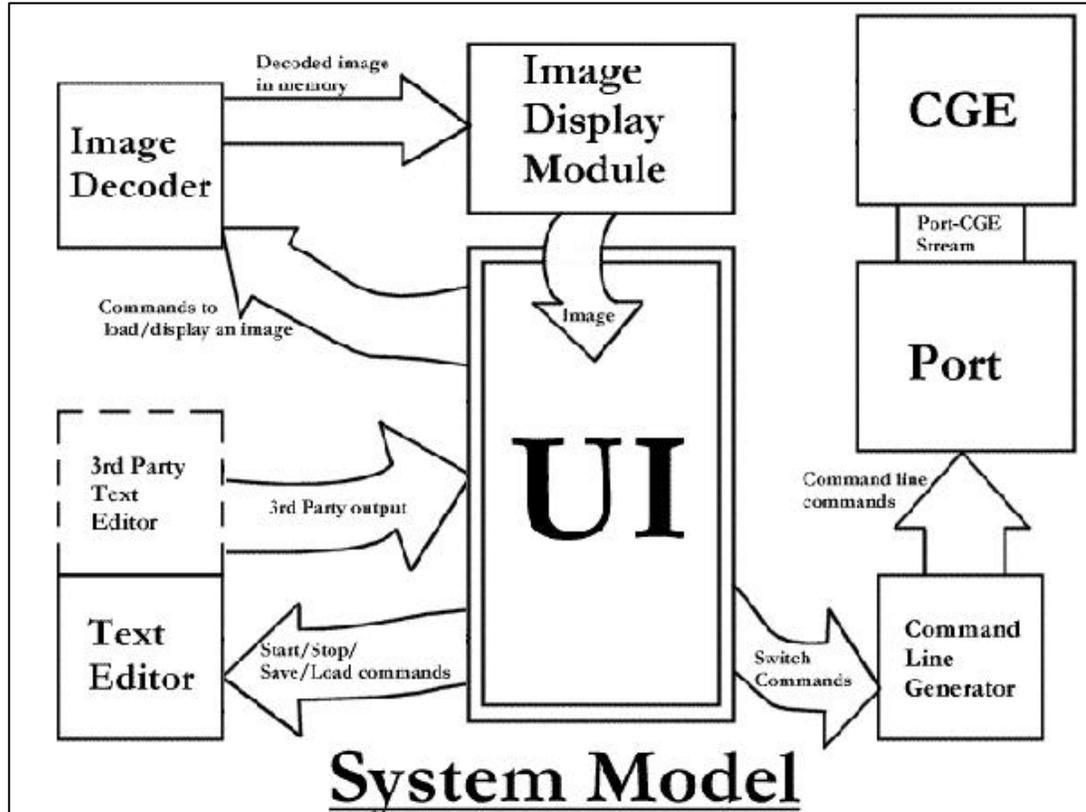


Figure F.1 Entity-Relation Diagram for the external items in CSPN

The CSPN-CSPL GUI allows the user to interact with the CSPN system without the user needs to enter the command line. GUI will provide interfaces for the user in the forms of icons and pull down menus. GUI will interpret the user input and send the interpretation to port where the command will be executed. Then, GUI will interpret the output to the user. They will be in the forms of visual output, sound, error messages and process status box. For example, if there is an error in the input, port will give the error message to the GUI. It indicates that the command line cannot be executed, because there is an error with the input from the user. GUI will tell the user about the input error with an error message and a sound. The process of interaction will continue until the user stops running the system. The interaction is shown in figure F.1.