

**REVISION****REV****DESCRIPTION****DATE**

1.1

Initial Revision

06.10.2005

DigART**DIGITAL CIRCUIT SIMULATOR PROJECT****Software Requirements Analysis Report****Number of Pages****Doc****Appendixes**

12

A	B	C	D	E	F	G	H
–	–	–	–	–	–	–	–



STARSOFT
Middle East Technical University
06531 ODTÜ, Ankara-TÜRKİYE

Controlled Folder:

TBD

A4

1

PREPARED BY

NAME	TITLE/ROLE	REV	SIGNATURE
Elif SAYGI	Quality Engineer	1.1	
Halit DEVELIOGLU	Project Manager	1.1	
Serhat KOYULMUS	Software Developer	1.1	
Seniz SOZER	Configuration Manager	1.1	
Volkan ICEL	Software Developer	1.1	

CONTRIBUTIONS

NAME	TITLE/ROLE	REV	SIGNATURE
Serkan BAL	Advisor	1.1	

CHECKED BY

NAME	TITLE/ROLE	REV	SIGNATURE
Serkan BAL	Advisor	1.1	
Elif SAYGI	Quality Engineer	1.1	
Volkan ICEL	Software Developer	1.1	

TABLE OF CONTENTS

<u>Paragraph No</u>	<u>Page</u>
1. PROJECT DESCRIPTION	5
1.1. BACKGROUND	5
1.2. PROJECT DEFINITION	5
1.3. LITERATURE SURVEY	6
2. THE PROCESS.....	7
2.1. TEAM ORGANIZATION.....	7
2.2. PROCESS MODEL.....	7
2.3. MAJOR CONSTARINTS	7
2.3.1. Time Constraints.....	7
2.3.2. Design Constraints.....	7
2.3.3. Lack of Experience	7
2.4. PROJECT SCHEDULE	8
3. REQUIREMENTS.....	9
3.1. FUNCTIONAL REQUIREMENTS	9
3.1.1. FILE OPERATIONS	9
3.1.2. EDIT OPERATIONS	10
3.1.3. VIEW OPERATIONS	11
3.1.4. INSERT OPERATIONS.....	12
3.1.5. SIMULATION OPERATIONS.....	13
3.1.6. SCRIPT OPERATIONS.....	14
3.1.7. CIRCUIT DISPLAY OPERATIONS	15
3.1.8. TOOLS	15
3.1.8.1. Custom Circuit Element Creation Tool.....	15
3.1.8.2. Ease Access to Circuit Elements Tool	16
3.1.9. HELP	17
3.2. NONFUNCTIONAL REQUIREMENTS	17
3.2.1. HARDWARE REQUIREMENTS.....	17
3.2.2. SOFTWARE REQUIREMENTS	17
4. BEHAVIORAL MODEL	18
4.1. USE CASE DIAGRAMS	18
4.1.1. General View	18
4.1.2. Manage File Operations.....	19
4.1.3. Manage Edit Operations	19
4.1.4. Manage View Operations	20
4.1.5. Manage Insert Operations.....	20
4.1.6. Manage Simulation Operations	21
4.1.7. Manage Script Operations	21
4.1.8. Manage Circuit Display Operations.....	21
4.1.9. Manage Tools	22
4.1.10. Manage Help Operations	22
4.1.11. Overview of the System.....	23

4.2.	ACTIVITY DIAGRAMS	24
4.2.1.	Display/Hide Toolbar	25
4.2.2.	Insert Item.....	25
4.2.3.	Manage Simulation and Glow Mode	26
4.2.4.	Manage Viewport	27
4.2.5.	Manage Script.....	28
5.	APPENDIX.....	29

1. PROJECT DESCRIPTION

1.1. BACKGROUND

The establishment of a standard tool for circuit design is essential for electronics technology. The capability of estimating the prospective fabrication yield according to design peculiarities represents a valuable means for comparing and evaluating design alternatives. Simulation has a high importance in CAD system and simulation tools will have to keep up with the rapidly changing needs of design environments, both in their capabilities and in their style of use. Therefore circuit simulation tools have a market share that can not be underestimated.

1.2. PROJECT DEFINITION

The main aim of the circuit simulation is to analyze the circuit in terms of its responses and to display the circuit in a user friendly form. Verification of circuit performance is the most important expectation of such a simulation tool. The simulation tool is also used with the optimization programs to achieve a better performance quality in certain parts of the circuit. Being able to implement the very complex and big circuits before manufacturing process both saves time and reduce the cost of the project. Simulation requires efficient device modeling to be able to match the full functionality of solid-state physics. All of the components of the digital circuit are defined. The role of circuit simulation is not limited to design and manufacture, it is a very useful tool in education for practical laboratory work. By the simulation programs, students can learn digital components and their responses. Digital circuit and design lectures play an important role in engineering programs in universities. Our researches show that most of the programs in use are designed in professionals' perspective or lecturers' perspective or students' perspective, but none of them is specially designed to satisfy both the students' and lecturer's requirements. All of the team members have taken course Ceng232 and designed circuit in Diglog. It is not that hard for us to specify the requirements of the project from this side. For students the worst part is the bad user interface. It takes too much effort to implement the circuit that they have already drawn on paper. The testing is another problem as the user has to manually test the parts of the circuit, which is again time consuming, they may even skip testing for all inputs. When we consider the lecturer's role, that is grading the program, we see that most simulation tools we analyzed does not meet the requirements of the lecturers, that is testing and analyzing the homeworks of the students. Regarding those we will try to develop the program to be useful for both sides. We will give too much care to the user interface and script support will be provided. The program will be Diglog compatible, since for educational purposes it is the most commonly used program in Universities like Berkeley, Dallas, Metu.

DigArt Project is implementation of a graphical tool which is used for editing digital electronics circuits and executing them. Graphical User Interface of DigArt enables users to design and edit digital circuits easily. Validity of the designed circuit is checked and its operation is tested. All digital circuit elements will be ready to use in our DigArt project.

1.3. LITERATURE SURVEY

During our research on digital circuit simulation tools such as DigSim, DLSim. We have come up with different features and listed them in a matrix in the appendix section.

After conducting a literature survey and from the meetings with our assistant and by the experiences of the team members we have concluded our project features as:

- Full Cut, Copy and Paste of any group of circuit elements.
- Automatically detects signal line connections.
- Place your own text notations on the diagram.
- Automatically reroutes lines at right angles when you move symbols.
- Very simple diagram editing: most symbol placement and wire editing operations do not even require a menu selection.
- Special power and ground symbols automatically create connections and generate the correct simulation values.
- Pop-up menus give you direct access to common commands from anywhere on the diagram
- Easy to use point-and-click user interface.
- Create hyperlinks to move about in circuit or load other files.
- Diglog-like save file format.
- Control through the GUI.
- Suitable for simulation at gate or register transfer level.
- Single-step and clock-step simulator control.
- Basic gates (AND, OR, etc.)
- Alu components (Adders, shifters, multipliers)
- Memory elements
- Hierarchical design through user defined modules
- Script Support
- Diglog Compatibility

2. THE PROCESS

2.1. TEAM ORGANIZATION

When we look at the group members we see that all of us have same background knowledge and we are in the same experience level in software development. All members will participate in design and implementation. Decisions on problems and approaches will be made by group consensus. We think the Democratic Decentralized team structure best fits to our group which does not have an assigned leader. There will be task leaders instead according to our specializations during design. The roles have been assigned to group members at the proposal report. We have chosen a group leader only for organization purposes. The communication is horizontal among us. Each group member is in communication with others and has to inform about his part the changes he/she made to all the group members.

2.2. PROCESS MODEL

The project has 3 months design and 4 months implementation phase. But we think that we should start coding in design phase also and provide prototypes for later improvements. We choose an object oriented approach in the project which gives the chance for reusability. We have to test each phase also in order to avoid possible errors. We will mix prototyping and linear sequential methods suitable for our project.

2.3. MAJOR CONSTRAINTS

2.3.1. Time Constraints

We have 7 months time to complete the design and implementation phases. We want to enhance the properties of our project. We brainstorm on those enhancements but since we are also fourth year students some of our ideas may not be implemented due to the time limitation and the courses we have. Time management will be our biggest concern.

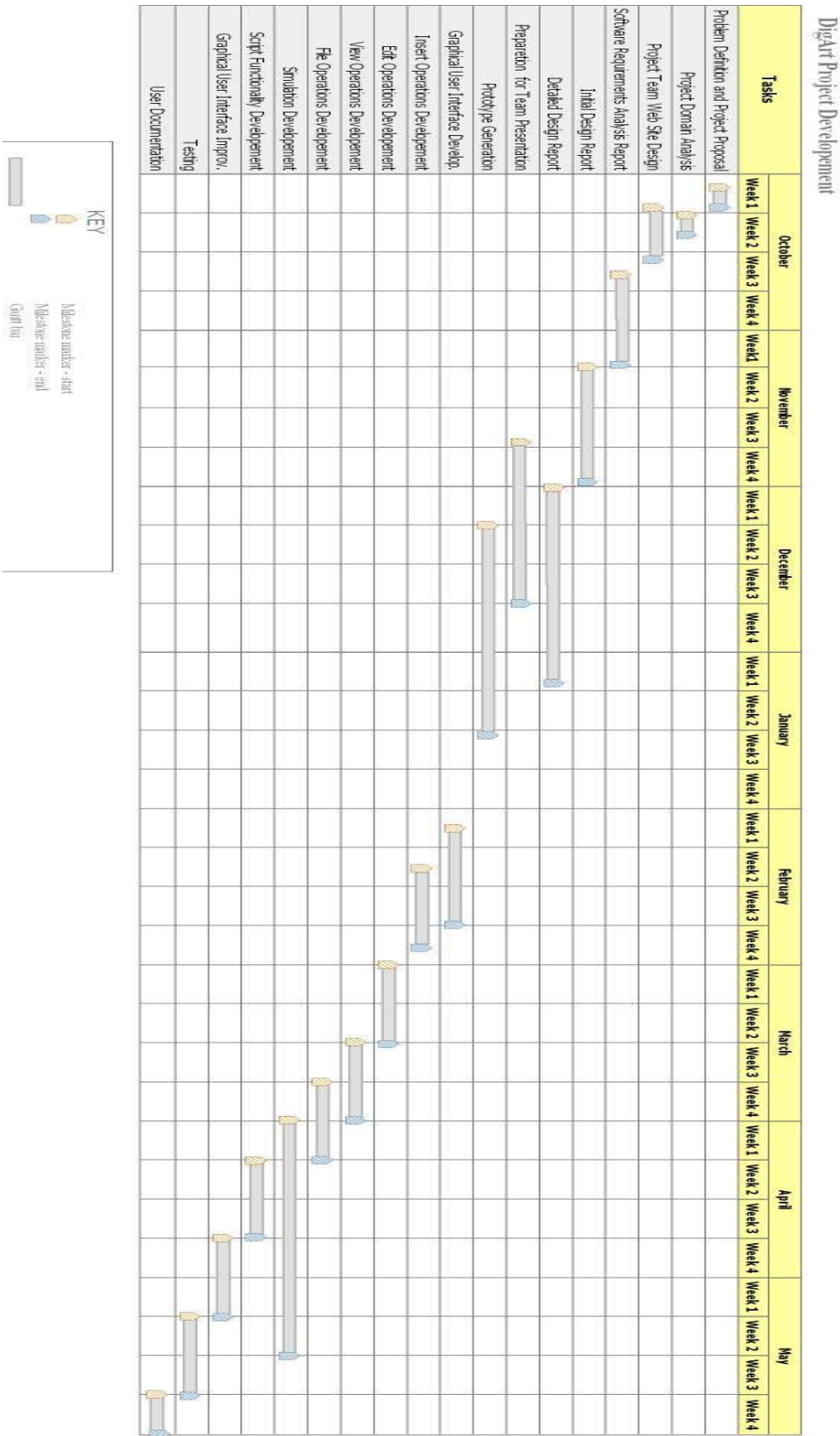
2.3.2. Design Constraints

Java will be used as programming language so that we can use the object oriented concepts and reusability will be achieved. We will use JHDL. The group members will learn about Java and JHDL.

2.3.3. Lack of Experience

None of us have been in a software project and the group members have not designed a graphical software project before. The course Ceng477 will enhance our abilities and knowledge on graphic applications.

2.4. PROJECT SCHEDULE



3. REQUIREMENTS

3.1. FUNCTIONAL REQUIREMENTS

3.1.1. FILE OPERATIONS

DA_SRA_01

DigART shall provide the user to select a file in its own file format “.daf” to save current circuit.

DA_SRA_02

DigART shall save the current circuit to the file selected by the user in format “.daf”.

DA_SRA_03

DigART shall provide the user to select a file to export current circuit into DIGLOG file format “.lgf”.

DA_SRA_04

DigART shall export the current circuit to selected file by the user into DIGLOG file format “.lgf”.

DA_SRA_05

DigART shall provide the user to select a file with extension “.daf” to load.

DA_SRA_06

DigART shall load the file with extension “.daf” selected by the user.

DA_SRA_07

DigART shall provide the user to select a DIGLOG file to import.

DA_SRA_08

DigART shall import and load the DIGLOG file selected by the user.

DA_SRA_09

DigART shall print the displayed part of the current circuit.

DA_SRA_10

DigART shall provide the user to open a new circuit window.

DA_SRA_11

DigART shall open a new circuit window by closing the current circuit window.

DA_SRA_12

DigART shall close the current circuit window.

DA_SRA_13

DigART shall provide the user to select a file to save the displayed part of the current window as an image (as “jpeg” format).

DA_SRA_14

DigART shall save the displayed part of the current window as an image (as “jpeg” format) to the selected file by the user.

3.1.2. EDIT OPERATIONS**DA_SRA_15**

DigART shall provide the user to select item(s) of the current circuit.

DA_SRA_16

DigART shall display the selected item(s) by the user by colouring.

DA_SRA_17

DigART shall provide the user to cut the selected item(s) of the current circuit.

DA_SRA_18

DigART shall delete the cut item(s) by the user from the displayed circuit.

DA_SRA_19

DigART shall provide the user to copy the selected item(s) of the current circuit.

DA_SRA_20

DigART shall provide the user to paste the cut or copied item(s) to the current circuit.

DA_SRA_21

DigART shall insert the cut or copied item(s) by the user to the current circuit when the user select to paste.

DA_SRA_22

DigART shall provide the user to delete the selected item(s) of the current circuit.

DA_SRA_23

DigART shall delete the item(s) selected by the user of the current circuit.

DA_SRA_24

DigART shall provide the user to delete all the item(s) of the current circuit.

DA_SRA_25

DigART shall delete all the item(s) of the current circuit.

DA_SRA_26

DigART shall provide the user to undo last user action on the current circuit.

DA_SRA_27

DigART shall undo the last user action on the current circuit.

DA_SRA_28

DigART shall provide the user to redo last undo action on the current circuit.

DA_SRA_29

DigART shall redo the last undo action on the current circuit.

3.1.3. VIEW OPERATIONS

DA_SRA_30

DigART shall provide the user to zoom in on the current circuit.

DA_SRA_31

DigART shall zoom in the current circuit display.

DA_SRA_32

DigART shall provide the user to zoom out on the current circuit.

DA_SRA_33

DigART shall zoom out the current circuit display.

DA_SRA_34

DigART shall provide the user to zoom overview current circuit display to see the entire circuit. (“zoom overview” means the widest area that all the items of the circuit are displayed)

DA_SRA_35

DigART shall zoom overview the current circuit display.

DA_SRA_36

DigART shall provide the user to pan right the view of point on the current circuit display.

DA_SRA_37

DigART shall pan right the view of point on the current circuit display.

DA_SRA_38

DigART shall provide the user to pan left the view of point on the current circuit display.

DA_SRA_39

DigART shall pan left the view of point on the current circuit display.

DA_SRA_40

DigART shall provide the user to pan up the view of point on the current circuit display.

DA_SRA_41

DigART shall pan up the view of point on the current circuit display.

DA_SRA_42

DigART shall provide the user to pan down the view of point on the current circuit display.

DA_SRA_43

DigART shall pan down the view of point on the current circuit display.

DA_SRA_44

DigART shall provide the user to select and hide any displayed toolbar.

DA_SRA_45

DigART shall hide the toolbar selected by the user.

DA_SRA_46

DigART shall provide the user to select and display any hidden toolbar.

DA_SRA_47

DigART shall display the hidden toolbar selected by the user.

3.1.4. INSERT OPERATIONS
DA_SRA_48

DigART shall provide the user to select and insert circuit item to the current circuit.

Note: Available circuit items will be given in Detailed Design Description.

DA_SRA_49

DigART shall insert the selected item by the user to the current circuit.

DA_SRA_50

DigART shall provide the user to insert connector line to the current circuit.

DA_SRA_51

DigART shall insert a connector line to the current circuit.

DA_SRA_52

DigART shall provide the user to insert a text label of which text is entered by the user.

DA_SRA_53

DigART shall display the inserted text label by the user on the current circuit display.

DA_SRA_54

DigART shall provide the user to insert a region rectangle for bounding a part of the circuit.

DA_SRA_55

DigART shall display the region rectangle on the current circuit display.

3.1.5. SIMULATION OPERATIONS
DA_SRA_56

DigART shall provide the user to change the mode of the program to “Simulation Mode ON”.

(In “Simulation Mode ON” mode, the values of all inputs/outputs are computed and displayed.)

DA_SRA_57

DigART shall run in the mode “Simulation Mode ON” if it is selected by the user.

DA_SRA_58

DigART shall provide the user to change the mode of the program to “Simulation Mode OFF”.

(In “Simulation Mode OFF” mode, the values of all inputs/outputs are not computed and displayed.)

DA_SRA_59

DigART shall run in the mode “Simulation Mode OFF” if it is selected by the user.

DA_SRA_60

DigART shall provide the user to select the mode of the program to “Glow Mode ON”.

(In “Glow Mode ON” mode, the values of lines are computed and displayed with coloring.)

DA_SRA_61

DigART shall run in the mode “Glow Mode ON” if it is selected by the user.

DA_SRA_62

DigART shall provide the user to select the mode of the program to “Glow Mode OFF”.

(In “Glow Mode OFF” mode, the values of lines are not computed and displayed with their default colors.)

DA_SRA_63

DigART shall run in the mode “Glow Mode OFF” if it is selected by the user.

DA_SRA_64

DigART shall provide the user to evaluate a set of files with a set of inputs by comparing the actual outputs with expected outputs and see the evaluation results.

DA_SRA_65

DigART shall evaluate a set of files specified by the user with a set of inputs by comparing the actual outputs with expected outputs and display the evaluation results.

DA_SRA_66

DigART shall provide the user to modify the selected circuit item's properties.

DA_SRA_67

DigART shall set the user defined properties of the selected circuit item.

3.1.6. SCRIPT OPERATIONS

DA_SRA_68

DigART shall provide the user to write circuit scripts. (These circuit scripts shall provide the full circuit functionalities that DigART supports)

DA_SRA_69

DigART shall provide the user to select a file with extension “.das” to save the written circuit scripts.

DA_SRA_70

DiagART shall save the written circuit scripts to selected “.das” file by the user.

DA_SRA_71

DigART shall provide the user to open a “.das” file.

DA_SRA_72

DigART shall verify and open the selected “.das” file by the user.

DA_SRA_73

DigART shall provide the user to compile the current circuit script file with extension “.das”.

DA_SRA_74

DigART shall compile the current circuit script file with extension “.das”.

DA_SRA_75

DigART shall display the syntax error(s) of the compiled script file after compilation.

DA_SRA_76

DigART provide the user to run the compiled circuit script file.

DA_SRA_77

DigART shall provide the user to determine the input values for the circuit script.

DA_SRA_78

DigART shall run and display the proper output for the compiled script file selected by the user with the determined input values.

3.1.7. CIRCUIT DISPLAY OPERATIONS
DA_SRA_79

DigART shall display the current circuit schema in a window.

Note: Close, minimize, maximize, move, restore, resize events will be defined for the window that displays the current circuit schema.

DA_SRA_80

DigART shall provide the user to select and translate circuit item(s) on the circuit display.

DA_SRA_81

DigART shall translate circuit item(s) with its connector lines on the circuit display.

DA_SRA_82

DigART shall provide the user to select and rotate circuit item(s) on the circuit display.

Note: The circuit item can be rotated 90 degree left or right for each rotation action.

DA_SRA_83

DigART shall rotate circuit item(s) on the circuit display.

DA_SRA_84

DigART shall provide the user to connect one or more outputs to one or more inputs with connector lines.

DA_SRA_85

DigART shall connect the user selected output(s) to user selected input(s).

3.1.8. TOOLS
3.1.8.1. Custom Circuit Element Creation Tool
DA_SRA_86

DigART shall provide the user to create custom circuit element by integrating core circuit elements. (Custom circuit element is a small full functional circuit that accepts some inputs and generates proper outputs)

DA_SRA_87

DigART shall provide the user to save the custom created circuit elements by the user.

DA_SRA_88

DigART shall save the custom created circuit elements to “.dae” file.

DA_SRA_89

DigART shall load the “.dae” files and list the custom created circuit element.

DA_SRA_90

DigART shall provide the user to insert the custom created circuit item to the current circuit.

DA_SRA_91

DigART shall simulate the proper outputs of the inserted custom created circuit element in the current circuit.

DA_SRA_92

DigART shall provide the user to modify the saved custom created circuit element. (Modification means the changing the internal circuit schema of the custom created circuit element)

DA_SRA_93

DigART shall save the user modification to the custom circuit element to the related “.dae” file.

DA_SRA_94

DigART shall provide the operator delete a custom created circuit element.

DA_SRA_95

DigART shall delete the custom created circuit element selected by the user.

3.1.8.2. Ease Access to Circuit Elements Tool
DA_SRA_96

DigART shall provide the user to add a circuit element to the ease access circuit element list.

DA_SRA_97

DigART shall list the added circuit elements by the user on the ease access circuit element list.

DA_SRA_98

DigART shall provide the user to select and delete a circuit element from the ease access circuit element list.

DA_SRA_99

DigART shall delete selected circuit elements by the user from the ease access circuit element list.

DA_SRA_100

DigART shall provide the operator to insert a circuit element from the ease access circuit list to the current circuit schema.

DA_SRA_101

DigART shall insert the selected circuit element from the ease access circuit list to the current circuit schema.

3.1.9. HELP**DA_SRA_102**

DigART shall provide the user to open a help window.

DA_SRA_103

DigART shall open the help window.

DA_SRA_104

DigART shall provide the user to select a help topic.

DA_SRA_105

DigART shall display the help information about the selected topic by the user.

Note: Help information will consist of information about all circuit elements and how to use the program.

3.2. NONFUNCTIONAL REQUIREMENTS**3.2.1. HARDWARE REQUIREMENTS****DA_SRA_106**

Pentium 233 MHz, 128 MB of RAM, 30 MB of free hard disk space, Super VGA(800x600 resolution) video adapter and monitor, keyboard and mouse or compatible pointing device.

3.2.2. SOFTWARE REQUIREMENTS**DA_SRA_107**

DigART shall run on the Windows and Linux operating systems.

DA_SRA_108

DigART software shall be developed with Java language with object oriented programming approach.

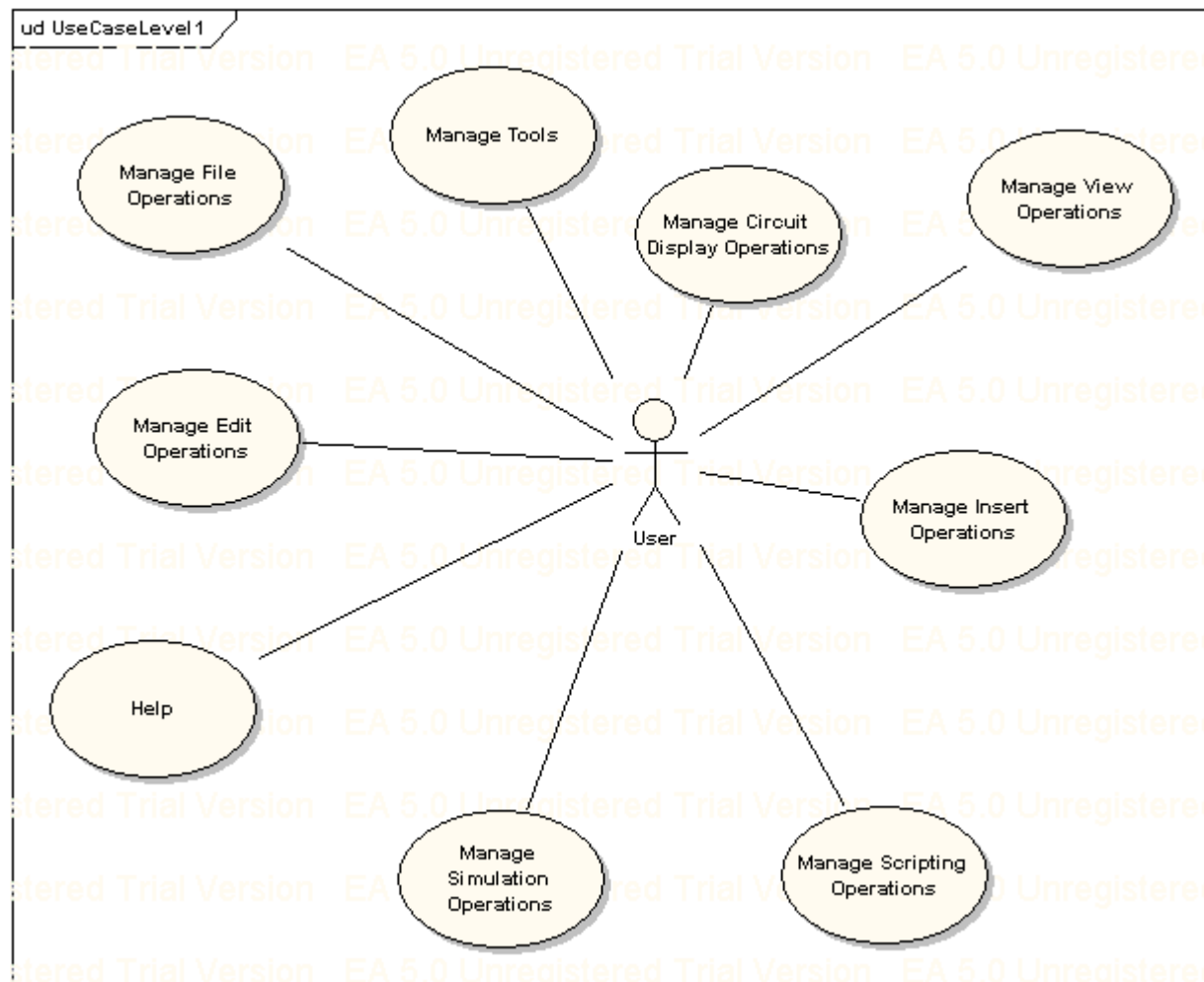
4. BEHAVIORAL MODEL

We have drawn the Use Case Diagrams and Activity Diagrams in order to explain the behaviour of DigART system.

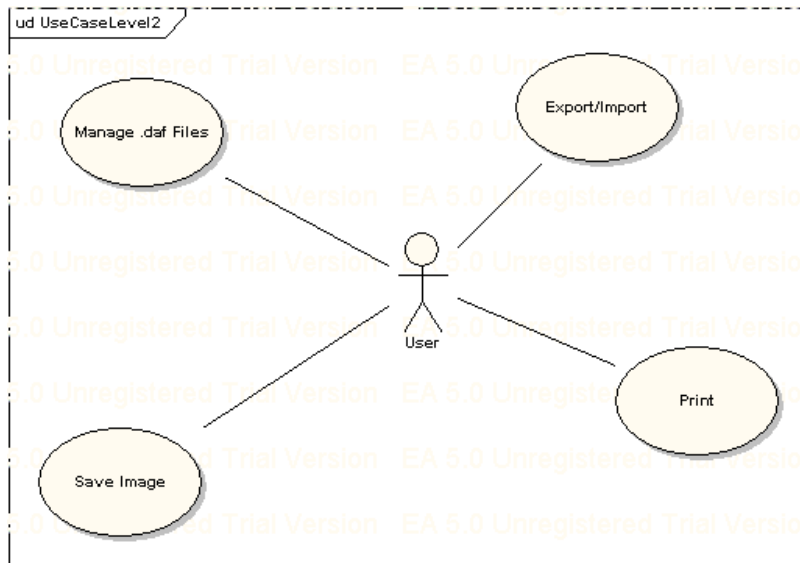
4.1. USE CASE DIAGRAMS

The first diagram which is given under the section 4.1.1 shows the general view of the modules. In the following sections, there are detailed behaviours of each module. In the last section of Use Case Diagrams which is 4.1.11, the overview of the system's behaviour is shown.

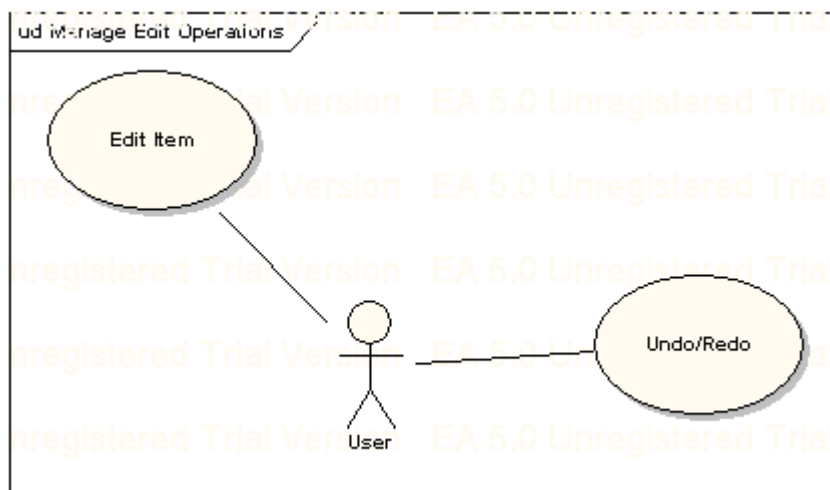
4.1.1. General View



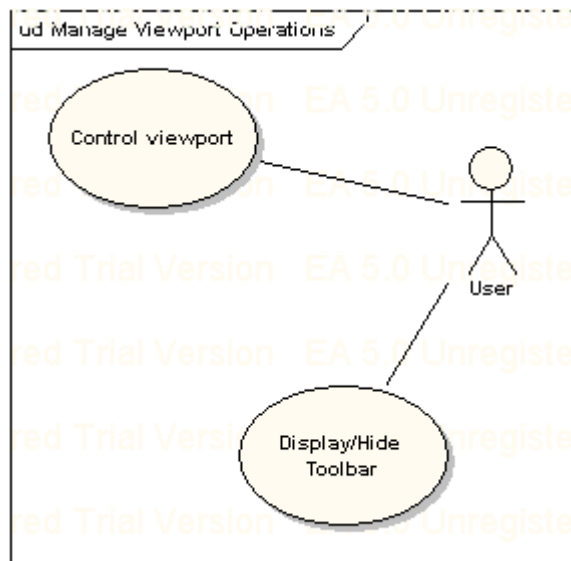
4.1.2. Manage File Operations



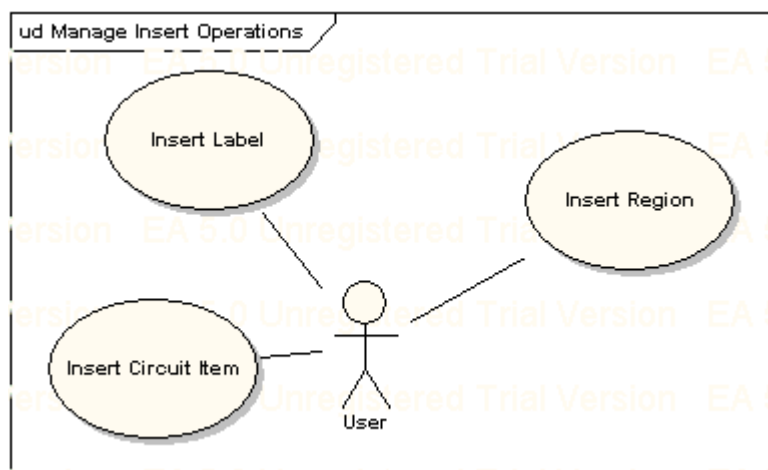
4.1.3. Manage Edit Operations



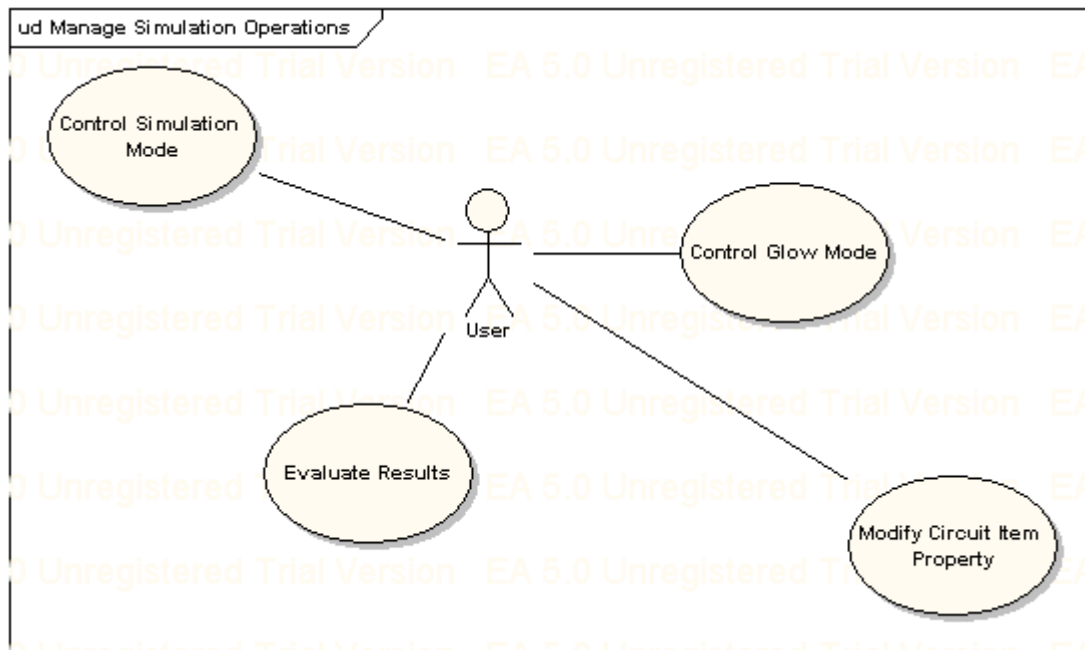
4.1.4. Manage View Operations



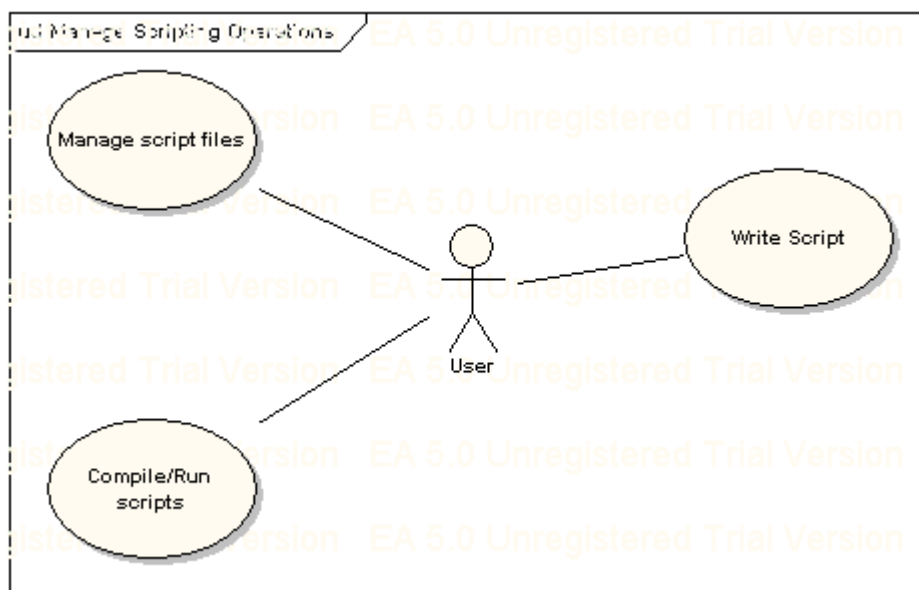
4.1.5. Manage Insert Operations



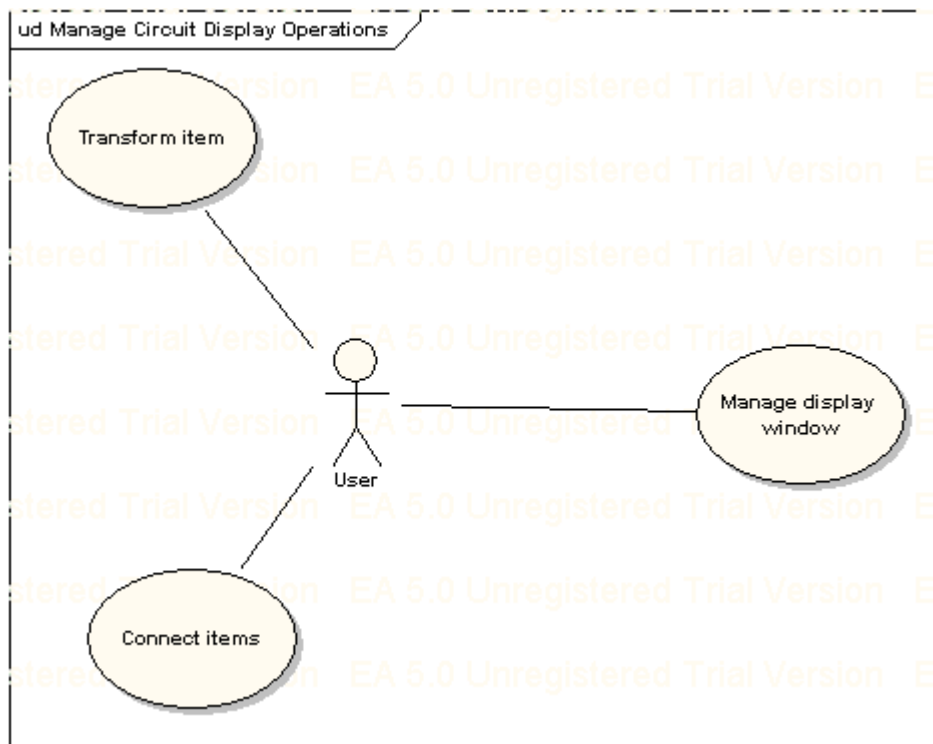
4.1.6. Manage Simulation Operations



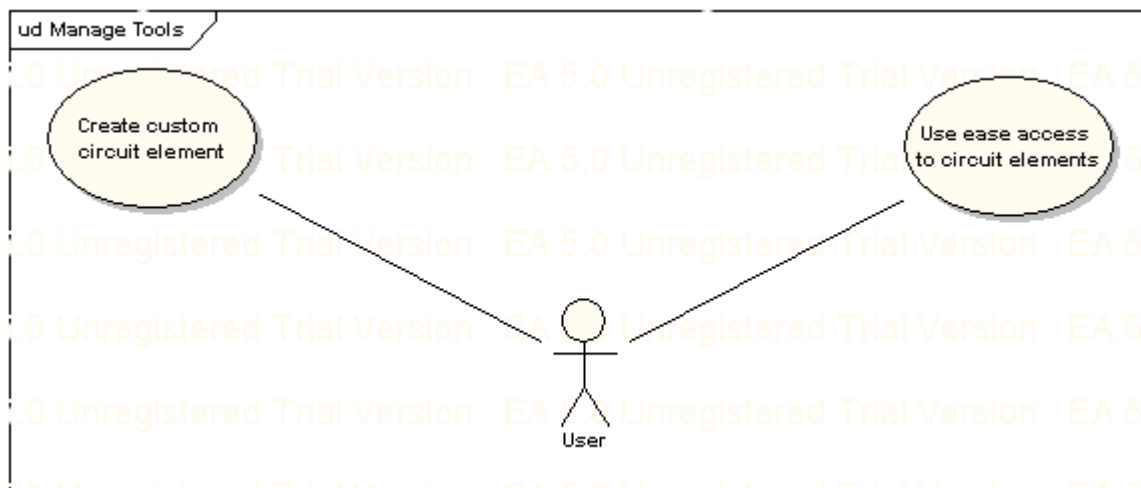
4.1.7. Manage Script Operations



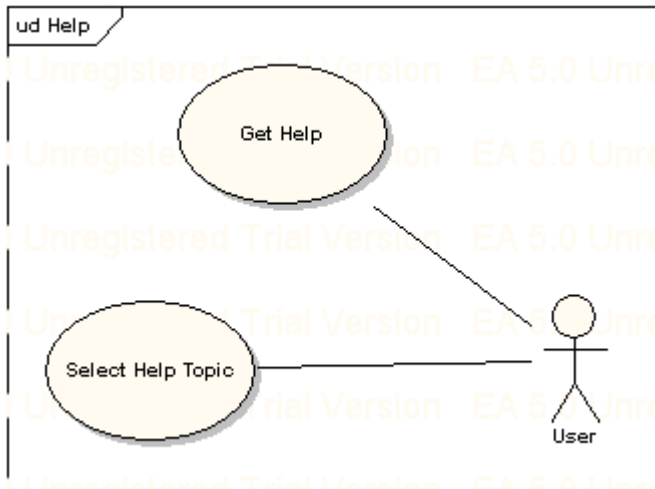
4.1.8. Manage Circuit Display Operations

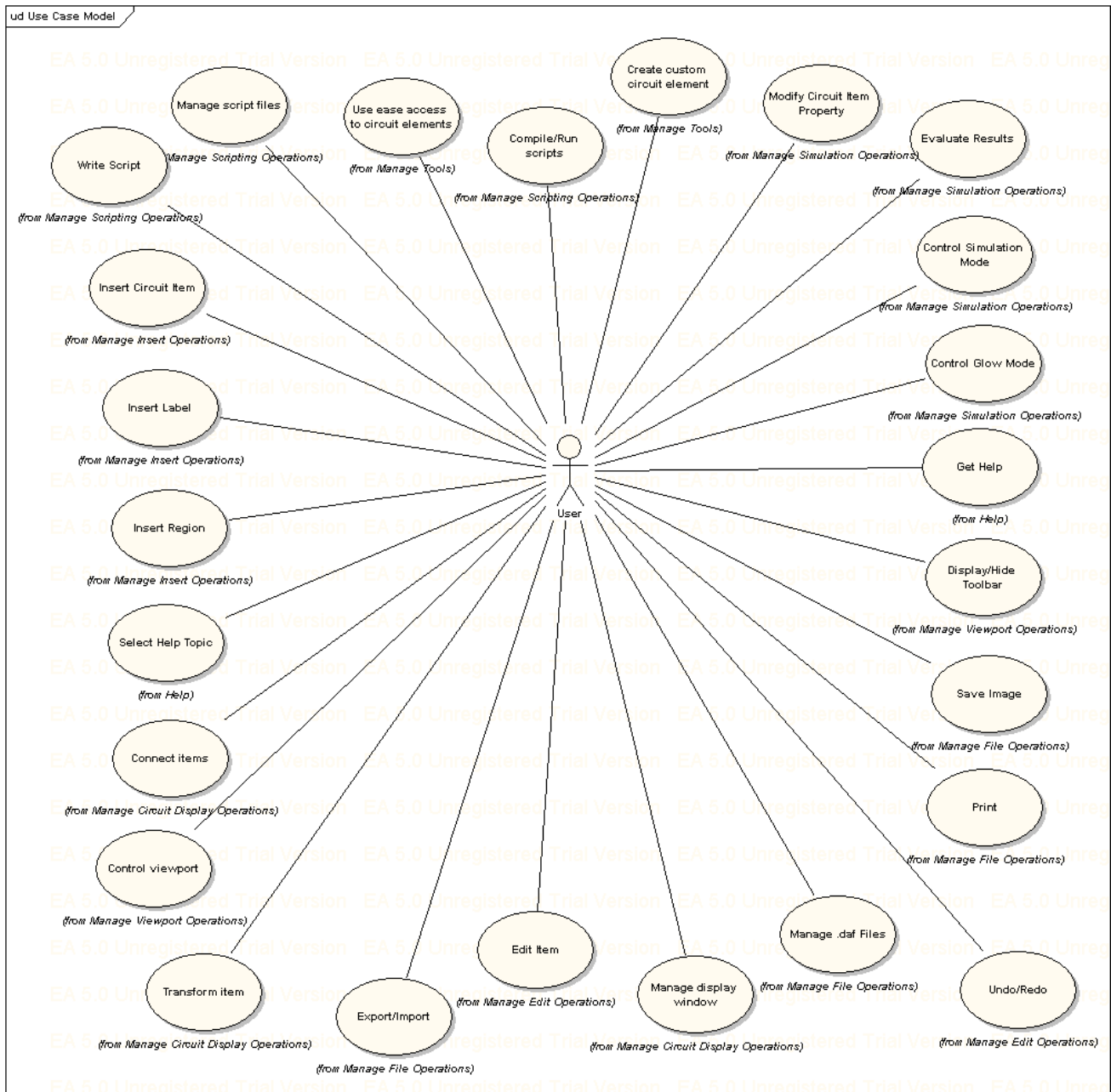


4.1.9. Manage Tools



4.1.10. Manage Help Operations

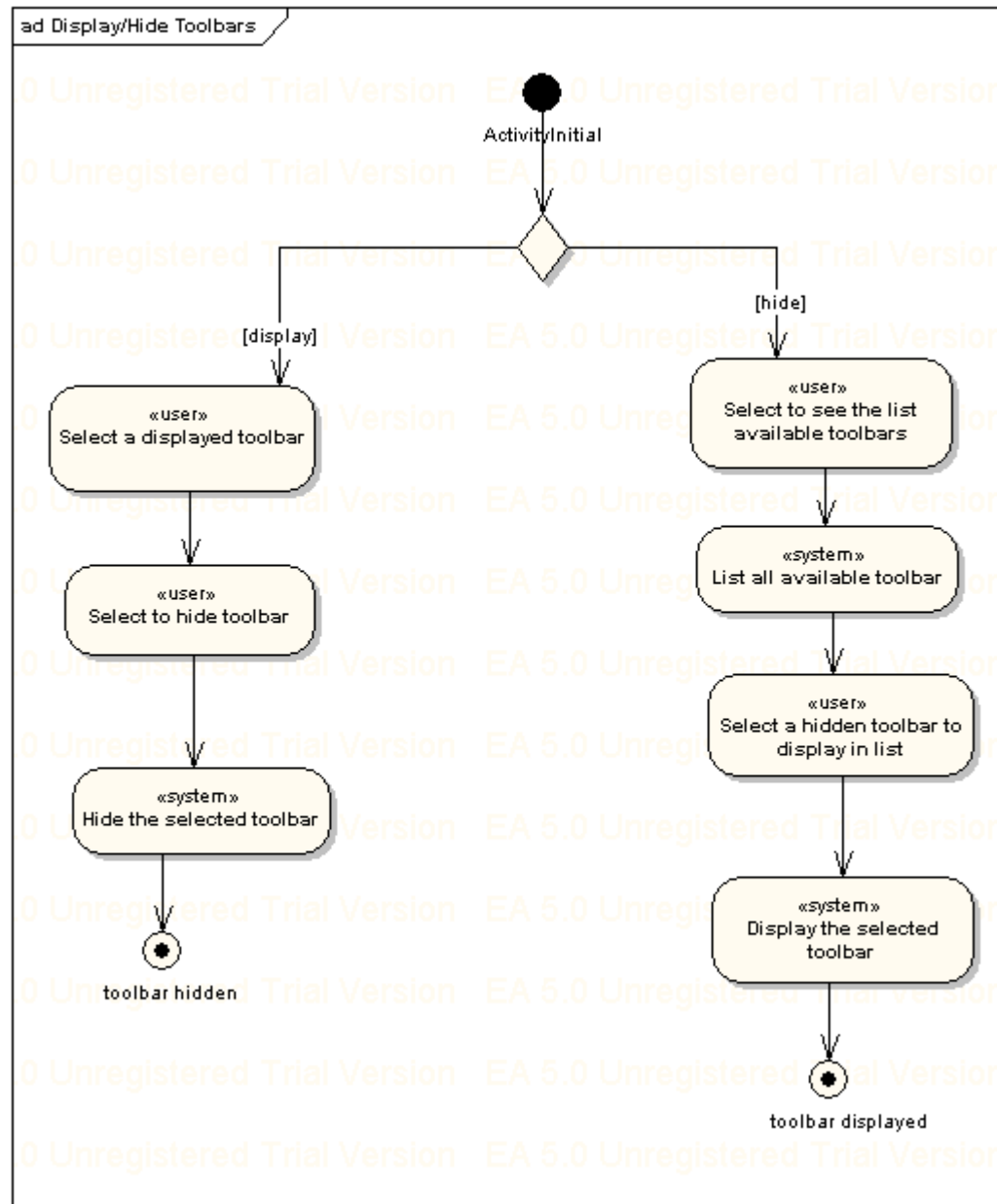




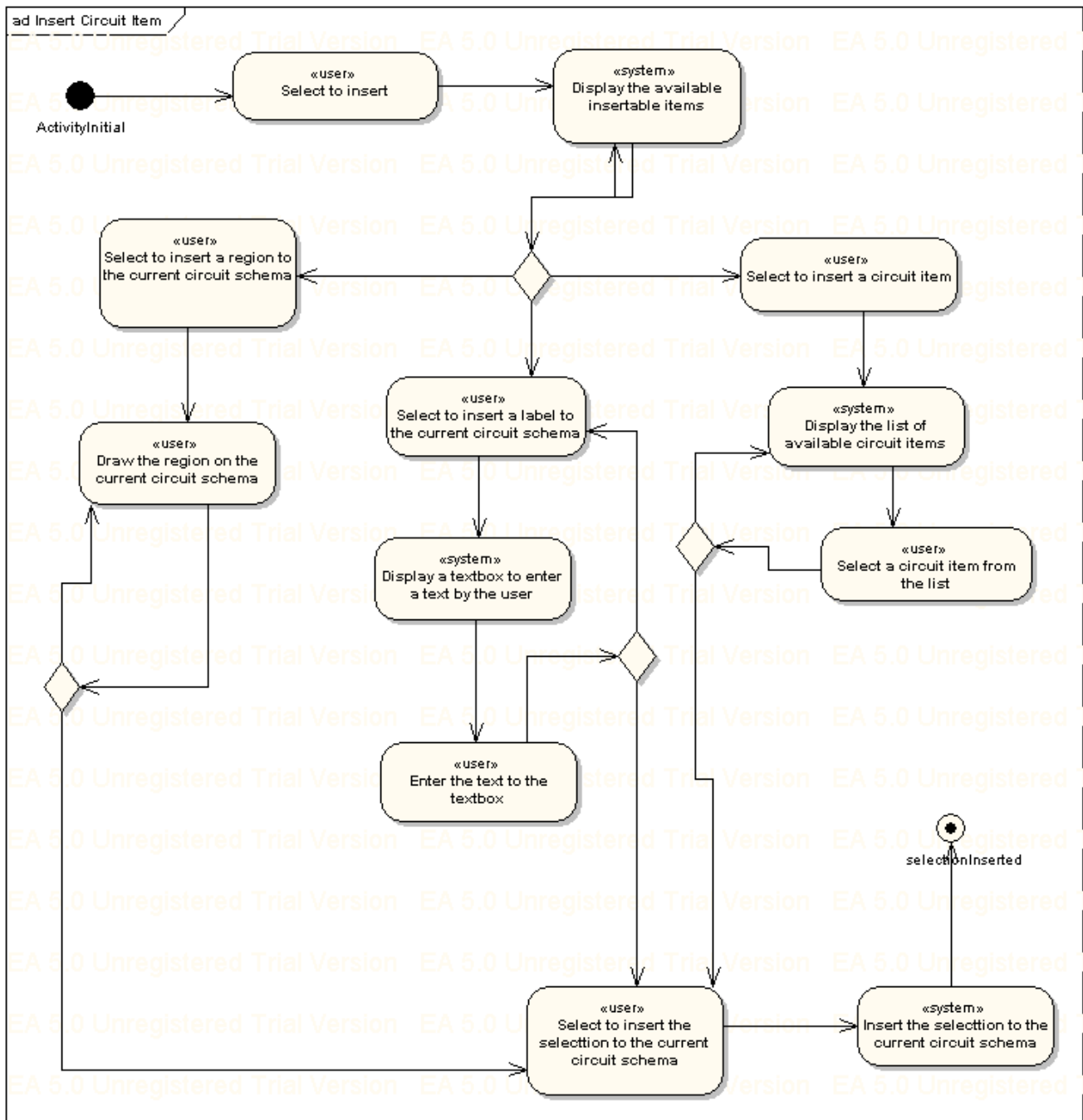
4.2. ACTIVITY DIAGRAMS

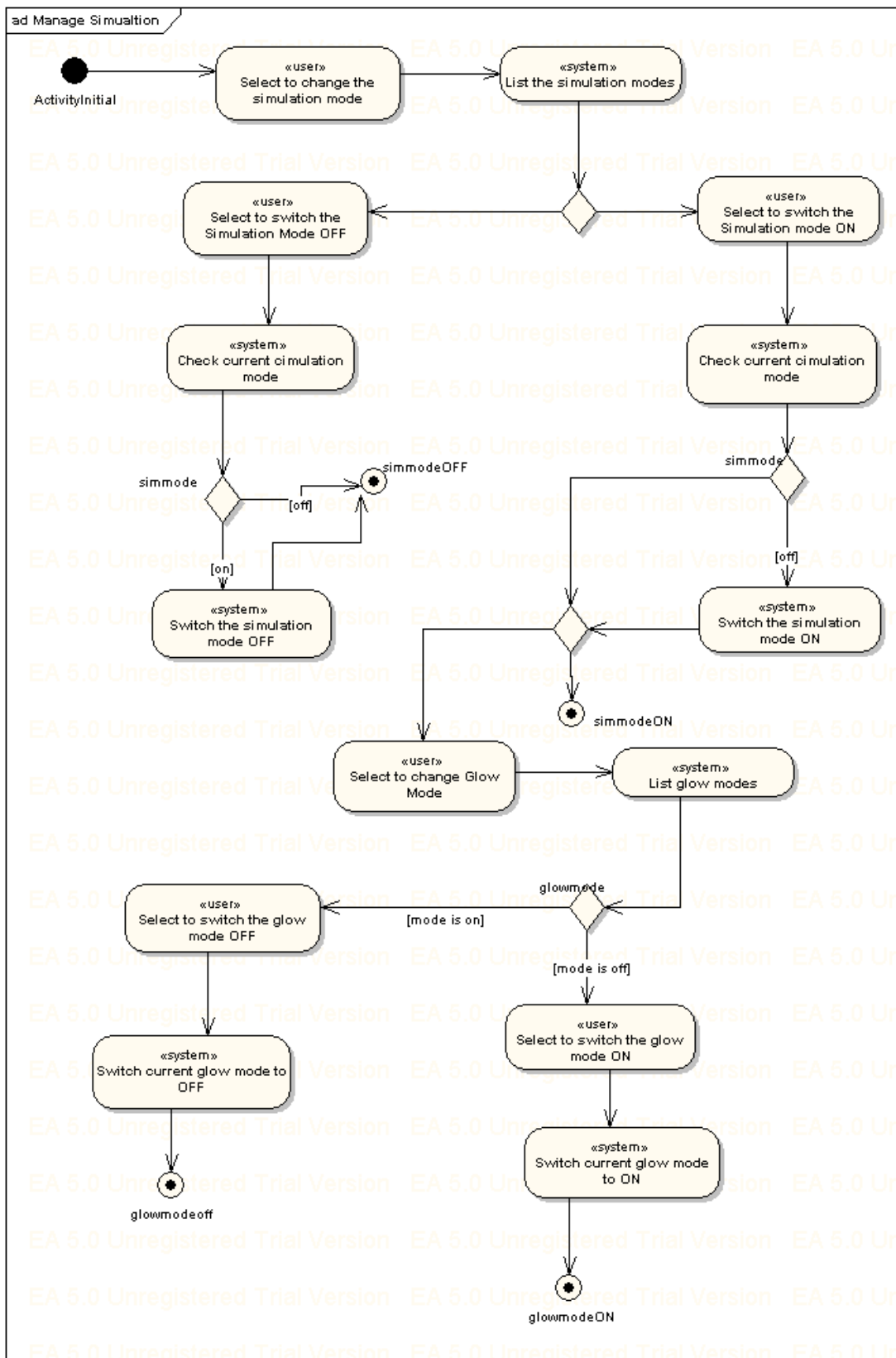
We included the activity diagrams of the critique modules.

4.2.1. Display/Hide Toolbar

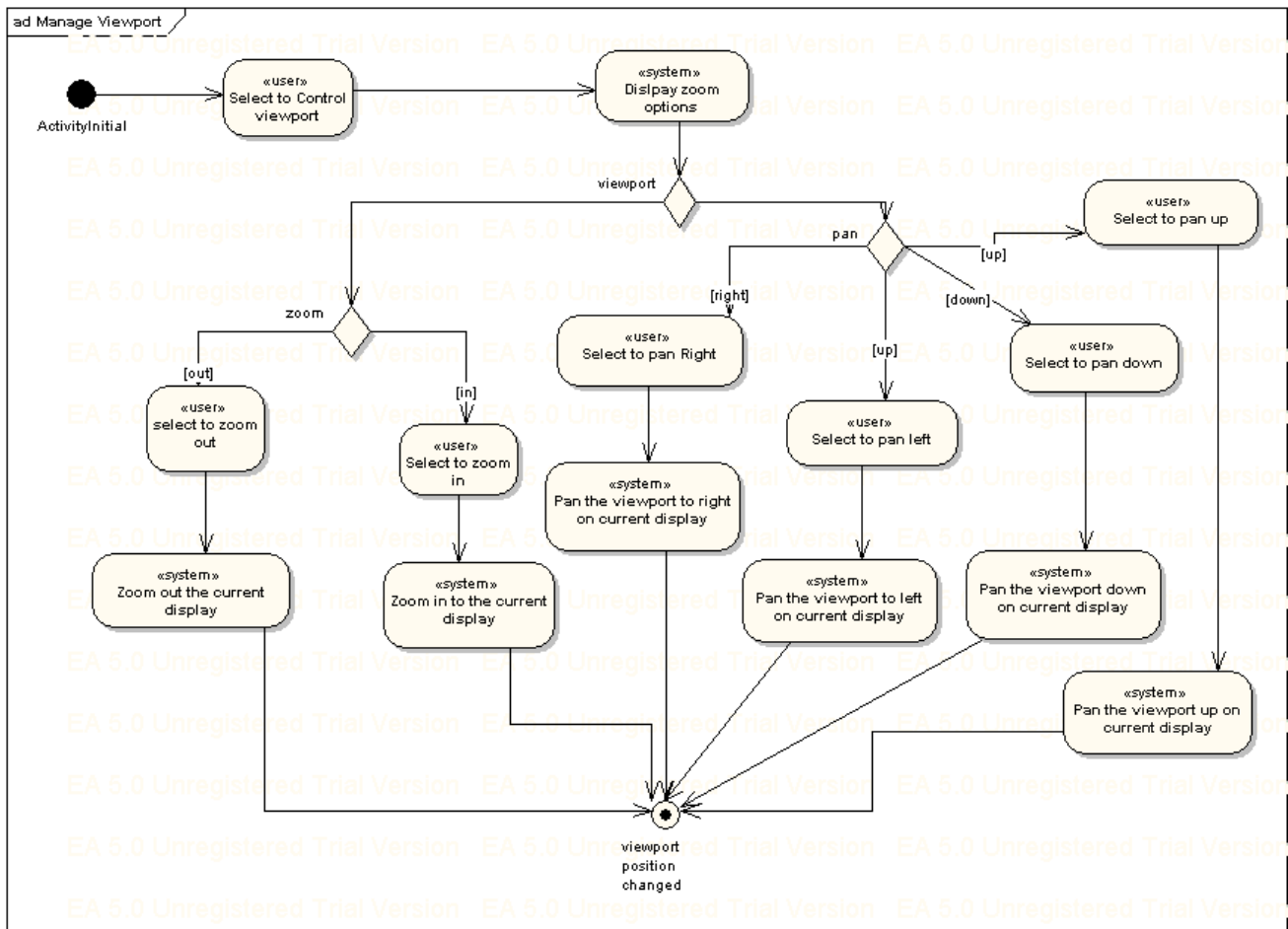


4.2.2. Insert Item

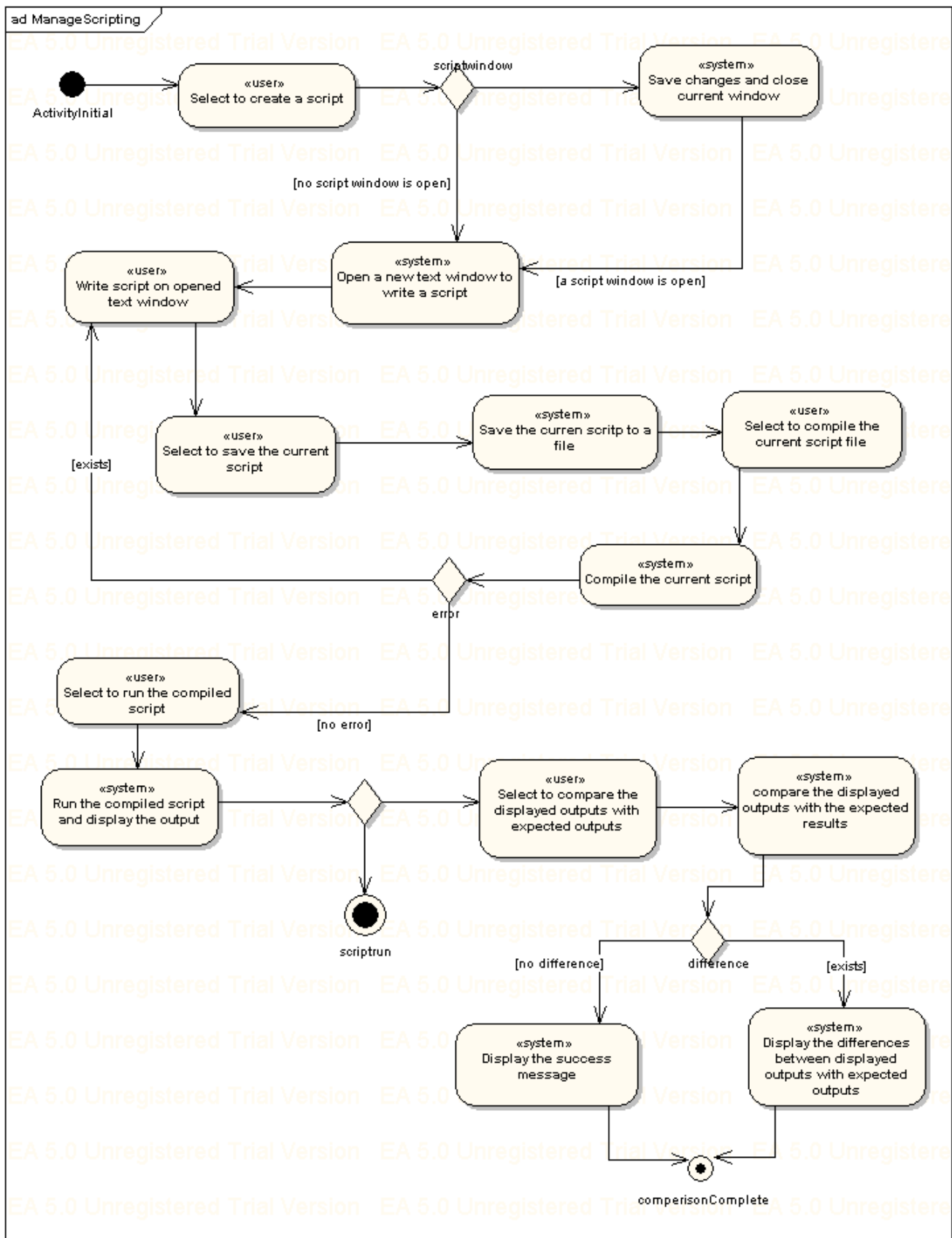




4.2.4. Manage Viewport



4.2.5. Manage Script



Features	TopSPICE	DEEDS	DLSim	Digital Simulator	Chipmunk System	Micro-Cap 8	DigSim	B2Logic	DigArt
Full Cut/Copy/Paste	√	√	√	√		√	√		√
Detection of Signal Line Connection	√				√				√
User Text Notations	√	√	√	√	√	√	√	√	√
Rerouting Lines	√			√		√	√		√
Simple Diagram Editing Without Menu Access	√	√	√	√		√	√	√	√
Pop-up Menus									√
Easy GUI	√	√		√		√	√		√
Creating Hyperlinks									√
Save File Format	√	√		√	√	√	√	√	√
Simulation at Gate or Register Transfer Level	√	√		√	√	√	√	√	√
Single Step and Clock Step Simulator Control	√	√	√	√	√	√	√	√	√
Alu Components (adders, shifters, etc.)				√	√	√	√		√
Memory Elements				√	√	√	√		√
Hierarchical design through user defined modules									√
Animated Simulation							√		√