

Applicability of Agent Technology for Software Release Management



B.A.D.A.S. Bogoda
University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
08/10001
www.lib.mrt.ac.lk

Faculty of Information Technology

University of Moratuwa

September 2010

Applicability of Agent Technology for Software Release Management

B.A.D.A.S. Bogoda



University of Moratuwa, Sri Lanka.
08/10001
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Dissertation submitted to the Faculty of Information Technology,
University of Moratuwa, Sri Lanka for the partial fulfillment of the
requirements of the Degree of M.Sc. in Artificial Intelligence

September 2010

Declaration

I declare that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a Degree or a Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organization.

B.A.D.A.S. Bogoda

Name of Student

Signature of Student

Date

Supervised by 

University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Prof. Asoka S. Karunananda

Name of Supervisor(s)

Signature of Supervisor(s)

Date

Dedication

To My Parents



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Acknowledgements

I am heartily thankful to my supervisor, Asoka S. Karunananda, whose encouragement, guidance and support from the initial to the final level enabled me to develop this solution.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Abstract

Information Technology industry is one of the most widely spreading industry around the world in recent past as its applicability and adaptability nature for various streams. Influence created by IT industry, on various streams help to accelerate their development in large portion with in a short period of time. Because of quick reaction in problem solving and easy way of storing and retrieving information people tend to replace existing manual systems by computerized systems.

Enterprise level applications are developed as a combination of several components. Modification done in a one component is used to fulfill some functionality of other component. When preparing those modification to deliver for the customer it is required to know which component's modification is highly depend on each other and which is not depend on other. Currently this handle by human where developers go through the modification and identify which component is highly depending on the modification done on other components. Developers need to have good communication among each other to identify correct order, the components should arrange. Once the order is identified they start compiling and building jar files. This process is highly time consuming task as there are frequent updates done for the code when fixing issues. This subject to reduce developer's effective time he can work. It's a big burden for the company as well.

In this thesis it will discuss how to solve above problem by automating the software release management process using Multi-agent technology. In the literature review chapter it will discuss about different researches conducted related to software release management domain. In the design chapter it will give high level picture about the design. Then gradually it will explain how agents solve this by communicating with each other, under implementation chapter. At the evaluation chapter it discuss how implemented solution has been tested and what are the other advantages it can gain by overcoming limitations of the existing release management tools. Under conclusion chapter it discusses whether each objective has achieved by providing appropriate test samples.

Contents

	Page
Chapter 1 – Introduction	01
1. 1 Introduction	01
1. 2 Background and motivation	01
1. 3 Aim	02
1. 4 Objectives	02
1. 5 Resource Requirements	02
1. 6 Summary	03
Chapter 2 – Current trends in Software Release Management	04
2.1 Introduction	04
2.2 Waterfall Model	04
2.2.1 Requirements	05
2.2.2 Design	05
2.2.3 Implementation	06
2.2.4 Verification	06
2.2.5 Maintenance	07
2.3 Maintaining software product	07
2.4 Movements in Software Release Management	08
2.4.1 Standardized Release Management System	08
2.4.2 Cost Effective Release Management System	12
2.4.3 Flexible Release Management Systems for Component Based Products	13
2.5 Summary	16
Chapter 3 – Technology behind in Software Release Management	17
3. 1 Introduction	17
3. 2 Technologies already in use	17
3.2.1 CruiseControl	17
3.2.2 Maven	19
3.2.3 Multi Agent Systems Technology	20
3.2.3.1 Agent Communication	20

3.2.3.2 Ontology	21
3.2.3.3 Lifecycle of Agent	21
3. 3 Summary	22
Chapter 4 – Multi Agent approach for Release Management	23
4. 1 Introduction	23
4. 2 Proposed Solution	23
4.2.1 Inputs	23
4.2.2 Outputs	23
4.2.3 Process	24
4.2.4 Users	24
4.2.5 Features	24
4. 3 Functional Specification	24
4. 4 Summary	25
Chapter 5 – Design of Release Management Tool	26
5. 1 Introduction	26
5. 2 Analysis and Design	26
5.2.1 Initialization	27
5.2.1.1 Retrieving release dates	27
5.2.1.2 Setup repository libraries	27
5.2.1.3 Handling authentication	27
5.2.1.4 Retrieving history data	28
5.2.2 JADE Multi Agent System	28
5.2.2.1 Agent creation	28
5.2.2.2 Update local files	28
5.2.2.3 Agent communication	29
5.2.2.4 Agent deletion	29
5.2.3 Apache Ant Compiler	29
5.2.3.1 Invoke ant targets	29
5.2.3.2 Building jar files	30
5.3 Agent's Lifecycle	30
5.4 Class Diagram and Activity Diagram	30
5.5 Summary	31

Chapter 6 – Implementation	32
6. 1 Introduction	32
6. 2 Initialization	32
6.2.1 Utility classes	32
6.2.2 Retrieving release dates	33
6.2.3 Setup repository libraries	33
6.2.4 Handling authentication	34
6.2.5 Retrieving history data	35
6. 3 JADE Multi Agent System	35
6.3.1 Agent creation	35
6.3.2 Update local files	36
6.3.3 Agent communication	37
6.3.4 Agent deletion	38
6. 4 Apache Ant Compiler	38
6.4.1 Invoke ant targets	39
6.4.2 Building jar files	39
6. 5 Summary	40
Chapter 7 – Evaluation	42
7. 1 Introduction	42
7. 2 Evaluating Reduction of Human Interaction	42
7. 3 Evaluating Reduction of Network Traffic	43
7. 4 Minimize the time spent on Release Management task	43
7. 5 Reduce Exceptions occur in Compilation and Build process	43
7. 6 Optimize CPU memory usage	43
7. 7 Summary	44
Chapter 8 – Conclusion and Further work	45
8.1 Introduction	45
8.2 Conclusion	45
8.2.1 Reduce Human Interaction	45
8.2.2 Reduce Network traffic	45
8.2.3 Minimizing the time spent on release management task	46

8.2.4 Reduce exceptions occur in compilation and build process	46
8.2.5 Optimize CPU memory usage	46
8.3 Problem Encountered	46
8.4 Limitation	47
8.5 Further Works	47
8.6 Summary	48
References	49
Appendix A	52
Appendix B	57
Appendix C	58



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Figures

	Page
Figure 2.1 – Waterfall Model	04
Figure 3.1 – Architecture of CruiseControl	18
Figure 3.2 – Architecture of Multi-agent system	20
Figure 5.1 – Top Level Architecture of the Proposed Multi Agent System	26
Figure 5.2 – Lifecycle of the Agent	30
Figure A.1 – Memory usage in initialization	52
Figure A.2 – Memory usage after termination	53
Figure A.3 – Initial state of agent creation	54
Figure A.4 – Agents’ communication progress	55
Figure A.5 – End of agents’ communication	56
Figure B.1 – Generated jar files	57
Figure B.2 – Apache Ant script file	57
Figure C.1 – Class Diagram	58
Figure C.2 – Sequence Diagram	59