

**TABLE OF CONTENTS**

<b>CHAPTER</b>	<b>TITLE</b>	<b>PAGE</b>
	<b>DECLARATION</b>	<b>ii</b>
	<b>DEDICATION</b>	<b>iii</b>
	<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
	<b>ABSTRACT</b>	<b>v</b>
	<b>ABSTRAK</b>	<b>vi</b>
	<b>TABLE OF CONTENTS</b>	<b>vii</b>
	<b>LIST OF TABLES</b>	<b>xii</b>
	<b>LIST OF FIGURES</b>	<b>xiii</b>
	<b>LIST OF ABBREVIATIONS</b>	<b>xvi</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1 Overview	1
	1.2 Background of the problem	4
	1.3 Problem Statement	7
	1.4 Project Aim	8
	1.5 Objective of Study	8
	1.6 Scope of the Project	8
<b>2</b>	<b>LITERATURE REVIEW</b>	<b>9</b>
	2.1 Overview	9
	2.2 Code Generation Technology	10

2.3	Pervious Reviews on Code Generation	11
2.3.1	Comparison of Dataflow Architecture and Real Time Workshop Embedded Coder	11
2.3.2	Reviews the Object Oriented Design on ERT	12
2.4	ERT Code Generation Frameworks	13
2.4.1	UML-RT	16
2.4.1.1	Time Managing	17
2.4.1.2	UML-RT and Code Generator Features	20
2.4.1.3	UML-RT and Robot Controller	21
2.4.2	Giotto	24
2.4.2.1	Time Managing with Giotto Compiler	26
2.4.2.2	Giotto Code Generation	31
2.4.2.3	A Giotto–Base Autonomous Helicopter System	31
2.4.3	Simulink	33
2.4.3.1	Time Managing in Simulink	35
2.4.3.2	Simulink and Code Generator (Real-Time Workshop) Features	36
2.4.4	Labview	36
2.4.4.1	Time Managing	37
2.4.4.2	Labview Code Generation	38
2.4.5	Component oriented programming (COP)	39
2.4.5.1	Mapping of component behaviour to task and time managing	40
2.4.5.2	Analysis pattern for autonomous Mobile Robot software	40
2.4.5.3	COP Framework	41
2.4.5.4	Components approach in COP	42
2.5	Discuss on the Evaluation of the Framework	43

3	<b>RESEARCH METHODOLOGY</b>	
3.1	Overviews	46
3.2	Research Methodology	47
3.3	Operational Research Framework	49
3.4	ERT Case Study	51
	3.4.1 Card and Fingerprint based Time Recording Terminal	52
	3.4.2 Concurrency operation in STPro	54
4	<b>CODE GENERATOR FOR EMBEDDED REAL TIME SOFTWARE</b>	55
4.1	Overviews	55
4.2	Criteria of Evaluations for CBD Methodologies in ERT Code Generator	56
4.3	Reason on choosing the four criteria	58
4.4	UML RT	58
	4.4.1 Iterative development in UML RT	59
	4.4.2 Optimized Design Concept	60
	4.4.3 Large scale development in UML RT	61
	4.4.4 Integration and adaptation	62
4.5	Giotto	62
	4.5.1 Iterative development	63
	4.5.2 Optimization development	64
	4.5.3 Large scale development	66
	4.5.4 Interaction and adaptation	67
4.6	Simulink	67
	4.6.1 Iterative development	68
	4.6.2 Optimizing generated code	71
	4.6.3 Large-scale development	76
	4.6.4 Integration and adaption	77
4.7	Labview	77
	4.7.1 Iteration development on Labview	77
	4.7.2 Optimizing Generated Code	79
	4.7.3 Large scale development	81

	4.7.4	Integration and adaption	81
4.8		Discussion on the Evaluation of Criteria for Code Generators	81
4.9		Discusstion	84
<b>5</b>		<b>RATIONAL ROSE REAL TIME TOOL</b>	<b>85</b>
5.1		Introduction	85
5.2		Component Composition	86
5.3		Code generation	90
	5.3.1	Capsule	91
	5.3.2	Capsule State Diagrams	92
	5.3.3	Classes	93
	5.3.4	Associations	93
	5.3.5	Dependency	95
	5.3.6	Internal messages	95
5.4		Following criteria by Rational Rose RT	97
	5.4.1	Supporting the Large Scale by Rational Rose RT	98
	5.4.2	Supporting the Iterative Development by Rational Rose RT	101
	5.4.3	Discussion	102
<b>6</b>		<b>COP TOOL</b>	<b>104</b>
6.1		Introduction	104
6.2		COP TOOL requirement	105
	6.2.1	Module 1:Component Development	105
	6.2.2	Module 2:Component Integration	105
	6.2.3	Module 3: Code generation	106
6.3		COP tool design	106
	6.3.1	COP Composition Class Diagram	108
	6.3.2	COP Code Generation class diagram	108
	6.3.3	User Interface	112
	6.3.4	The Tool architecture design	115
6.4		Following criteria by COP tool	117

	6.4.1	Supporting the Large Scale by COP tool	117
	6.4.2	Supporting the Iterative Development by COP tool	119
	6.4.3	Discussion	121
7		<b>CONCLUSION</b>	124
	7.1	Summary	115
	7.2	Research Contribution	126
	7.3	Future works	127
		<b>REFERENCES</b>	128

**LIST OF TABLES**

<b>TABLE NO</b>	<b>TITLE</b>	<b>PAGE</b>
2.1	Evaluation based on criteria UML RT commercial system	18
2.2	Evaluation based on criteria of Giotto commercial system	27
2.3	Evaluation based on criteria Simulink commercial system	34
2.4	Summary of the comparative evaluation	45
3.1	Shows a STPro system which is provided access control system	53
4.1	Introduce some code generation criteria	56
4.2	Summary of the comparative evaluation of frameworks based on criteria	82
5.1	Send and received of capsule code	96
6.1	Comparing the COP features and Rational Rose	123

## LIST OF FIGURES

FIGURE NO	TITLE	PAGE
1.1	Example of embedded systems	2
1.2	Embedded real time system	3
2.1	Programming model of a node processor (wikender, 1999).	14
2.2	Sequence diagram with time consuming	17
2.3	Timing diagram example	18
2.4	Capsule structure diagrams	22
2.5	Capsule structure diagrams	22
2.6	Capsule state transaction diagrams	23
2.7	Component diagram	23
2.8	Traditional control systems development process	25
2.9	The Giotto based control system development	25
2.10	Giotto time tasking diagram	26
2.11	Giotto program example (control off or controller on specification on helicopter controller)	29
2.12	Workflow of the design framework that iteratively refines code generation using schedulability	30
2.13	Time tasking on Giotto	32
2.14	There are no race conditions	32
2.15	The Giotto case block in simulink	33
2.16	Typical output logic for a digital delay generator	38

2.17	Architecture of Lab VIEW in embedded real time system	39
2.18	Architecture Pattern in COP	41
2.19	Motor control composite component	42
2.20	A PID component documented in block form	43
3.1	Research Design Methodology	48
3.2	Operational Framework	50
3.3	STPro Attendance system	54
4.1	Giotto based control systems development	64
4.2	Shows the rapid prototyping development process	70
4.3	Algorithm design and prototyping	71
4.4	Three same product plot on Simulink	75
4.5	Three same product on matlab	75
5.1	class diagram in the STPro System	87
5.2	UML-RT structure model of the STPro	87
5.3	Use case diagram of the STPro System	88
5.4	Sequence diagram representing message sequences between the Capsules	89
5.5	State diagram of STPro	90
5.6	Generalization in class diagram	89
5.7	Relations in class diagram	96
5.8	Relations between class and capsule	96
5.9	Relations in class diagram	97
5.10	considering the library in Rational Rose	101
6.1	Use case diagram of the composition part	107
6.2	Class diagram of the composition part	108
6.3	Class diagram of the code generation part	109
6.4	Definition of code generator	109
6.5	Header of initialization	109
6.6	Body of initialization	110
6.7	Data type declaration	110
6.8	Data declaration error by underline	110
6.9	Execution of Capsule	111
6.10	Main of project	111



6.11	Synchronization part of code generator	111
6.12	Composition of COP tool	113
6.13	Composition of COP tool and ComponentsTab	114
6.14	Code generation of COP tool	115
6.15	Tool architecture design	116
6.16	Composition of COP tool and library Tab	118
6.17	COP Library is reachable by Check Box	118
6.18	Error connections	120
6.19	Representing data type error in generated code	121

**LIST OF ABBREVIATIONS**

COP	-	Component Oriented Programming
RTS	-	Real Time System
ERT	-	Embedded Real Time System
MASCOT	-	Modular Approach to Software framework Operation and Test
CBSE	-	Component Base Software System
CBD	-	Component Base Development
GUI	-	Graphical User Interface
OOP	-	Object Oriented Programming
AOP	-	Aspect Oriented Programming
RTOS	-	Real Time Operation System
AMR	-	Autonomous Mobile Robot
DSP	-	Digital Signal Processor
REI	-	Rose Extensibility Interface
ADL	-	Architecture Description Languages
GRT	-	Generic Real Time
FPGA	-	Field-Programmable Gate Array
regex	-	Regular Expression
UI	-	User Interface
SOA	-	Service Oriented Architecture