

FINAL REPORT STUDENT PROJECT

I-MHERE 2012

GraphBT

Integrated Software Development Tool in Behavior Tree

Disusun oleh:

Agung Pratama

Ardi

Chairunnisa Atimas N.

Emerson Chan Simbolon

Ikhsanul Habibie

Nurul Qomariyah



Fakultas Ilmu Komputer Universitas Indonesia

2012

Table of Contents

Table of Contents.....	3
Summary	4
I. Introduction	5
1.1 Background	5
1.2 Project Description.....	5
II. Objectives and Goals.....	8
2.1 Objectives.....	8
2.2 Goals	8
III. Method and Implementation	9
3.1 Method / Stages.....	9
3.2 Planned Project Schedule	9
3.3 Project Schedule Realization.....	10
IV. Project Result.....	11
4.1 Project Result	11
4.2 Achieved Work.....	18
V. Conclusion and Recommendation.....	20
5.1 Conclusion	20
5.2 Recommendation.....	20
VI. Project Member	21
6.1 List of Project Member and Job Description.....	21
6.2 Log.....	22

Summary

One of the most important steps in developing software is requirement phase. In this phase clients will interact with the developer and explain useful information about the desired software they require. However, there are many cases where client can't deliver the requirement in accordance with what they actually wanted and some other requirements are incomplete.

There are several software development methodologies that are widely used. Unfortunately, there are many methodologies that can't handle inconsistencies in the requirement phase, even for some common methodology such as RUP. In order to avoid inconsistency and misleading information, formal software methodology is developed. One of the formal methodologies is Behavior Engineering (BE). This methodology offers building complex systems in formal way but still considerably simple compared to another formal software methodology such as B-method. Behavior Tree (BT) is one the Behavior Modeling Language that consists of components' behaviors that are constructed to represent all business processes of a system.

The target of the project is to develop and integrate a software development tool for creating a Behavior Tree. This tool consists of diagram editor for create and edit Behavior Tree graphically, text editor for textual editing, model checker to verify the created model, debugger tool and execution simulation to debug and simulate the given Behavior Tree, and finally code generator to generate Behavior Tree to an executable code. The resulted product will be delivered as an Eclipse plug-in.

The implementation stage is divided into several steps. The first step is separately developing major features as an Eclipse plug-in needed for the tool, which is BT Model Checker, BT Code Generator, BT Debugger and Execution Simulator, and Diagram and Textual Editor for Behavior Tree. The next step of implementation is integrating the entire produced tool into a single Eclipse plug-in that contains all functionality. During the implementation step, some important supplementary documents such as User Manual and Installation Guide are also provided.

I. Introduction

1.1 Background

There are several steps in developing software. One of the vital steps in developing software in Software Engineering is requirement gathering. In this phase, client will provide the developer with the specification of the required software. In this development stage, most of the clients usually lack knowledge of software development, thus the requirement informed to the developers often derailed from what they actually wanted. Another problem arises when the requirement specified is incomplete.

There are many methodologies that can be used to develop software. One example is prescriptive methodology. This methodology is using artifacts in the software development. Before implementation phase, the artifacts should already fix. Another example is incremental prescriptive methodology. It starts with less detail artifacts, the process prescriptive process repeats with more details artifacts until the client satisfied.

One of the methodologies is Behavior Engineering (BE). BE is proposed to build complex systems formally in an easier way. Behavior Engineering utilizes Behavior Modeling Language (BML) that consists of three views; Behavior Tree (BT), Composition Tree (CT), and Structure Tree (ST). These three views respectively represent complete behavior, composition, and structural integrated views of a system.

Universitas Indonesia, Faculty of Computer Science's Formal Method in Software Engineering Laboratory has contributed in developing Behavior Tree by creating and improving tools such as, debugging tool, BT2SAL translator, source-code generator, etc. However, those tools can't be used to develop a software development using Behavior Tree notations as they weren't integrated yet. The integration of those tools into an integrated development tools makes software development process easier to do. By using SAL Model Checker, user can easily check whether a represented model is consistent or not, User can easily get Java code by using BT Code Generator, and to simulate the execution from BT specification using BT Debugger.

1.2 Project Description

Software development utilizes software methodology to develop software. The purpose of using software methodology is to provides communication between developer and clients about the software that client need, and also to manage the development process. The common software methodology consists of gathering requirement phase, analysis, design, implementation, and testing. The requirement gathered from client sometimes can become complex, contains redundancy, ambiguous and incomplete information. The analysis process will make sure the requirement is complete. The design process contains the plan about how the software is built and matches the requirement. Implementation process contains coding process in order to make sure the design can be implemented into an executable program. The testing process will make sure all the requirement handled well when user operates the software.

Behavior engineering as a new software methodology offers many advantages in developing software. It uses formal approach in designing the software using Behavior Tree (BT). Because of the formal structure of BT, the specification can be model-checked and the code can be generated automatically. By using BT, it will reduce the effort of programmer in implementation phase and only concentrate on creating the BT design. The BT also can be created separately by many people in one time so that each programmer can be more focused on their own work. The BT can also be integrated into Integrated Behavior Tree (IBT) with some rules and refinement.

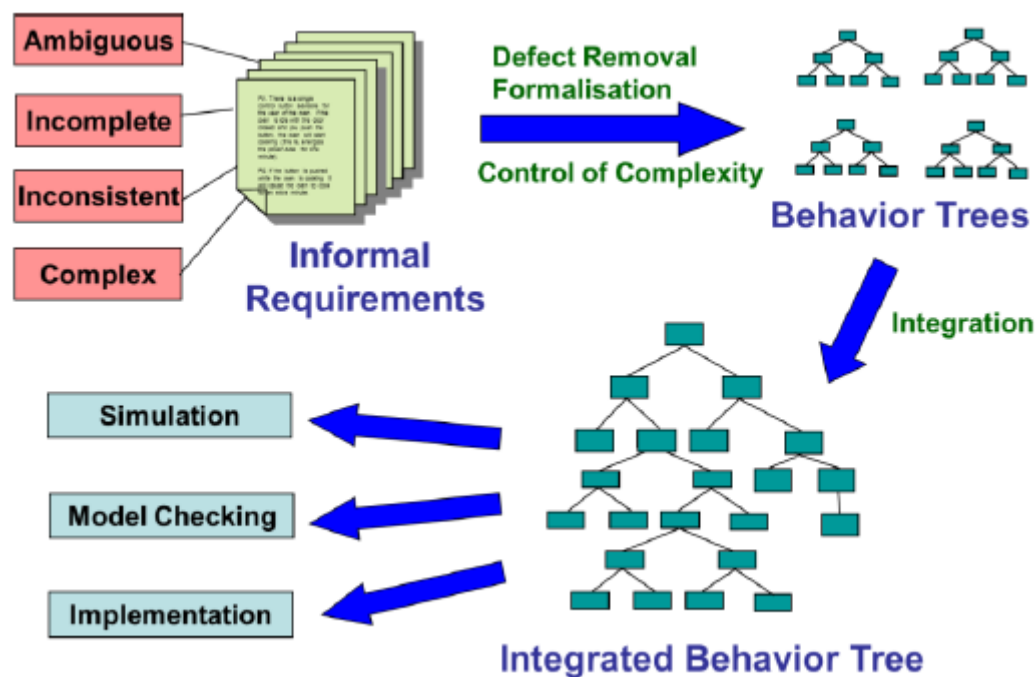


Figure 1. BE development step (ARC Center for Complex System)

FMSE laboratory in Faculty of Computer Science in Universitas Indonesia is currently works on Behavior Engineering (BE). There are some latest research provides tools that can be used to manipulate BT, namely:

1. BT Model-checker
This research was conducted by Niken Listya Pratiwi in 2011. The model-checker is used to check the correctness of BT model.
2. BT Trace
This research was conducted by Ferdiansyah Dolot in 2011. The tool provides simulation of BT execution. Each node will be traced with respect to execution flow.
3. BT code generator
This research was conducted by Emerson C Simbolon in 2012. The tool provides automatic code generator of BT and require an XML file of BT specification.

The project works on a tool to manipulate BT specification into an executable program. The tool also provides mechanism to check the model of BT using SAL and animation trace of the BT. The proposed software development process using the tool is shown in Figure 2.

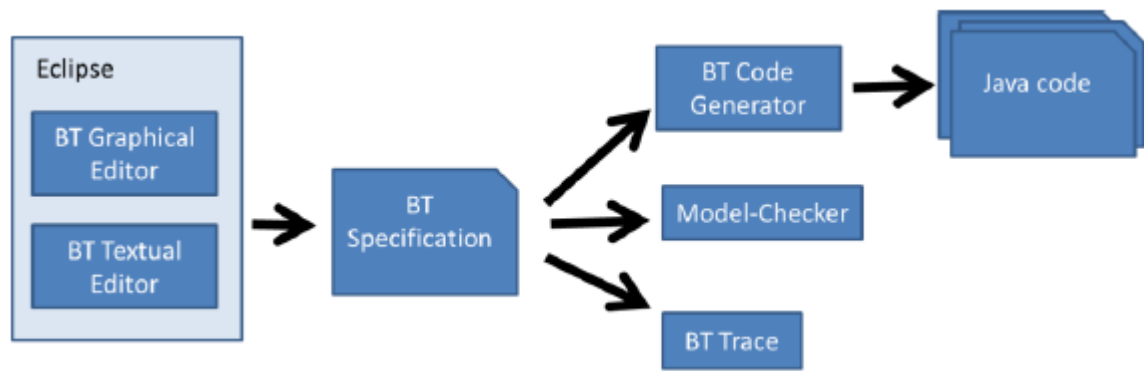


Figure 2. Software Development Process

II. Objectives and Goals

2.1 Objectives

The objectives of this project are:

1. Tool integration for constructing Behavior Tree.
2. Provide easier way to create Behavior Tree by providing graphical editor.

2.2 Goals

The goals that are expected from this project are:

1. Develop integrated BT development tools which will have following features:
 - a. BT Text Editor
Used for editing the BT source code. TextBE is used for this purpose.
 - b. BT Graphics Editor
In this project, we decide to use graphic editor as the main interface. User can insert BT node and the connection among BT node graphically.
 - c. BT Model Checker
Model checker is used to ensure the correctness of a BT design. When the BT is model-checked, there are two possible outcomes: if the BT correct, the tool will generate the code; otherwise, it will display error line along with the option to run the BT simulation. The use of model checker is for advanced user who already knew about formal method. As for aliveness and safeness the user should insert the right node to be checked.
 - d. BT Execution Simulator
Display the simulation of the BT execution. It will mark the error node (if occurred).
 - e. BT Code Generator
Generate executable code from BT specification.
 - f. Update site
As the resulted product from this project is an Eclipse plug-in, an update site is provided so that user can easily download the latest version directly from their Eclipse with ease.
2. Documentation
The tools are also provided with supplementary documents which consist of the following: User manual, installation guide, release note, technical report, and final report.

III. Method and Implementation

3.1 Method / Stages

The project implementation for GraphBT consists of 7 steps, which are:

1. Study and learn to build plug-in in Eclipse environment
2. Analysis and design
3. Implementation
4. Testing design
5. Integration to Eclipse
6. Documentation
7. Finishing

The detailed stages for GraphBT implementation are described in the table below:

No.	Phase	Description
1.	Study and learn to build plug-in in Eclipse environment	In this stage each team member will learn some basic concepts in developing Eclipse plug-in. In addition to that, each member will learn specific tool that will be used for each different tasks.
2.	Analysis and design	In this stage the business process of the tool is analyzed to fulfill the needs of building a system in Behavior Tree efficiently. The process also involved the design of the tool architecture including the utilization of Eclipse Framework
3.	Implementation	This stage is the phase where the previously analyzed design is implemented into an Eclipse plug-in.
4.	Testing design	The testing process is done by examining every implementation artifacts, such as: BT Code Generator, BT Debugger Tool and Simulation and BT Model Checker. The user experience aspect is also being tested so the user can use the tool with minimal effort.
5.	Integration to Eclipse	All of the plug-in parts are integrated so it can be used altogether in a single view (diagram editor).
6.	Documentation	All of the produced plug-ins is supplemented by several documents which has different purpose.
7.	Finishing	Final stage for cleaning any unfinished tasks.

3.2 Planned Project Schedule

Week	Description
1-4	Study and learn to build plug-in in Eclipse Framework Training
3-15	Analysis and Design
5-20	Implementation
10-20	Testing design
15-20	Integration to eclipse
1-20	Documentation (Progress report, technical report, analyze and design documentation, user manual)
20-22	Finishing

3.3 Project Schedule Realization

Week	Description
1-8	Study and learn to build plug-in in Eclipse Framework Training
5-15	Analysis and Design
6-22	Implementation
15-20	Testing design
16-22	Integration to eclipse
1-22	Documentation (Progress report, technical report, analyze and design documentation, user manual)
21-22	Finishing

IV. Project Result

4.1 Project Result

General View

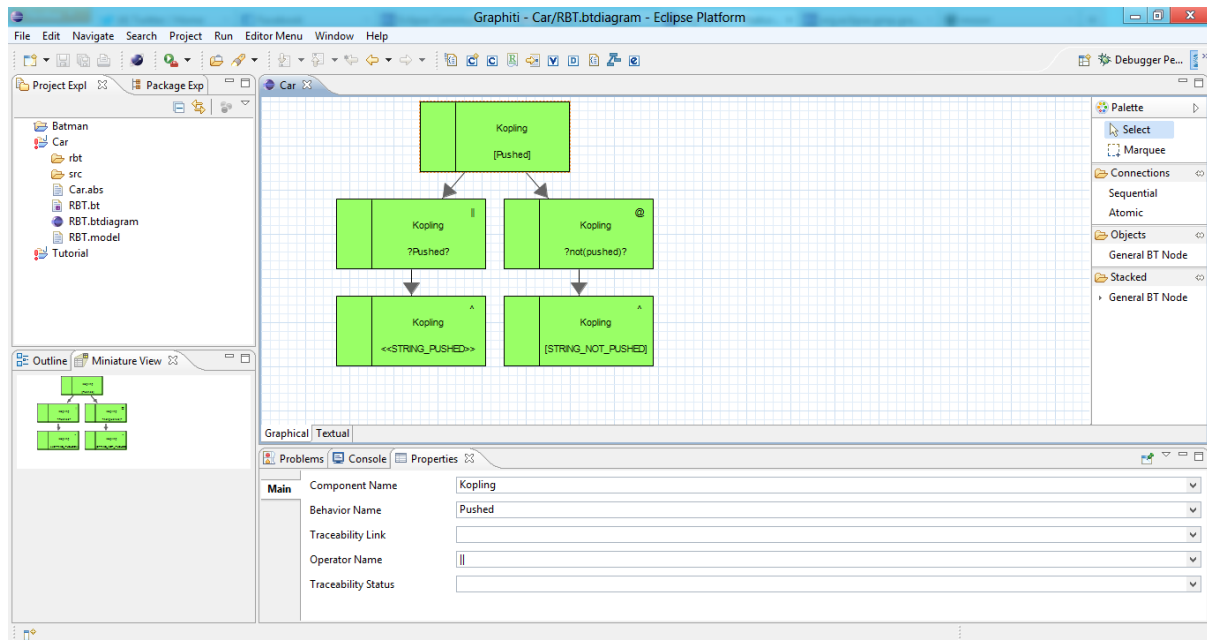


Figure 4.1 General View

Property View for BT Node

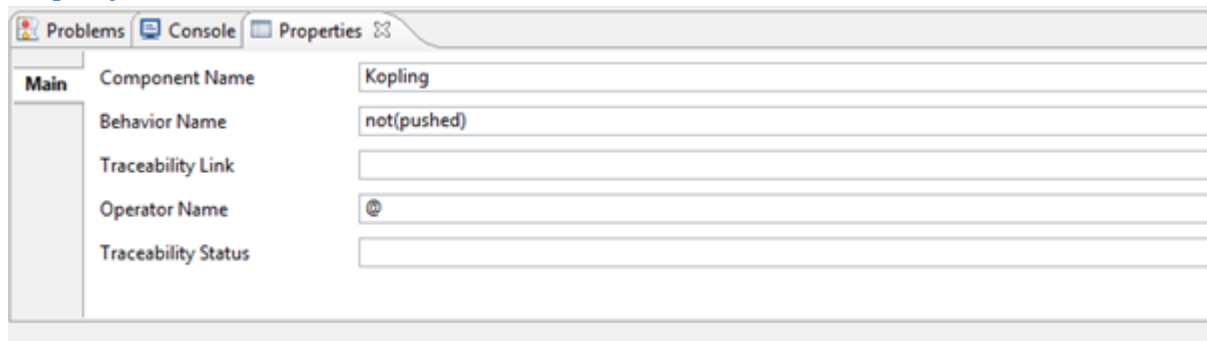


Figure 4.2 Property View for BT Node

Miniature View

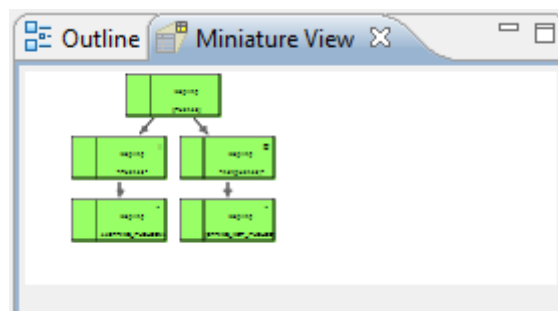
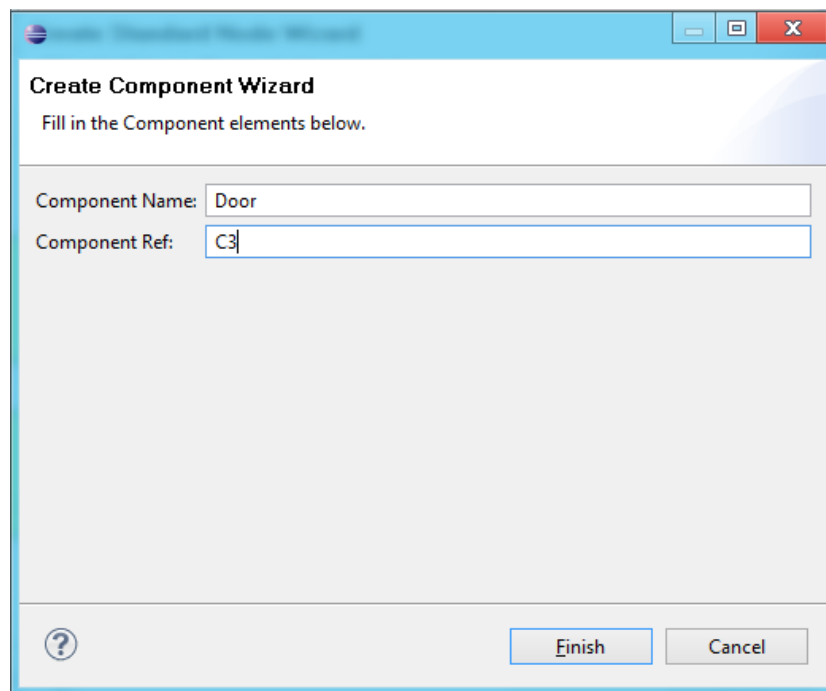


Figure 4.3 Miniature View

Add Component

Specifying a new Behavior Tree component in GraphBT can be performed by using add component feature.

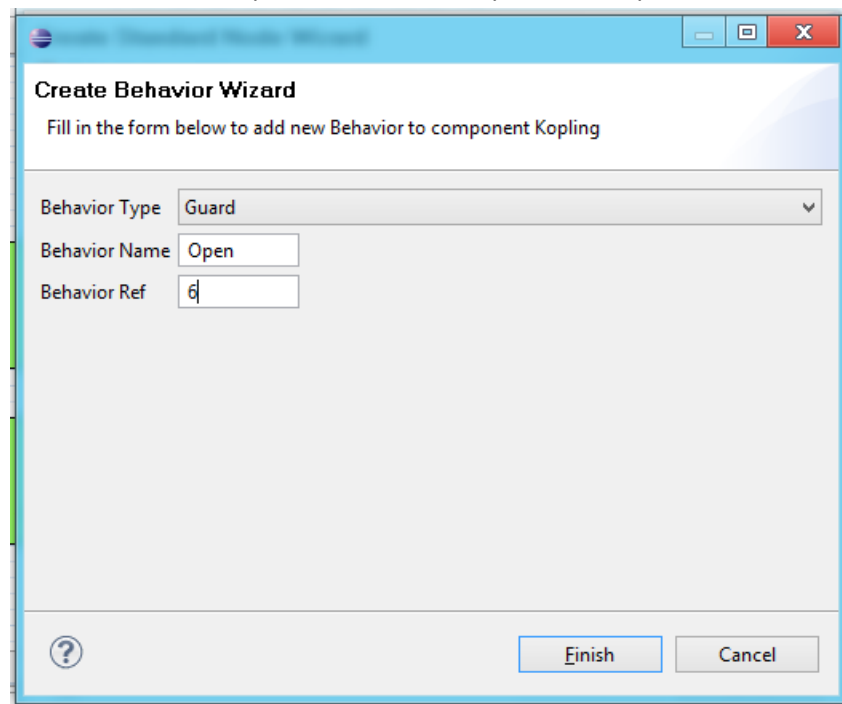


The 'Create Component Wizard' dialog box is shown. It has a title bar with standard window controls. The main area contains two text input fields: 'Component Name' with the value 'Door' and 'Component Ref' with the value 'C3'. Below the inputs is a large empty rectangular area. At the bottom, there is a question mark icon on the left and 'Finish' and 'Cancel' buttons on the right.

Figure 4.4 Create Component Wizard

Add Behavior

Adding new behavior can be accomplished after the respective component has been specified.



The 'Create Behavior Wizard' dialog box is shown. It has a title bar with standard window controls. The main area contains three input fields: 'Behavior Type' is a dropdown menu showing 'Guard', 'Behavior Name' is a text box with 'Open', and 'Behavior Ref' is a text box with 'd'. Below these is a large empty rectangular area. At the bottom, there is a question mark icon on the left and 'Finish' and 'Cancel' buttons on the right.

Figure 4.5 Create Behavior Wizard

Manage Components

BT Components and their respective behaviors can be managed using Manage Component Wizard which can be invoked by pressing Manage Component button in the toolbar.

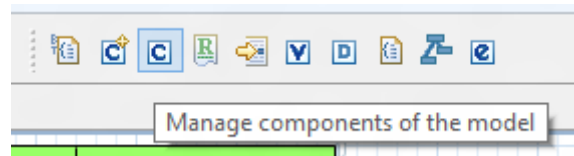


Figure 4.6 Manage Components icon in the toolbar

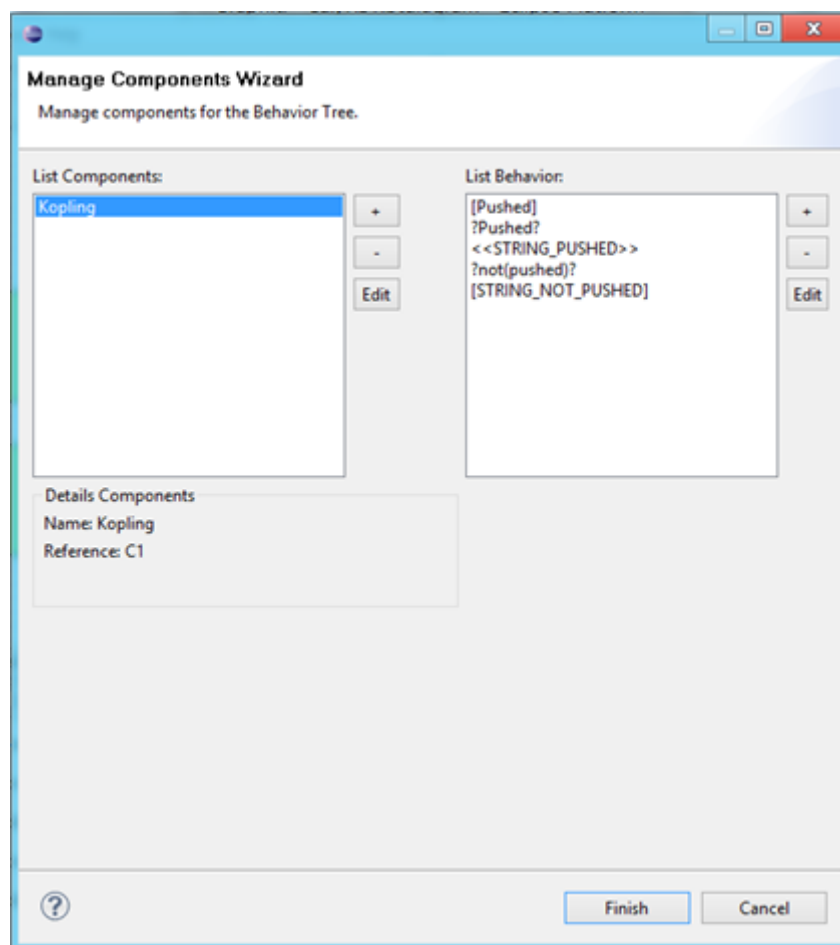
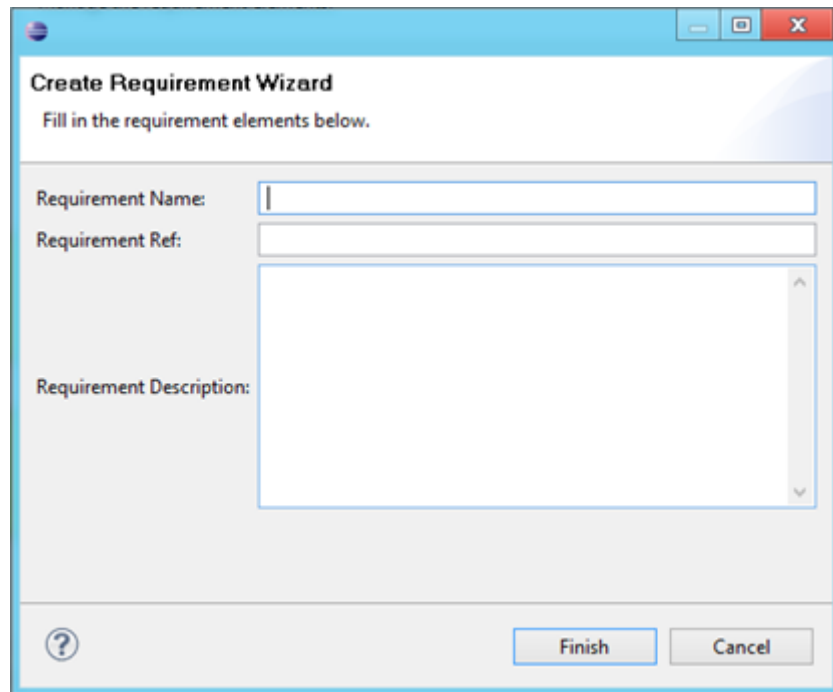


Figure 4.7 Manage Components wizard

Add Requirement



The image shows a 'Create Requirement Wizard' dialog box. It has a title bar with standard Windows window controls. The main area contains three input fields: 'Requirement Name' (a single-line text box), 'Requirement Ref' (a single-line text box), and 'Requirement Description' (a multi-line text area). Below these fields is a question mark icon on the left and 'Finish' and 'Cancel' buttons on the right.

Figure 4.8 Manage Components wizard

Manage Requirement

User can add, modify, and delete requirements using Manage Requirement feature.

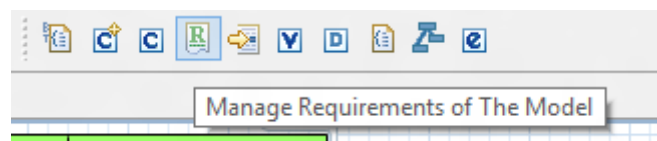


Figure 4.9 Manage Requirements icon in the toolbar

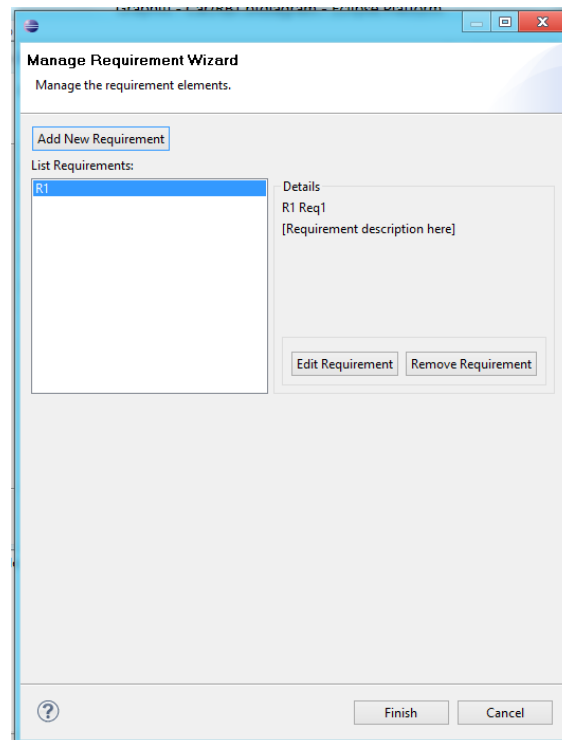


Figure 4.9 Manage Requirement wizard

Edit Requirement

User can edit any created Requirements by selecting a Requirement from selection list and press Edit Requirement button afterward.

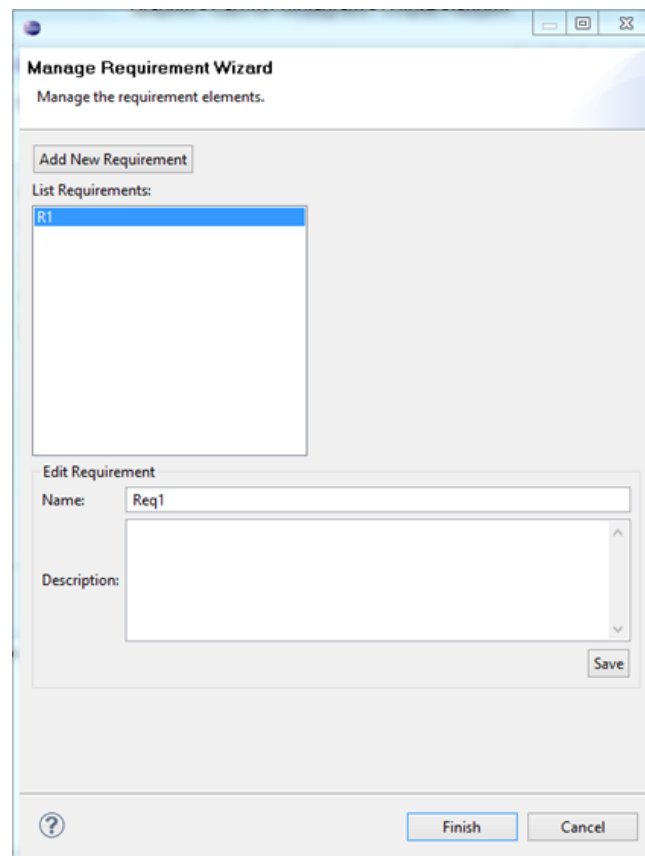


Figure 4.10 Edit Requirement view in Manage Requirement Wizard

Add Behavior Tree Node feature

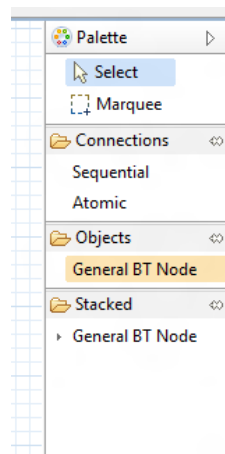


Figure 4.11 Select General BT Node label in the palette to create a new BT Node

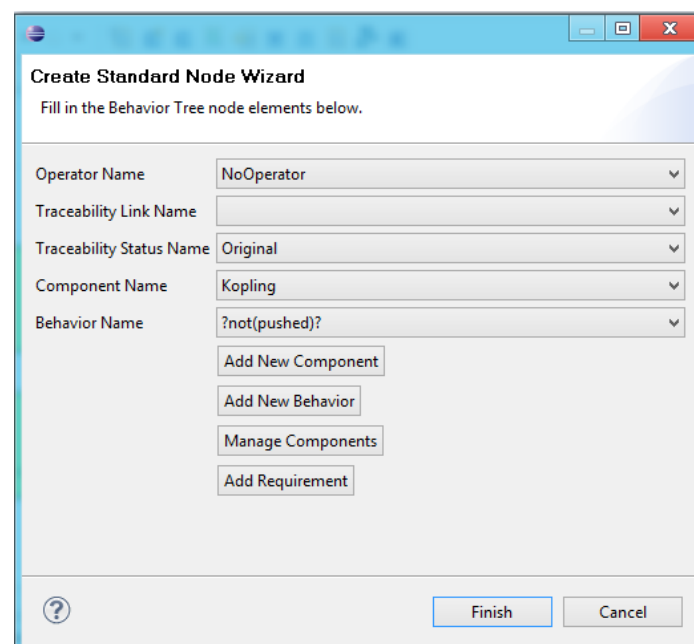


Figure 4.12 User can specify a new BT Node in Create Standard Node wizard

Add Behavior Tree Connection feature

There are two types of connection in Behavior Tree specification: sequential and atomic connection. To create an edge between nodes, click Sequential or Atomic label from the palette, then click both nodes

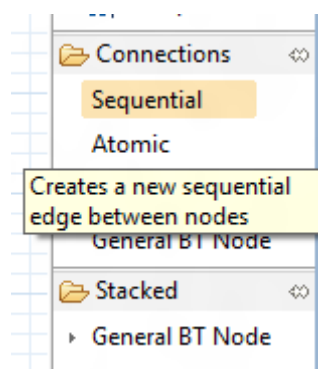


Figure 4.13 Behavior Tree Connection

Validate Behavior Tree



Figure 4.14 Validate BT icon in the toolbar

After validating the BT, a message will appear to inform whether the BT is valid or not.

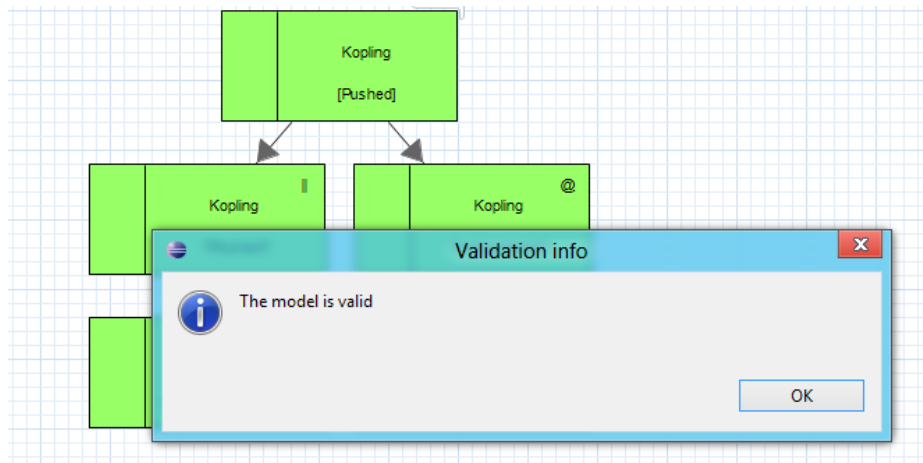


Figure 4.15 Validation BT

Generate BT Code

After the Tree is added to the diagram editor, user can generate the .bt code by pressing Generate BT Code button in the toolbar. The generated BT code will then appear in the Project Explorer.

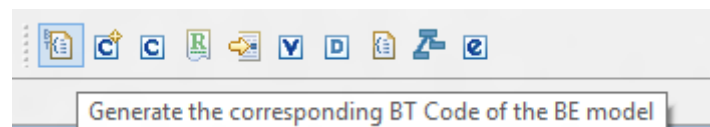


Figure 4.16 Generate BT code icon in the toolbar

Generate Java Code

Given a BT model exists, user can generate executable code using Generate Java Code feature which obviously will translate the BT model to Java code.

Verify Behavior Tree

This tool can be used to verify the BT model.

Debugging Tools

User can also debug an already created BT diagram by clicking Debug BT Diagram button in the toolbar. A new perspective will be opened. In this perspective, user can animate the Behavior Tree execution and monitor the execution sequence. This feature will also inform user about the error found in the Behavior Tree design.

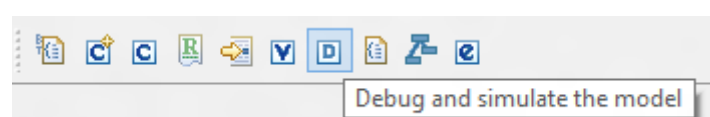


Figure 4.17 Debug and simulate BT icon in the toolbar

The detailed achieved work and constraints for this project is explained in the table below

Achieved Work and Constraints Table

No.	Plan	Result	Constraints	Solution Plan to the Constraints
1.	Create BT Model Checker	BT Model Checker		
2.	Create BT Execution Simulator and Debugger	BT Execution Simulator and Debugger Tool		
3.	Create BT Code Generator	BT Code Generator		
4.	Create Diagram Editor and Textual Editor Development Tool for BT	GraphBT as Diagram Editor and Textual Editor Development Tool	Some known bugs exists, such as 1) some basic features can't be invoked from context menu, 2) create first node as reversion operator doesn't work properly	Send bug report, ask to the forum
5.	Tool Integration	Integrated tool which contains Diagram Editor and Textual Editor Development Tool, BT Execution Simulator and Debugger, BT Code Generator		
6.	User Manual, Release Note, and Installation guide	User Manual, Release Note, and Installation guide		
7.	Final Report and Technical Report	Final Report and Technical Report		

V. Conclusion and Recommendation

5.1 Conclusion

The goal of this project is to develop and integrate software development tool for creating Behavior Tree. One of the most important features from the resulted product is diagram editor for developing Behavior Tree. This feature will provide significant advantage for Behavior Tree research and development because it will make Behavior Tree easier to construct. Other major features provided in the resulted product are BT Code Generator for generating BT model to executable code, BT Model Checker for Behavior Tree verifying purpose, and BT Debugger Tools and BT Execution Simulator for debugging and simulation purpose.

5.2 Recommendation

The current development of Behavior Tree concept is not fully established yet to support a detailed system development. However, we can still use it for simulation and model check purpose but still leave out the implementation phase to be implemented by a real human software engineer. The implementation process should be researched thoroughly before Behavior Tree shown its full capability in automatic implementation phase.

VI. Project Member

6.1 List of Project Member and Job Description

Position	Person in Charge	Responsibility
Project Manager	Emerson Simbolon	<ul style="list-style-type: none">- Manage the time and milestone/checkpoint of each job point- Manage the relation between Stakeholders- Manage the team project- Manage bug and solution- Manage resource needed
System Analyst	Agung Pratama	<ul style="list-style-type: none">- Analyze the requirement and data flow design
Lead Programmer	Ardi	<ul style="list-style-type: none">- Manage and coordinate the implementation of software design
Programmer	Team	<ul style="list-style-type: none">- Implement the software design into a real program
System Tester	Agung Pratama	<ul style="list-style-type: none">- Test the software before deploying it to make sure there is no critical bug after the release
Documenter	Ikhsanul Habibie	<ul style="list-style-type: none">- Create the software documentation, like software specification, database design, etc
UI Designer	Chairunissa Atimas N., Ikhsanul Habibie	<ul style="list-style-type: none">- Design User Interface(UI) of the tool. Also concern about user experience, easiness, and intuitiveness in using the software
Outsource Manager	Ardi	<ul style="list-style-type: none">- Find and manage outsources
Administration	Chairunissa Atimas N.	<ul style="list-style-type: none">- Manage the money income and outcome of project, also buy things

6.2 Log

Log Kerja Emerson Simbolon

	Hari, Tanggal	Deskripsi	Jumlah Jam
April	Senin, 2 April	Eksplorasi mengenai spray dan graphiti	2
	Rabu, 4 April	Rapat awal greenc(l)aude	2
	Jumat, 6 April	Instalasi spray dan mempelajari cara penggunaan	4
	Senin, 9 April	Rapat mingguan	2
	Rabu, 11 April	Mempelajari spray	5
	Minggu, 15	Instalasi dan mempelajari graphiti	5
	Senin, 16 April	Rapat mingguan	2
	Selasa, 17 April	Menganalisa kekurangan yang ada pada code generator dan mengumpulkan hasil analisa yang dibutuhkan nantinya	5
	Jumat, 20 April	Persiapan presentasi	1
	Jumat, 20 April	Presentasi grup IMHERE	1
	Senin, 23 April	Rapat mingguan	2
	Jumat, 27 April	Mempelajari EMF	2
	Minggu, 29 April	Membuat diagram pemodelan BT	4
Mei	Senin, 7 Mei	Rapat mingguan	2
	Senin, 28 Mei	Melakukan desain flow dari komponen plugin	4
Juni	Senin, 25 Juni	Membuat account github, mempelajari dan mengunggah pekerjaan	4
	Senin, 25 Juni	Rapat mingguan	2
	Selasa, 26 Juni	Merapikan environment kerja, supaya kondusif dalam mengerjakan proyek bersama	2
Juli	Senin, 2 Juli 2012	Debug error dari pengerjaan awal dan menambahkan library untuk menampung method yang common	6
	Senin, 2 Juli 2012	Rapat mingguan	2
	Rabu, 4 Juli 2012	Mengerjakan "connection feature"	5
	Senin, 9 Juli 2012	Memperbaiki fitur update value yang sudah dikerjakan oleh habibie	4
	Senin, 16 Juli 2012	Rapat IMHERE	1
	Selasa, 17 Juli 2012	Memperbaiki error, mempelajari cara kerja wizard dan menambahkan fitur untuk membuat BT Diagram dari wizard	6
	Jumat, 20 Juli 2012	Membuat wizard untuk membuat node baru	3
	Senin, 23 Juli 2012	Memperbaiki wizard untuk membuat BT Diagram	2
	Selasa, 24 Juli 2012	Menambahkan form untuk buat komponen baru, membuat behavior baru, mempelajari cara kerja editor dan menambahkan editor untuk mengedit file dengan ekstensi ".btdiagram"	8
	Sabtu, 28 Juli 2012	integrasi wizard, properties, dan diagram editor	1
	Minggu, 29 Juli 2012	Menambahkan fitur untuk menghasilkan BT File dari diagram	5
Agustus	Sabtu, 4 Agustus	Menambahkan model TextBE	5
	Senin, 6 Agustus	Presentasi MONEV	1
	Senin, 6 Agustus	Mengerjakan fitur yang masih memiliki bug	2
	Senin, 6 Agustus	Mempelajari cara memasukkan icon	1.5
	Senin, 6 Agustus	Mempelajari cara kerja validasi	1
	Selasa, 7 Agustus	Mempelajari cara mengeluarkan instance .xml dari sebuah .bt	3
	Kamis, 9 Agustus	Integrasi BT Debugger dengan BT Code Generator	4
	Senin, 13 Agustus	Menggabungkan BT Debugger, BT Code Generator, dan menambahkan wizard untuk membuat BT Project	4
	Sabtu, 18 Agustus	Memperbaiki "add connection feature"	5

	Rabu, 22 Agustus	Menambahkan fitur “apply layout”, memperbaiki debugger dan tool code generator	8
	Kamis, 23 Agustus	Memperbarui wiki dan manajemen issue	1.5
	Jumat, 24 Agustus	Menambahkan fitur “extract from file”	6.5
	Sabtu, 25 Agustus	Menambahkan icon	2
	Selasa, 28 Agustus	Memperbaiki resource	3
	Rabu, 29 Agustus	Manage progress	2
September	Sabtu, 1 September	Memperbaiki fitur pembuatan connection	1
	Minggu, 2 September	Memperbaiki fitur menghapus node	4
	Selasa, 4 September	Memperbaiki atomic node dan menambahkan mekanisme pada fungsi perhitungan tinggi behavior tree	5
	Rabu, 5 September	Memperbaiki bug yang terjadi dan menambahkan fitur menghapus isi editor diagram	4

Log Kerja Agung Pratama

	Hari, Tanggal	Deskripsi	Jumlah Jam
April	Rabu, 4 April	Instalasi Zest dan software presequites	2
	Rabu, 4 April	Eksplorasi mengenai Zest dan mencari tutorial serta example awal, serta membuat custom node	6
	Kamis, 5 April	Melanjutkan custom node di Zest	4
	Senin, 9 April	Rapat Mingguan	2
	Senin, 16 April	Rapat Mingguan	2
	Senin, 16 April	Mendesain plugin debugger untuk mengintegrasikan context menu pada .xml	2.5
	Senin, 23 April	Integrasi parser Emerson dengan plugin Eclipse	3
	Senin, 23 April	Rapat mingguan	2
	Senin, 23 April	Integrasi parser Emerson dengan plugin Eclipse	6.5
	Selasa, 24 April	Memperbaiki kesalahan pada saat integrasi parser	2
	Selasa, 24 April	Mempelajari pembuatan custom layouting	4
	Rabu, 25 April	Modifikasi model hasil parser	2.5
	Minggu, 29 April	Membuat tombol pada debugging tools	6
	Senin, 30 April	Melanjutkan pembuatan tombol pada debugging tools	3
Mei	Selasa, 1 Mei	Membuat view untuk tampilan list variabel dan memodifikasi BTParser	7
	Rabu, 2 Mei	Membuat watch list variabel	8
	Kamis, 3 Mei	Mengubah BT Parser	8
	Jumat, 4 Mei	Diskusi dengan Pak Ade	2
	Jumat, 4 Mei	Modifikasi BT Parser	5
	Senin, 21 Mei	Desain BT Simulator	11
	Selasa, 22 Mei	Desain BT Simulator	10
	Rabu, 23 Mei	Desain BT Simulator	10
	Kamis, 24 Mei	Implementasi simulasi pada composition: sequence, parallel, dan alternative	8
	Sabtu, 26 Mei	Implementasi atomic, state, guard, dan selection	11
	Minggu, 27 Mei	Implementasi internal input dan internal output	3
	Senin, 28 Mei	Implementasi external input dan external output	6
Juni	Jumat, 1 Juni	Membuat implementasi generate BT Trace File dan testing plugin	8
	Sabtu, 2 Juni	Menulis tentang behavior tree dan behavior engineering	3
	Senin, 4 Juni	Menulis tentang teknologi yang dipakai: zest, draw2d, eclipse plugin,	8

		JFace dan menemukan kesalahan pada studi kasus Producer Consumer	
	Selasa, 5 Juni	Menulis rancangan spesifikasi dari debugging tool, merancang studi kasus hasil revisi Producer Consumer	11
	Rabu, 6 Juni	Menulis rancangan spesifikasi debugging tool dan animasi pada debugger tool.	9
	Kamis, 7 Juni	Menuliskan simulasi pada penanganan penjadwalan	4
	Jumat, 22 Juni	Progress report dan berdiskusi dengan Pak Ade	4
	Sabtu, 23 Juni	Menambahkan fitur "Save to JPG" dari sebuah diagram	5
	Senin, 25 Juni	Rapat mingguan	2
Juli	Senin, 2 Juli	Rapat mingguan dan mempelajari control version Git pada Eclipse	3.5
	Senin, 9 Juli	Rapat mingguan dan mempelajari deployable plugin sebagai .zip file	3.5
	Selasa, 31 Juli	Rapat mingguan dan review integration code dengan Ikhsanul Habibie	2
August	Senin, 6 Agustus	Rapat mingguan, sinkronisasi github, dan testing plugin graph editor	5
	Selasa, 14 Agustus	Sinkronisasi github, dan testing plugin graph editor	1

Log Kerja Ardi

	Hari, Tanggal	Deskripsi	Jumlah Jam
April	Minggu, 8 April	Mempelajari struktur kode sumber	1.5
	Senin, 9 April	Mempelajari kode untuk translasi SAL	2
	Selasa, 10 April	Mempelajari definisi kelas-kelas Java	1.5
	Rabu, 11 April	Mengintegrasikan beberapa kode ke dalam sebuah project	1.5
	Jumat, 13 April	Diskusi terkait referensi	3.5
	Sabtu, 14 April	Diskusi terkait kode sumber	1
	Senin, 16 April	Rapat dan presentasi	1.5
	Jumat, 20 April	Presentasi IMHERE	3.5
	Senin, 23 April	Instal Phyton, dan mengujinya dengan studi kasus	5
	Selasa, 24 April	Rapat mingguan, dan mempelajari SAL-BT Translator	5
	Rabu, 25 April	Mempelajari bahasa SAL	1.5
	Jumat, 27 April	Diskusi gambaran umum dari BT Integrated Tool	0.5
Mei	Senin, 30 April	Membaca tesis Toby Myers	3
	Kamis, 3 Mei	Melanjutkan membaca tesis Toby Myers	3
	Senin, 7 Mei	Mempelajari studi kasus untuk BT: Producer Consumer	3
	Selasa, 8 Mei	Memperbaiki bagian BEGIN BT2SAL Translator	4
	Rabu, 9 Mei	Mempelajari kode translasi milik BT2SAL Translator, dan membandingkannya dengan translator Nisansala	2
	Kamis, 10 Mei	Modifikasi kode translasi milik BT2SAL Translator	3
	Jumat, 11 Mei	Diskusi dan pertemuan	3
	Kamis, 17 Mei	Membaca laporan SAL Language	2
	Jumat, 18 Mei	Melakukan eksperimen dan perbandingan antara BT2SAL dan Nisansala dengan studi kasus r6 Oven	5
	Sabtu, 19 Mei	Melakukan eksperimen untuk studi kasus Oven	4
	Senin, 21 Mei	Melakukan eksperimen untuk studi kasus Producer Consumer. Perbaiki BT2SAL Translator	4
	Selasa, 22 Mei	Menguji BT2SAL Translator untuk mencari kelemahannya, dan mendiskusikannya.	5
	Rabu, 23 Mei	Membuat BT Specification untuk studi kasus Train	1
	Kamis, 24 Mei	Merangkum dan menganalisa hasil	3
	Jumat, 25 Mei	Diskusi progress	4
	Sabtu, 26 Mei	Mendiskusikan kode BT2SAL Translator	4
	Minggu, 27 Mei	Dokumentasi kode	4

	Senin, 28 Mei	Reversion untuk BT2SAL Translator	9
	Selasa, 29 Mei	Implementasi aturan SELECTION	7
	Jumat, 30 Mei	Implementasi aturan reference	4
Juni	Jumat, 1 Juni	Menulis tentang behavior tree dan behavior engineering	2
	Sabtu, 2 Juni	Menulis tentang kode SAL	2
	Minggu, 3 Juni	Menulis tentang Trace Animator dan mempelajari serta membuat spesifikasi BT untuk studi kasus Car System	4
	Senin, 4 Juni	Menguji studi kasus dengan semua tools BE dan tulis hasilnya	3
	Selasa, 5 Juni	Menulis tentang SALTrace dan BTTrace	2
	Rabu, 6 Juni	Menulis tentang BT2SAL Translator	2
	Kamis, 14 Juni	Membaca dan mempelajari aturan translasi pada penelitian sebelumnya	4
	Jumat, 15 Juni	Memperbaiki translasi event dan internal input	3
	Sabtu, 16 Juni	Menguji studi kasus dengan BT2SAL Translator yang telah diperbaiki	3
	Senin, 18 Juni	Dokumentasi kode	2
	Minggu, 24 Juni	Mempelajari protocol Peterson dan Bakery di tutorial SAL	3
	Senin, 25 Juni	Desain UX model checking dan integrasi BT2SAL Translator ke Eclipse	7
	Selasa, 26 Juni	Memperbaiki bug di internal input dan internal output	4
Juli	Jumat, 7 Juli	Membuat alur operasi dan desain UI model Checking	4
	Sabtu, 8 Juli	Desain UI model Checking	1.5
	Selasa, 17 Juli	Tutorial SWT	1.5
	Kamis, 19 Juli	Setup proyek GitHub	1.5
	Jumat, 20 Juli	Mempelajari UI Component dan layout SWT	2
	Minggu, 22 Juli	Mempelajari SWT + JFace	1.5
	Selasa, 24 Juli	Membuat layout model checking	2
	Kamis, 26 Juli	Mempelajari semua proyek	1.5
	Selasa, 31 Juli	Membuat UI pada proyek, dan pertemuan	3
Agustus	Selasa, 7 Agustus	Integrasi model yang telah diverifikasi dan menambahkan Formula View	9
	Rabu, 8 Agustus	Melanjutkan integrasi	2
	Senin, 13 Agustus	Memperbaiki penambahan Formula view	4
	Selasa, 14 Agustus	Memperbaiki bug pada wizard model	5
	Agustus	Menuliskan formula data file ke eclipse	2.5
	Agustus	Memperbaiki bug pada wizard Verify	1.5
September	Agustus	Integrasi model checker, diskusi dan integrasi graphbt tool	4.5
	Senin, 10 September	Mempelajari bagaimana melakukan setting path dari command prompt ke cygwin	4
	Selasa, 11 September	Install phyton	1
	Rabu, 12 September	Integrasi model checker	6

Log Kerja Ikhsanul Habibie

	Hari, Tanggal	Deskripsi	Jumlah Jam
April	Senin, 9 April	Rapat mingguan	2
	Kamis, 12 April	Mempelajari BT dan paper tentang BT secara general	2
	Senin, 16 April	Rapat mingguan	2
	Selasa, 17 April	Instalasi dan mempelajari fitur-fitur graphiti secara umum	3

	Rabu, 18 April	Pembuatan general BT node dengan model dummy	2
	Kamis, 19 April	Pembuatan general BT node dengan model dummy	4
	Jumat, 20 April	Presentasi grup IMHERE	2
	Senin, 23 April	Penambahan direct editing	2
	Senin, 23 April	Rapat mingguan	2
	Kamis, 26 April	Debugging BT Node, menambahkan connection	3
	Jumat, 27 April	Memperbaiki connection dari BT Node, menambahkan property editing untuk text pada BT Node	1
	Minggu, 29 April	Menambahkan listener untuk property	2
Mei	5 Mei	Rapat rutin mingguan	2
Juni	25 Juni	Rapat rutin mingguan	2
Juli	Senin, 2 Juli	Rapat mingguan	2
	Rabu, 4 Juli	Mengerjakan "connection feature"	2
	Minggu, 8 Juli	Memperbaiki fitur direct edit	3
	Senin, 9 Juli	Memperbaiki fitur update	2
	Senin, 16 Juli	Melakukan beberapa perbaikan di beberapa bagian fitur seperti add, connection, direct edit, dan membuat branch baru di gitHub	2
	Rabu, 18 Juli	Mempelajari wizard	2
	Kamis, 19 Juli	Mengimplementasikan wizard pertama untuk create node	2
	Jumat, 20 Juli	Membuat rancangan awal untuk property	2
	Senin, 23 Juli	Revamping wizard untuk add node baru	2
	Rabu, 25 Juli	Integrasi dengan program untuk property yang sudah diperbaiki	2
	Kamis, 26 Juli	Memperbaiki property untuk component dan behavior	1

Log Kerja Chairunissa Atimas N.

	Hari, Tanggal	Deskripsi	Jumlah Jam
April	Minggu, 8 April	Mempelajari ZEST	1
	Senin, 9 April	Rapat mingguan	2
	Minggu, 15 April	Mempelajari Graphiti	1
	Senin, 16 April	Rapat mingguan	2
	Kamis, 19 April	Design UI	4
	Jumat, 20 April	Finalisasi design UI	2
	Senin, 23 April	Rapat mingguan	2
	Senin, 30 April	Browsing icon pada Eclipse	1
Mei	5 Mei	Rapat rutin mingguan	2
Juni	25 Juni	Rapat rutin mingguan	2
Juli	Sabtu, 7 Juli	Membuat desain UI Create New Component & Create New Project	4
	Minggu, 8 Juli	Membuat desain UI Manage Library	2
	Kamis, 19 Juli	Membahas Prototype dan mempelajari pembuatan UI	4
	Sabtu, 21 Juli	Membuat desain UI New Behavior & Requirements	3
	Kamis, 26 Juli	Mempelajari dan membuat UI Manage Component	5
	Jumat, 27 Juli	Membuat UI Manage Requirement dan Edit Behavior	5
	Selasa, 31 Juli	Rapat mingguan dan melanjutkan pembuatan UI	6
Agustus	6 Agustus	Menambahkan edit behavior wizard, edit requirement wizard, melakukan validasi untuk setiap wizard dan merapihkan wizard	6
	7 Agustus	Melanjutkan validasi wizard dan merapihkan wizard	6
	14 Agustus	Membuat site untuk Proyek	3
September	3 September	Memperbaiki website	3

	10 September	Edit GUI for wizard manage component, new BT Project, and create	4
	13 September	Create manage library wizard, description field for component and behavior, and requirement compact	4
	17 September	Membuat add behavior di create component, dan memperbaiki laporan	6

LOG KERJA NURUL QOMARIYAH

Log Kerja Nurul Qomariyah

	Hari, Tanggal	Deskripsi	Jumlah Jam
Juli	Sabtu, 14 Juli	Tutorial Eclipse Plugin Development (SWT, JFace, Perspective, Actions, Dialogs, Views)	5
	Senin, 16 Juli	Tutorial Eclipse Plugin Development (Track Resource Changes, Preference Pages, Properties)	3.5
		Rapat Koordinasi IMHERE	1
	Kamis, 17 Juli	Tutorial Eclipse Plugin Development (Properties)	2
	Senin, 23 Juli	Membuat form Properties	3
Agst	Senin, 6 Agustus	Membuat validasi Wizard Create Component	3
Sept	Rabu, 12 September	Menulis laporan akhir – Technical Report dan Final Report	2
	Kamis, 13 September	Rekap Log kerja	5