

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

Application of the Context Map

----Modeling the Rational Unified Process and Rational ClearQuest

Yaozhong Chen

A MAJOR REPORT

IN

THE DEPARTMENT

Of

COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

CONCORDIA UNIVERSITY

MONTREAL, QUEBEC, CANADA

February 2002

© Yaozhong Chen, 2002



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-68459-8

Canada

Abstract

Application of the Context Map

---- Modeling the Rational Unified Process and Rational ClearQuest

Yaozhong Chen

As much as each of us would like to get up every day with only a few things to focus on, the competitive world we live in rarely allows such a luxury. Balancing complexity with simplicity is an art, and it is one that we must master. In today's competitive information technology world, the pressure is on to stay in front of your competition. The value and visibility of your development projects compel you to reduce time-to-market without sacrificing quality and with lower cost. There are so many high level representing technologies for information system and software engineering. With so many representing technologies, how do we choose the one that will help us get ahead of the curve? An advanced, powerful, easy to learning and lower cost technology named Context Map is introduced as an common modeling language. The purpose of this project is to illustrate that Context Map is not only a methodology and formalized notation for representing the knowledge, but also it's a powerful tool itself which can be used directly to manipulate on the formatted knowledge. The functionality and attributes of Context Maps are demonstrated by modeling of Unified Process Concepts, Rational Unified Process framework and Rational ClearQuest tool set.

Acknowledgements

No work can be done by the author alone, and many people participated in making the work possible. I would like to thank all of the individuals who have contributed on the representing methodology----Context Map for their hard work and creative think. I'd like to give special thanks to Dr. W. M. Jaworski who is both the creator of this methodology and my supervisor. My thanks are also extend to Dr. Peter Grogono , my co-supervisor. Finally, I would like to express my deep gratitude to my wife, Yuqing Tian, for her support.

Table of Contents

<i>List of Figures</i>	<i>vii</i>
1. Introduction	1
1.1 Purpose	1
1.2 Background	1
1.3 Project Scope.....	3
2 Context Map Introduction.....	4
2.1 Context Map Overview	4
2.2 Context Map Notation.....	5
3 Rational Unified Process(RUP) Introduction	7
3.1 RUP Overview	7
3.2 RUP Concept.....	8
4 Representing RUP by Context Map.....	9
4.1 Notation	9
4.2 Schema of the RUP map.....	10
4.3 Phase and Disciplines	12
4.4 Detailed RUP Work Flow Map	13
4.5 RUP Tools Map.....	15
4.6 RUP Templates Map	16
4.7 Comparison of the RUP and Context Map	17
5 Rational ClearQuest(RCQ) Introduction	19
5.1 ClearQuest Overview	19
5.2 Representing of Defect Tracking System Schema in ClearQuest.....	19
5.3 Query in ClearQuest	21
6 Representing RCQ by Context Map.....	23
6.1 RCQ Schema representing in Context Map.....	23
6.2 Data Stored in Context Map.....	25
6.3 Context query.....	27
6.4 Further investigating on other functions and comparison summary of both defect tracking tools	28

7	<i>Conclusion And Recommendation</i>	30
7.1	General Conclusion	30
7.2	Recommendations for Future Works	31
	<i>Bibliography</i>	33
	<i>Appendix</i>	34
1.	Discipline— Business Modeling View	34
2.	Discipline—Requirements View	36
3.	Discipline— Analysis and Design view	38
4.	Discipline— Implementation view	40
5.	Discipline— Test view	42
6.	Discipline— Deployment view	44
7.	Discipline— Configuration & Change Management view	45
8.	Discipline— Project Management view	47
9.	Discipline— Environment view	49

List of Figures

Figure 1 Context Map Example.....	5
Figure 2 The Rational Unified Process Overview	7
Figure 3 RUP Concepts.....	8
Figure 4 RUP Context Map Schema	10
Figure 5 Phases and Disciplines in the RUP Context Map.....	13
Figure 6 Detailed RUP Work Flow Context Map	14
Figure 7 RUP Templates Context Map	17
Figure 8 State transition Matrix in ClearQuest	20
Figure 9 Action Access Control Matrix in ClearQuest.....	21
Figure 10 Query tool of ClearQuest	22
Figure 11 Records displayed in ClearQuest	22
Figure 12 Defect Tracking System Schema in Context Map	24
Figure 13 Sample of the defect issue state flow	24
Figure 14 Defect Records Stored in Context Map	26
Figure 15 Context Query	27

Important Notice:

All of the Context Map Syntax and Patterns shown in above figures are Copyrights by W.M. Jaworski 1988-2002.

1. Introduction

1.1 Purpose

The purpose of this work is to modeling a software engineering process called Unified Process using Context Map then comparing it with a tool to implement the process called Rational Unified Process. Thus, many scattered software develop process concepts and relationship among these concepts are integrated into one context map so that a simple and clear view is presented before user's eyes. Moreover, the prominent context query tool gives users a convenient and flexible method to manipulate and control the entries. In addition, the Rational ClearQuest is also being modeled and compared to further explore many variant features of the Context Map and to illustrate the possibilities of merging those separated concepts into one view base.

1.2 Background

The Context Map, which originally was called jMap (joined map), was first introduced by Dr. W.M. Jaworski. The technology was initially developed for recovering and refining knowledge from legacy systems. By using the concept of spreadsheet structure, it is feasible to describe and process conceptual information. As its other name--joined map--showing, it can easily integrate different information into one consistent map. It can be applied in many domains, such as modeling, mining, evaluation of contexts, enterprises, methods, processes, projects, artifacts, databases, websites, information

system, knowledge models, ... with generic templates, domain experts, proprietary notational technology.

The Rational Unified Process or RUP product is a commercial tool for software engineering process developed by Rational Software Corporation [8]. It provides a disciplined approach to assigning tasks and responsibilities within a development organization. Its goal is to ensure the production of high-quality software that meets the needs of its end users within a predictable schedule and budget. The key to achieve the delicate balance between delivering quality software and delivering it quickly (the software paradox!) is understanding of the essential elements of the process and to follow the guidelines for tailoring the process to best fit user's project's specific needs.

Rational ClearQuest is a tool for the Defects and Changes Tracking within software development process. Different organizations follow various processes for handling software defects, enhancement requests, and other change requests. Rational ClearQuest provides a process that meets the needs of most organizations, but can also be easily customized to meet your exact process requirements. Everyone involved in the software development process needs to understand not only how changes are being implemented on an individual level, but also how these changes impact the development project as a whole. With Rational ClearQuest you can track defects and enhancement requests, assign work activities, and assess the real status of projects throughout the life cycle.

1.3 Project Scope

- 1) Learning the representing method and notation of the *Context Map* .
- 2) Getting familiar and understanding of Unified Process and Rational Unified Process framework. By modeling with Context Maps, comparing these two technologies and propose the refinement or enhancement.
- 3) Investigating and exploring functionality of the Context Map. It's not only a method and notation to representing the knowledge but also can be used as a tool to implement the functionality of other tools, such as Rational ClearQuest.
- 4) Make a conclusion for this research work and provide recommendations for future works.
- 5) A deliverables project package will contain a report and a set of Context Map.

2 Context Map Introduction

2.1 Context Map Overview

Context Map is a formal representing method for information system with a set of predefined formal notation. It consists of unlimited number of context tuples that is a generic association of set members cast in roles. In the extended spreadsheet, a column of roles and the related set members define context tuples. Graphically, context tuple is represented by a compounded edge and the connected compounded nodes. The aggregation of the context tuples forms Context Map. The Context Map allows modeling, mining and evaluation of context, processes and view of information system with generic templates and domain experts.

Through an example in the Figure 1, the Context Map's terminology and syntax will be introduced as following:

- The bold {} is called Set, such as {Phase} and {Disciplines Set/Sub Phase}
- The elements under the bold {} call Set Member, such as Inception and Elaboration.
- The capital letters in the column 1 to 5 are Set Roles, such as the letter L and N.
- The lower case letters or digits in the column are Member Roles, such as 1, 2, t, l and v.
- The columns form 1 to 5 are called context tuples.
- The column 6 is the count of the Member Roles.
- The column 7 is the count of the Set Members.

	1	2	3	4	5	239	240	241
1						237	1	{ContextTopic UniqueId}
2	1	2	3	4	5	237		Id
3						237	13	{ContextView}
4	v	v	v	v	v	5		Phases
17						70	4	{Phase}
18						66		Inception
19						67		Elaboration
20						67		Construction
21						68		Transition
22	S	S	S	S	S	70	9	{Discipline Set/Sub Phase}
23	1	1	1	1	1	14		Business Modeling
24	2	2	2	2	2	12		Requirements
25	3	3	3	3	3	12		Analysis & Design
26	4	4	4	4	4	11		Implementation
27	5	5	5	5	5	12		Test
28	6	6	6	6	6	13		Deployment
29	7	7	7	7	7	12		Configuration @ Change Management
30	8	8	8	8	8	14		Project Management
31	9	9	9	9	9	10		Enviroment
1372						237	9	{Author}
1373	v	v	v	v	v	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
1374	v	v	v	v	v	73		Map © by Yaozhong Chen 2002
1375	v	v	v	v	v	73		Map Editing WMJ

Figure 1 Context Map Example

2.2 Context Map Notation

Symbols used within this report:

- Set Roles symbols:

A - (A)ggregation of columns - context tuples

E - (E)dge properties

F - (F)low graph nodes

L - (L)flow graph with cycles

N - (N)ode properties

V - (V)alue

S - (S)equence

G - (G)uard

R - (R)esource

I - (I)dentifier

X - Cartesian Product

- **Set Member Roles symbols:**

v - marker

m - (m)iddle of 'arrow'

f - tail of 'arrow'

t - head of 'arrow'

b = f/t

f - (f)rom node

t - (t)o node

l - (l)oop

b = f/t - both nodes component

f - (f)rom node component

t - (t)o node component

l - (l)oop node component

y - yes

o - otherwise

c - (c)ompute

r - (r)ead

u - (u)pdate

d - (d)elele

x - component of Cartesian Product

numerical value

integer value

3 Rational Unified Process(RUP) Introduction

3.1 RUP Overview

The RUP has two dimensions:

- The horizontal axis represents time and shows the lifecycle aspects of the process.
It represents the dynamic aspect of the process.
- The vertical axis represents disciplines, which group activities logically by nature.
It represents the static aspect of the process.

The graph also shows how the emphasis varies over time. For example, in early iterations, we spend more time on requirements, and in later iterations we spend more time on implementation.

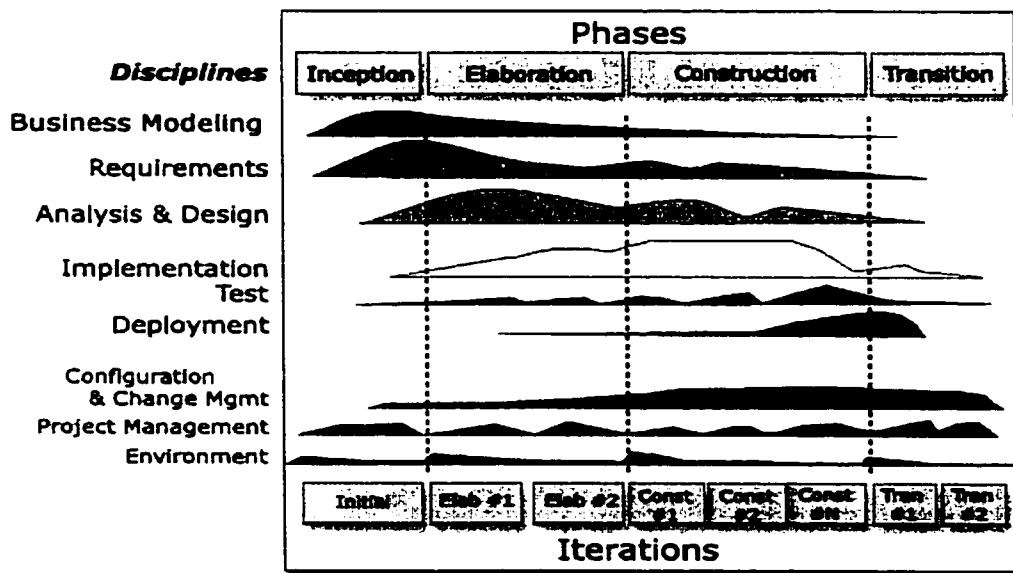


Figure 2 The Rational Unified Process Overview

3.2 RUP Concept

A process is a set of partially ordered steps intended to reach a goal; in software engineering the goal is to build a software product, or to enhance an existing one; in process engineering, the goal is to develop or enhance a process. In RUP, these are organized into a set of disciplines to further define the workflow and other process elements. Each state of the workflow consists of many activities implemented by classified developer roles. Each activity takes some inputs then produces the outputs, it's called artifacts in the RUP . The relations can be showed clearly in Figure 3.

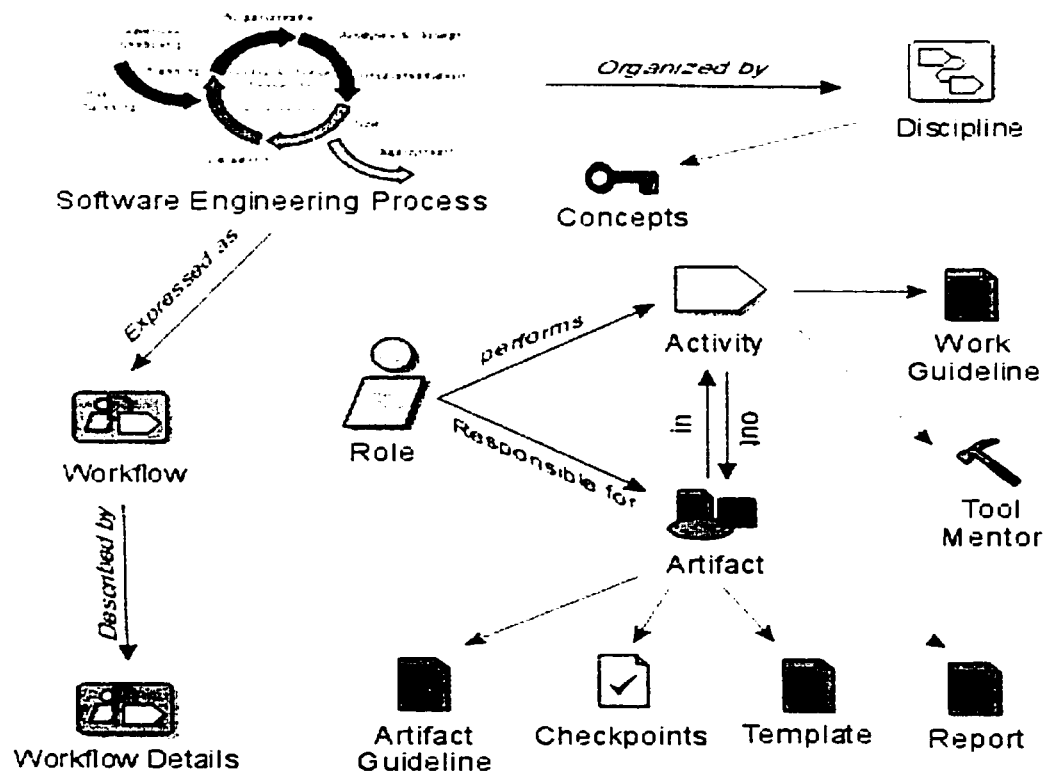


Figure 3 RUP Concepts

4 Representing RUP by Context Map

The software engineering concepts and structure in the RUP is clear, but how to find the relations among each specific elements is not easy and obvious. The RUP tool is just a web based container, all of the elements are classified and hyperlinked together. Usually, we need browse through several hyperlinks to find the relation between two elements. Even when we find the relations, RUP still can not address the question such as: in which activities a certain artifact is involved or in which stage an artifact is created or updated? All these problems can be easily addressed in the Context Map. These features will be described in the following sections.

4.1 Notation

Notations used in Figure 4 through Figure 7 are described below:

I – (I)dentifier Tuples unique ID [row 1]; the number is the unique id increased by one.

A-- (A)ggregation of columns - context tuples; v is the mark indicated the aggregation of the tuple.

S – Sequence [row 20, 163 and 276]; the number indicated the sequence of the set members.

L – Flow states, it can be f – from, t – to, l – loop

R – (R)esource , Artifacts; c—create, r—reuse, u—update, d--delete

N— (N)ode properties ,Components; v—included.

F---- (F)low graph nodes , Sub state or flow, Components; f—mark.

4.2 Schema of the RUP map

	1	6	15	22	29	35	42	50	57	66	71	74	182	239	240	241
1														237	1	(Context Tuple Unique Id)
2	1	6	15	22	29	35	42	50	57	66	71	74	182	237		id
3														237	13	(Context View)
4	V													5		Phases
5		V												9		Business Modeling
6			V											7		Requirements
7				V										7		Analysis and Design
8					V									6		Implementation
9						V								7		Test
10							V							8		Deployment
11								V						7		Configuration & Change Management
12									V					9		Project Management
13										V				5		Environment
14													V	43		RUP Artifacts Templates
17														70	4	(Phase)
22	S													70	9	(Disallowed Set/Sub-Phase)
32														9	10	(Business Modeling)
43														7	8	(Requirements)
52														7	8	(Analysis and Design)
61														6	7	(Implementation)
69														7	8	(Test)
78														8	9	(Deployment)
88														7	8	(Configuration & Change Management)
97														9	10	(Project Management)
108														5	6	(Environment)
115	R	R	R	R	R	R	R	R	R	R	R	R	R	176	15	(RUP Tools)
131														65	33	(Role)
165	S	S	S	S	S	S	S	S	S	S	S	S	S	176	112	(RUP Tools Impact Section)
278	S	S	S	S	S	S	S	S	S	S	S	S	S	176	151	(Artifact Impact Section)
430	R	R	R	R	R	R	R	R	R	R	R	R	R	233	90	(Artifact Impact Section)
521														44	27	(Component)
581														44	776	(Component Section Header)
1372														237	9	(Component Section Header)
1373	V	V	V	V	V	V	V	V	V	V	V	V	V	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
1374	V	V	V	V	V	V	V	V	V	V	V	V	V	73		Map © by Yaozhong Chen 2002
1375	V	V	V	V	V	V	V	V	V	V	V	V	V	73		Map Editing WMJ

Figure 4 RUP Context Map Schema

In Figure 4, the schema of the RUP is described as following:

- Each phase(Row 17: L) consists of a sequence of disciplines or sub phase (Row 20: S).
- Each disciplines(Row 22: A) is a work flow consisted by several states and edges (Row 32 to 108: under L).
- Each state of the disciplines includes a sequence of Scenario(Row 165: S), Activities (Row 278: S)
- Each state of the disciplines takes different artifacts(Row 430: R) as inputs and produces many outputs.
- Each state of the discipline will involve some develop roles(Row 131: N).
- Also each state of the discipline can use different tools to assistant the development.
- Each tool (Row 115: A) can be used in different scenarios within certain phases or sub phases
- Each scenario(Row 165: F) contains several activities(Row 278: N)
- The template(Row 430: F and Row 14: v) of each artifact is made of many sections(Row 581: N)
- Each artifacts contains many components (Row 521).

The detail explanations of each parts of the map are given in the following sections.

4.3 Phase and Disciplines

In Figure 5 the relations between phases and disciplines are described by an example of the state flow of the discipline(sub phase) called Business Modeling.

Each phase is a sequence of the disciplines or sub phases. Such as, Inception phase can consist of any of the sub phase (Business Modeling, Requirements, Analysis & Design etc.) in a certain order which is represented by digits between row 23 and row 29.

Business Modeling sub phase contains 10 states(Access Business Status, Describe Current Business etc.), one state can move to other states.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	239	240	241
1															237	1	(Context Tuple Unique Id)
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	237		id
3															237	13	(Context View)
4	v	v	v	v	v										5		Phases
5						v	v	v	v	v	v	v	v	v	9		Business Modeling
17															70	4	(Phase)
18						v	v	v	v	v	v	v	v	v	66		Inception
19						v	v	v	v	v	v	v	v	v	67		Elaboration
20						v	v	v	v	v	v	v	v	v	67		Construction
21						v	v	v	v	v	v	v	v	v	68		Transition
22	s	s	s	s	s										70	9	(Discipline Set/Sub Phase)
23	1	1	1	1	1	v	v	v	v	v	v	v	v	v	14		Business Modeling
24	2	2	2	2	2										12		Requirements
25	3	3	3	3	3										12		Analysis & Design
26	4	4	4	4	4										11		Implementation
27	5	5	5	5	5										12		Test
28	6	6	6	6	6										13		Deployment
29	7	7	7	7	7										12		Configuration @ Change Management
30	8	8	8	8	8										14		Project Management
31	9	9	9	9	9										10		Environment
32															9	10	(Business Modeling)
33															1		Start
34															2		Access Business Status
35															3		Describe Current Business
36															3		Identify Business Processes
37															2		Refine Business Processes Definition

247										2		Detailing a Business Use Case Using Rational RequisitePro
276	S	S	S	S	S	S	S	S	S	176	151	(Activities)
374		V		V						3		Maintain Business Rules
404							V		V	4		Review the Business Object Model
405					V					3		Review the Business Use-Case Model
409		V	V	V					V	4		Set and Adjust Goals
412					V					3		Structure the Business Use-Case Model
428	R	R	R	R	R	R	R	R	R	233	90	(Artifact)
442					U	U			U	19		Business Vision
480					U					41		Review Record
489					U				U	33		Supplementary Business Specification
491					U					7		Target-Organization Assessment
1370										237	9	(Author)
1371	V	V	V	V	V	V	V	V	V	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
1372	V	V	V	V	V	V	V	V	V	73		Map © by Yaozhong Chen 2002
1373	V	V	V	V	V	V	V	V	V	73		Map Editing WMJ

Figure 6 Detailed RUP Work Flow Context Map

In Figure 6, part of the work flow map of the discipline named business modeling and the properties of each state in the work flow are clearly displayed. The detailed workflow and properties of the rest disciplines(They are listed in the Figure 5 under {Disciplines Set/Sub Phase}) please refer to figures in the Appendix Section 1 to 9. The meaning of each element is as follows:

- At each state or node (under row 30, Access Business Status, Describe Current Business, Identify Business Processes, Refine Business Processes Definition etc.), there are many activities (under row 276, Maintain Business Rules, Review the Business Object Model etc.) been performed. After the activities are accomplished, it will move to another state.
- At each state, if we look at the row 428 to the end of the map, some artifacts might be created (member role: c), reused (r) or updated (u).
- From row 129 to 132, several roles are involved in each state to perform certain activities.

- Under row 113, there are also some tools can be chosen to enhance or accelerate development process.

4.5 RUP Tools Map

	71	72	88	89	90	91	92	93	239	240	241
115									176	15	(RUP Tools)
116	v	v							9		Rational Suite AnalystStudio
117									23		Rational ClearCase
118	v	v	v	v	v	v	v	v	20		Rational ClearQuest
131									65	33	(RUP Tools)
165									176	112	(RUP Tools Mentor Scenario)
211									1		Eliciting Stakeholder Requests Using Rational ClearQuest
212									3		Managing Stakeholder Requests Using Rational ClearQuest and Rational RequisitePro
229									2		Defining Change and Review Notifications Using Rational ClearQuest
230									2		Establishing a Change Request Process Using Rational ClearQuest
231									6		Reporting Defect Trends and Status Using Rational ClearQuest
232									5		Reporting Review and Work Status Using Rational ClearQuest
233									8		Submitting Change Requests Using Rational ClearQuest
234									3		Viewing the History of a Defect Using Rational ClearQuest
278									176	151	(Activities)
285					v				4		Assess and Advocate Quality
296						v	v		3		Confirm Duplicate or Rejected CR
322					v		v		11		Determine Test Results
346	v	v							4		Elicit Stakeholder Requests
347			v	v			v		5		Establish Change Control Process
348								v	13		Establish Configuration Management (CM) Policies
350	v								7		Find Actors and Use Cases
363								v	10		Implement Test
400					v	v		v	4		Report on Configuration Status
402						v			3		Review Change Request
430	v	v	v	v	v	v	v	v	233	90	(Artifact)
448			v	v			v	v	24		Configuration Management Plan
450					v		v		30		Deployment Model
480	v	v							20		Requirements Attributes
481	v	v					v		27		Requirements Management Plan
503					v	v	v		20		Test Log
504					v	v	v		26		Test Plan
505					v	v	v		20		Test Results
506					v	v	v		19		Test Script
1372									237	9	(Artifact)

1373	v	v						73	Syntax and Patterns © by W.M. Jaworski, 1988-2002
1374	v	v						73	Map © by Yaozhong Chen 2002
1375	v	v						73	Map Editing WMJ

Figure 6 The Rational Tools

From the Figure above, the usage of the Rational tools are described:

- Through row 115 to 234, we can find each tool is used for certain scenario. Such as Rational Suite Analyst Studio can be used to Elicit Stakeholder Requests Using Rational ClearQuest.
- From row 165 to 402, a scenario consists of several activities. The relation of the activities and tools also can be linked together.
- From row 165 to 506, a scenario will take some artifacts as inputs then produce some outputs. So the tools and the artifacts are also connected.

4.6 RUP Templates Map

	188	189	238		241	
1			238	1	{Context Tuple Unique Id}	
2	188	189	238		Id	
3			238	11	{Context View}	
14	v	v	45		RUP Artifacts Components/Templates	
430			234	90	{Artifact}	
442	v		19		Business Use-Case	
443		v	18		Business Use-Case Realizations	
521			45	59	{Components}	
539	v	v	1		Illustrations of the Workflow	
540	v	v	1		Use-Case Diagrams	
541	v	v	1		Activity Diagrams	
542	v	v	1		Relationships	
543	v	v	1		Special Requirements	
544	v	v	1		Process Owner	
545	v	v	1		estimated improvement potential	
546	v	v	1		Risk	
547	v	v	1		measurable goals or objectives of the business use case	
548	v	v	1		Performance Goals	
549		v	1		Workflow Realization	
550		v	1		Activity Diagrams	
551		v	1		Interaction Diagrams	
552		v	1		Class Diagrams	
553		v	1		Derived Requirements	
554		v	1		Trace Dependency	
581			45	776	{Template Section Title}	Note:

687	v		2		2. Business Use Case Name 2.1 Brief Description	The description should briefly
688	v		2		3. Goals	A specification of the
689	v		2		4. Performance Goals	A specification of the metrics
690	v		2		4.1 <name of performance goal>	A brief description
691	v		2		5. Workflow	
692	v		2		5.1 Basic Workflow 5.1.1 <name of workflow step>	
702	v		2		1. Introduction	
703	v		2		1.1 Purpose	
704	v		2		1.2 Scope	
705	v		2		1.3 Definitions, Acronyms and Abbreviations	
706	v		2		1.4 References	
707	v		2		1.5 Overview	
708	v		2		2. Workflow Realization	
709	v		2		3. Derived Requirements	
1372			238	9	{Author}	
1373			74		Syntax and Patterns © by W.M. Jaworski, 1988-2002	
1374			74		Map © by Yaozhong Chen 2002	
1375			74		Map Editing WMJ	

Figure 7 RUP Templates Context Map

This figure shows each artifact contains some components and has a template. The contents (Purpose, Scope etc.) of the template have been provided for creating the artifacts. The description of each sections are listed under set {Note}.

4.7 Comparison of the RUP and Context Map

In many circumstances, the software engineering process --RUP -- need to be modified, adjusted, extended and tailored to accommodate the specific characteristics, constraints and history of the adopting organization.

First of all, we need to provide a completed knowledge base of software engineering process to user. Then we can modify extend or tailor it. In the RUP Context Map there is no information lost after it is converted from RUP. All of the contents from RUP have been transferred into Context Map. So we have a completed base now.

Second, both tools can be tailored. RUP provides a tool called RUP builder to tailor it. In Context Map, we can tailor it directly on the map by removing some tuples and rows in the map.

Third, both tools can be extend. In Context Map adding of new artifacts then assign some roles to delivery those artifacts at different stage of the discipline is straightforward. What we need to do is just adding some tuples.

Fourth, it is difficult to find a entire picture of when the artifacts is created, how many stages is involved to act with the artifact and when it is finished. Furthermore, some artifacts are used as inputs at some stage of the process in the RUP, but how to find when it is created. In the Context Map, we can easily address these questions and also can check the consistency of the artifact's lifecycle. It's easy as selecting the artifacts and then watching the member roles of the artifact; if there is no "c" in the same row, that means the artifact has not been created. If there are no entries in the row, that means the artifact is not been used, and so on.

Last, I have not found the way to track the status of the whole project, each activities or each artifacts using RUP. Although Rational has many tracking tools, such as ClearQuest which we will discuss within the following sections, they only can track specific items that are defined separately in the ClearQuest. It's also difficult to get the status of the project and certain artifacts within the same process view. Context Map can address these problems. The answer can be found at the end of this paper.

5 Rational ClearQuest(RCQ) Introduction

5.1 ClearQuest Overview

Rational ClearQuest is a tool for Defect and Change Tracking within software development process. With Rational ClearQuest you can track defects and enhancement requests, assign work activities, and assess the real status of projects throughout the life cycle. It also can be easily customized to exact process requirements from different organization. In the following section, taking a sample defect tracking system as an example, how the ClearQuest works will be introduced.

5.2 Representing of Defect Tracking System Schema in ClearQuest

Suppose we find a defect during a development, then we will need to analyze it and assign it to somebody to fix the defect. After the person fixed the problem, someone will evaluate it and decide if we can close this issue or not. This is the scenario of this sample system. To see the Figure 8 below, we need login to another ClearQuest tool called ClearQuest Designer to design that matrix. From the matrix in the Figure 8, we can clearly know how an issue is raised and moved from one state to another state by executing some actions. Submitted, Assigned etc. are states. The actions are Assign, Open, Close and so on. The properties or constrains of each action are defined in another matrix shown in the Figure 9. The action types are defined in the second column called type. All these types are predefined and fixed in ClearQuest. Some actions can be used to

move a issue raised in the ClearQuest from one state to another state. Some actions are just being performed within a certain state. The constrains are defined in the last column called Access Control. For example, the action submit can be done by any user. There are also other groups of user can be predefined within ClearQuest to associate with each action. There are also other matrixes existed in ClearQuest for the properties of each issue(called defect in this example) and some hooks or scripts can be used to define more sophisticated constrains or functions. We are not going to introduce them in detail here. Our purpose is just to compare two kinds of the representing methods.

To \ From	Submitted	Assigned	Opened	Resolved	Closed	Duplicate	Postponed	
Submitted								
Assigned	Assign						Assign	
Opened		Open		Reject	Re_open			
Resolved			Resolve					
Closed	Close	Close		Validate			Close	
Duplicate	Duplicate		Duplicate	Duplicate	Duplicate			
Postponed	Postpone	Postpone	Postpone					

Figure 8 State transition Matrix in ClearQuest

Action Name	Case State Type	Access Control	
Submit	SUBMIT	All Users	
Assign	CHANGE_STATE	All Users	
Open	CHANGE_STATE	All Users	
Resolve	CHANGE_STATE	All Users	
Validate	CHANGE_STATE	All Users	
Reject	CHANGE_STATE	All Users	
Re_open	CHANGE_STATE	All Users	
Close	CHANGE_STATE	All Users	
Duplicate	DUPLICATE	All Users	
Unduplicate	UNDUPLICATE	All Users	
Postpone	CHANGE_STATE	All Users	
Modify	MODIFY	All Users	
Delete	DELETE	All Users	
Import	IMPORT	All Users	
Init_Note_Entry	BASE	All Users	B.
Send_Email_Notif	BASE	All Users	

Figure 9 Action Access Control Matrix in ClearQuest

5.3 Query in ClearQuest

A query tool coming with ClearQuest is used to retrieve data. The query is built by ClearQuest Query Editor shown in the Figure 10. It is a graphic translation from SQL “Select” statement. It contains most of the functioning of SQL “Select” statement.

The result is viewed as the format shown in the Figure 11. If you need more friendly and sophisticated view format, another reporting tool called Crystal Report need to be installed with ClearQuest as a separate application. Then the view can be freely customized.

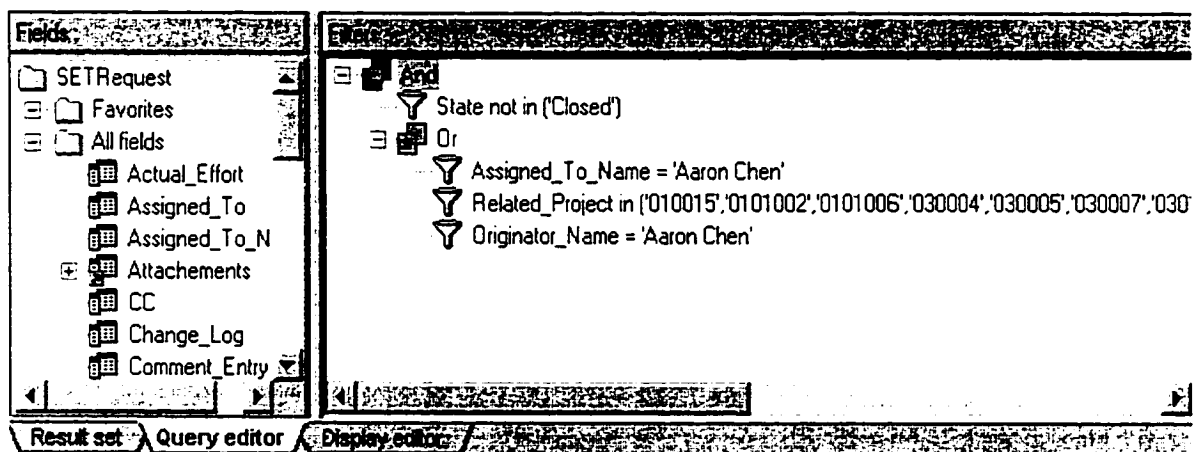


Figure 10 Query tool of ClearQuest

	id	State	Headline	Priority	Project
▶	SAMPL00000001	Opened	spelling error in login screen	3-Normal Queue	Classics
-	SAMPL00000002	Resolved	sales tax incorrect if item deleted from purchase	2-Give High Attention	Classics
-	SAMPL00000003	Resolved	cancel sale doesn't correctly repaint screen	3-Normal Queue	Classics
-	SAMPL00000004	Resolved	want more help on inventory report	3-Normal Queue	Classics
-	SAMPL00000005	Resolved	columns out of alignment	2-Give High Attention	Classics
-	SAMPL00000006	Opened	delete item not working correctly	2-Give High Attention	Classics
-	SAMPL00000007	Resolved	override price does not work	2-Give High Attention	Classics

Result set / Query editor / Display editor

Main	Notes	Resolution	Attachments	History	Customer
ID:	SAMPL00000001		State:	Opened	
Headline:	spelling error in login screen				
Project:	Classics		Keywords:		
Severity:	3-Average		Symptoms:		
Priority:	3-Normal Queue				
Owner:	lead				
Description:					

Apply
Revert
Print Record
Actions

Figure 11 Records displayed in ClearQuest

6 Representing RCQ by Context Map

6.1 RCQ Schema representing in Context Map

The Figure 12 and Figure 13 show schema of the defect tracking system in the ClearQuest.

From column 1 to 25, the definitions of the schema are as following. The detailed and expanded map please refer to Figure 14 in the next section.

- Under row 8, L: Each defect issue has several predefined states(Started, Submitted, Assigned, Opened, Resolved, Closed, Duplicated, Postponed)
- Under row 17, E: The actions (Submit, Assign, Open, Resolve, Validate, Reject, Re_open, Close, Duplicate, Unduplicate, Postpone, Modify, Delete) are defined as edges in the state machine. The defect issue states will be changed by performing those actions. Under row 31, G: The constrain is set up in terms of user groups, which is also the same as the constrain in the ClearQuest.
- Under row 36 , S: It is used to track the sequence or history of the state changes of each issue. In the Figure 13, the meaning of the sequence is obviously. The state history of the issue SAMPL0000002 is flowing in the order as start→submitted→Assigned→Opened→Resolved.

From column 26 to 78, the defect is recorded: (Detail refers to Figure 14 in next section)

- Row 36, I: The unique defect id is recorded under I.
- Row 77 to 293, N: The properties of each defect is recorded.
- Row 205, I: Other entities called customers are recorded.
- Row 293, I: Entities named projects are recorded.

6.2 Data Stored in Context Map

By Context Query on the whole map, two sample defect records stored in Context Map are shown in the Figure 14.

They contain three entities: Defect(its unique id is under Row 36), Customer(its unique id is under 205) and Project(its unique id is under 293). All three entities can be many-to-many relationship. To reduce the space, each entity only need to be populated once, then use their unique id or name as a foreign key import to other entities.

Column 26 represents one sample record. Its unique id is SAMPLE000000001 and all of its properties are under each row marked by "N". For example, this defect's headline is *spelling error in login screen*; the description is *SAMPL000000001 spelling error in login screen ClassicsPOS*; Priority is *3-Normal Queue*; Customer name is: *John Smith*; Project Name is *Project Account Audit*.

From row 37 and 38, we have mentioned in the previous section it can be used to track the state change history using the numbers.

	1	3	6	8	12	17	24	26	27			
1										78	1	(Context Tuple Id)
2	1	3	6	8	12	17	24	26	27	78		Id
3										78	4	(Context View)
4	V	V	V	V	V	V	V			25		Histroy
5										40		Defects
8										65	8	(State)
9										2		Started
10										24		Submitted
11										6		Assigned
12										11		Opened
13										22		Resolved
14										7		Closed
16										7		Postponed
17	E	E	E	E	E	E	E			25	13	(Action)
18	Submit									1		Submit
19		Assign								2		Assign
20				Open						1		Open
21					Resolve					1		Resolve
22						Validate				1		Validate
25							Close			3		Close
28										3		Postpone
31	G	G	G	G	G	G	G			25	4	(Duration)

32										18	Administrator
33										11	Issue Coordinator
34										1	Submitter
35										6	Assignee(as)
36	S	S	S	S	S	S	S			65 40	Defect ID: 40
37	1	2	3				4			5	SAMPL00000001
38	1	2		3	4	5				6	SAMPL00000002
77										40 40	Headline
78										1	spelling error in login screen
79										1	sales tax incorrect if item deleted from purchase
118										40 11	Description: Defect ID: 40
119										1	SAMPL00000001 spelling error in login screen
120										1	ClassicsPOS
130										40 4	SAMPL00000002 sales tax incorrect if item
132										9	deleted from purchase ClassicsPOS
133										12	2-Give High Attention
135										40 5	3-Normal Queue
136										5	Severity: 5
138										16	1-Critical
141										40 5	3-Average
147										40 4	Submitter: 40
148										10	Submit Date: 9/1/98
152										40 6	Owner: 6
156										7	Lead
176										40 12	Symptoms: 12
191										40 6	Resolution: 6
198										40 6	Resolution: 6
205										50 10	Customer: 10
206										2	Customer Name: John Smith
207										2	Ethan Hunt
293										43 3	Project name: 3
294										37	ClassicsPOS new01
296										3	Project Account Audit
301										78 2	Author: 2
302	V	V	V	V	V	V	V	V	V	78	Syntax and Patterns © by W.M. Jaworski, 1988-2002
303	V	V	V	V	V	V	V	V	V	78	Map © by Yaozhong Chen 2002

Figure 14 Defect Records Stored in Context Map

6.3 Context query

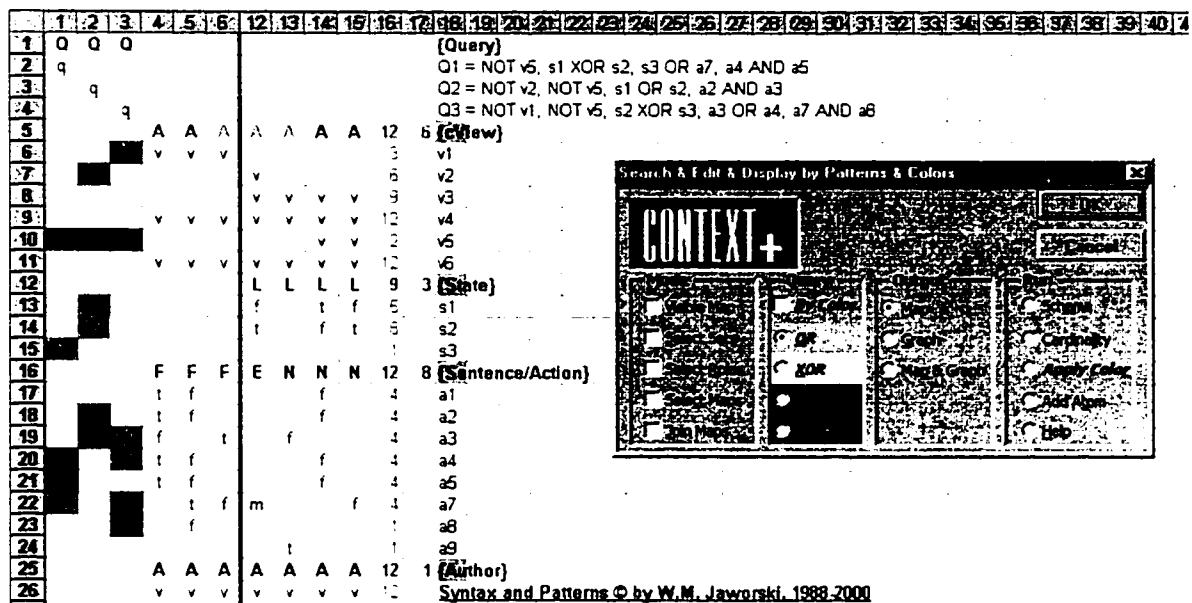


Figure 15 Context Query

Context Map also provides a set of tools. Context Query is one of them. The unique and prominent character of the Context Query is query directly on the data. It's not necessary to build predefined query tree or statement first, then start to query in the database. For simple query, you just need to select the combination of the set members then choose an option on the pop up query window. If you like to record query, you can save it as column(Examples are in column 1 to 3). The different colors represent the different logical operations. Green represents OR; Yellow represents XOR; Red represents AND; Blue represents NOT. Up to now, the most of the functions of the query tool have been implemented and tested by Mr. Kang Zhou, instructed by Dr. W. M. Jaworski, but have not been finalized. Details may be found at <http://www.gen-strategies.com>.

6.4 Further investigating on other functions and comparison summary of both defect tracking tools

From the previous examples, we can see that most of the functions in the ClearQuest have been included in the Context Map without any additional coding. For example, both can track the history, both come with several tools to manipulate data and design the schema etc.

Beside, the unique context query tools is more flexible than ClearQuest query tool. It can query directly on the data.

Furthermore, from easy understanding point of view, a whole picture of the defect tracking system is more easily understood and visualized in the Context Map. In ClearQuest, we need to go through several matrixes to get a whole picture. In Context Map, everything is in one map.

Also, we can define more sophisticated schema within the same map. For example, a property of the entity can only be changed by certain users at certain state. In ClearQuest, this schema need to be hidden in handles or scripts. Therefore, they can not be defined visually. On the contrary, to define this constraint is straightforward in the Context Map. In addition, the tracking on the state can also extend to other properties using the same method described in the previous section.

Last, the schema in ClearQuest is not easily reused in different system (Such as the schema of defect tracking system is not easily reused to a kind of helpdesk tracking system, even when they have very similar schemas.) and impossible to be merged to other schemas(Such as merge it into RUP). But within Context Map, we only need

modify some property names, then one system is change to another system. Also we just need attach the tracking schema to each activity or each artifact in the RUP Context Map, then we can track the status of the activity or artifact. These are other prominent characters of the Context Map merging to other schema and reuse it. Here, we address the issue raised in the section 4.7. Any element in the RUP Context Map can be tracked using the same schema as in the Figure 13.

7 Conclusion And Recommendation

7.1 General Conclusion

After we explored these different methodologies, many merits of the Context Map are revealed. They are summarized below:

- Machine readable: The matrix is a kind of format which is machine readable and it is very easy to transfer to other formats.
- Reuse: Many elements and schema can be reused by a little modification or joining the different schema maps.
- Formal: The notations have clear meaning; simple syntax allows it to become very easy to be understood, and very easy to be changed.
- Extendable on notations: Even we have already had a set of predefined notation, it is also allowed user to create their own notations to meet new needs.
- Consistency checking: The cardinality capability combining with well defined notations, the consistence checking on the schema becomes possible.
- Extendable on size: If the data reach the limit of each sheet in Excel, it is can be transfer to any kind of the commercial database, such as MS Access or MS SQL. This capability has been implemented by Minghui Han who was a master student in Concordia University. [4]
- Flexible: Due to the simplicity of the syntax, there are always many different ways to representing the same knowledge. It is also becoming a challenge to find the easiest and simplest schema to represent the knowledge.

- Advanced query: Along with Context query tools, query on the high level design schema is becoming possible. And directly querying on the data allows query more easy and visible.

In conclusion, Context Map can be treat as a kind of common visually modeling language, this attribute let it become a good choice for design layer of the CASE Tools. It can reduce a lot of knowledge learning, knowledge transfer from one format to another and representing efforts.

7.2 Recommendations for Future Works

After the investigation, there are still some improvements and detailed works need to be done to refine and enhance this methodology. Also some interesting works are recommended as following:

- Refine the ambiguity concepts, elements or set members within the RUP Context Map; then, combine tracking schema with RUP schema in order to provide the status tracking and quantified statistic for the RUP; finally, it can be developed to become a commercial process controlling tool.
- When the data becomes larger and larger, split data in one sheet into several sheets or transfer the data from Excel to another Commercial databases.
- Adding the constraint to each field in the Defect tracking map and generating user preferred interface, such as forms or windows, for viewing the contents in the map.

- Inputting the data in the map is very boring and there are some redundancies existing. Provide a guideline for design the schema, composing the map and inputting the data.
- Transferring the knowledge to the map is still done by domain experts, that means it is still done by manually. Maybe there are some automatic transferring can be accomplished in the near future.

Bibliography

- 1) W.M. Jaworski, all of the contents located on the General Strategies Inc web page
- 2) Jaworski, W.M. et al. (1987). The ABL/W4 methodology for system modeling.
System Research Journal 4(1), 23-37.
- 3) W.M. Jaworski, et al. "Representing processes, schemata and templates with *jMaps*",
Semiotica 125(1/3), 229-47, 1999.
- 4) Minghui Han, "Associative Data Model And Context Maps", Major Report.
Concordia University, 2001.
- 5) Ian Sommerville, "Software Engineering", Addison-Wesley, 5th edition, 1995.
- 6) Rational Unified Process Menu. <http://www.rational.com>.
- 7) Rational ClearQuest Menu. <http://www.rational.com>.
- 8) Rational, <http://www.rational.com>.
- 9) Concordia University, Thesis preparation and thesis examination regulations.
http://www-gradstudies.concordia.ca/SGS_WWW/publications.html.
- 10) Rob Kremer, A Concept Map Meta-Language,
<http://www.cpsc.ucalgary.ca/~kremer/dissertation/index.html>.

Appendix

Due to the RUP Context Map is too large to be viewed in one page, it is split into several Figures attached below:

1. Discipline— Business Modeling View

															237	1	{Context Tuple Unique Id}
6	7	8	9	10	11	12	13	14							237		id
															237	13	{Context View}
v	v	v	v	v	v	v	v	v	v	v					9		Business Modeling
															70	4	{Phase}
v	v	v	v	v	v	v	v	v	v	v					66		Inception
v	v	v	v	v	v	v	v	v	v	v					67		Elaboration
v	v	v	v	v	v	v	v	v	v	v					67		Construction
v	v	v	v	v	v	v	v	v	v	v					68		Transition
															70	9	{Discipline Set/Sub-Phase}
v	v	v	v	v	v	v	v	v	v	v					14		Business Modeling
															9	10	{Business Modeling}
															1		Start
															2		Access Business Status
															3		Describe Current Business
															3		Identify Business Processes
															2		Refine Business Processes Definition
															2		Design Business Processes Realizations
															2		Refine Roles and Responsibilities
															3		Explore Business Automation
															3		Develop a Domain Model
															4		End
R	R	R	R	R	R	R	R	R	R	R					176	15	{RUP Tools}
															9		Rational Suite AnalystStudio
															28		Rational RequisitePro
															56		Rational Rose Tool Set
															65	33	{Role}
															6		Business Designer
															3		Business-Model Reviewer
															8		Business-Process Analyst
															2		Stakeholder
S	S	S	S	S	S	S	S	S	S	S					176	112	{RUP Tools/Memory Scenario}
															4		Finding Actors and Use Cases Using Rational Rose
															10		Publishing Web-based Rational Rose Models Using Web Publisher
															3		Detailing Business Workers and Entities Using Rational Rose
															2		Finding Business Actors and Use Cases Using Rational Rose

2. Discipline—Requirements View

							237	1	{Context Table Unique Id}
15	16	17	18	19	20	21	237		id
							237	13	{Context View}
v	v	v	v	v	v	v	7		Requirements
							70	4	{Phase}
v	v	v	v	v	v	v	66		Inception
v	v	v	v	v	v	v	67		Elaboration
v	v	v	v	v	v	v	67		Construction
v	v	v	v	v	v	v	68		Transition
							70	9	{Discipline Set/Sub Phase}
v	v	v	v	v	v	v	12		Requirements
							7	8	{Requirements}
							1		Start
							3		Analyze the Problem
							4		Understand Stakeholder Needs
							2		Define the System
							3		Manage the Scope of the System
							2		Refine the System Definition
							2		Manage Changing Requirements
							2		End
R	R	R	R	R	R	R	176	15	{RUP Tools}
	v		v		v		9		Rational Suite AnalystStudio
	v		v			v	20		Rational ClearQuest
	v	v	v	v	v	v	28		Rational RequisitePro
						v	10		Rational Robot
						v	16		Rational TestFactory
						v	15		Rational TestManager
	v		v		v	v	56		Rational Rose Tool Set
							65	33	{Role}
						v	1		Requirements Reviewer
						v	1		Requirements Specifier
	v	v	v	v		v	6		System Analyst
						v	3		User-Interface Designer
						v	6		Software Architect
		v				v	4		Change Control Manager
S	S	S	S	S	S	S	176	112	{RUP Tools, Methods, Scenarios}
						v	2		Detailing a Use Case Using Rational Rose
	v		v				4		Finding Actors and Use Cases Using Rational Rose
						v	10		Publishing Web-based Rational Rose Models Using Web Publisher
						v	2		Structuring the Use-Case Model Using Rational Rose
	v		v				4		Detailing a Business Use Case Using Rational Rose
						v	10		Finding Actors and Use Cases Using Rational Rose
	v		v				3		Managing Stakeholder Requests Using Rational ClearQuest and Rational RequisitePro

V		V		V		7		Managing Use Cases Using Rational Rose and Rational RequisitePro
					V	5		Reporting Review and Work Status Using Rational ClearQuest
V		V				7		Capturing a Common Vocabulary Using Rational RequisitePro
					V	2		Detailing a Use Case Using Rational RequisitePro
V	V	V	V			5		Developing a Vision Using Rational RequisitePro
V	V	V	V		V	6		Managing Dependencies Using Rational RequisitePro
					V	2		Reviewing Requirements Using Rational RequisitePro
					V	4		Evaluating Test Coverage Using Rational TestFactory
					V	3		Evaluating the Results of Executing a Test Suite Using Rational TestFactory
					V	4		Using Rational TestFactory to Measure and Evaluate Code-based Test Coverage on Rational Robot Test Scripts
					V	3		Viewing Logs and Evaluating the Execute Test Suite Using the Rational TestManager
S	S	S	S	S	S	S	176	151 (Activities)
V	V	V					4	Capture a Common Vocabulary
					V		4	Detail a Use Case
					V		5	Detail the Software Requirements
V	V						3	Develop Requirements Management Plan
V	V	V	V				5	Develop Vision
	V						4	Elicit Stakeholder Requests
V	V	V					7	Find Actors and Use Cases
		V	V			V	4	Manage Dependencies
					V		1	Model the User-Interface
					V		2	Prototype the User-Interface
					V		3	Review Change Request
					V		4	Review Requirements
					V		2	Structure the Use-Case Model
R	R	R	R	R	R	R	233	90 (Artifact)
							8	Boundary Class
							18	Business Glossary
							24	Business Rules
			U				18	Business Use-Case
			U				1	Business Use-Case Realizations
				U			45	Change Request
		U					27	Glossary
			U	U	U		20	Requirements Attributes
			U	U	U		27	Requirements Management Plan
							27	Software Architecture Document
						U	17	Software Requirements Specification
		U		U			13	Stakeholder Requests
		U					33	Supplementary Business Specification
				U			18	Supplementary Specifications
			U		U	U	52	Use-Case Model(Specification)
							21	Use-Case Storyboard
							14	User-Interface Prototype
			U	U			31	Vision

								237	9	{Author}
v	v	v	v	v	v	v	v	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
v	v	v	v	v	v	v	v	73		Map © by Yaozhong Chen 2002
v	v	v	v	v	v	v	v	73		Map Editing WMJ
v	v	v	v	v	v	v	v	113		Map © by Yaozhong Chen 2002/02/18

3. Discipline— Analysis and Design view

								237	1	{Context Tuple Unique Id}
22	23	24	25	26	27	28		237		id
								237	13	{Context View}
v	v	v	v	v	v	v	v	7		Analysis and Design
								70	4	{Phase}
v	v	v	v	v	v	v	v	66		Inception
v	v	v	v	v	v	v	v	67		Elaboration
v	v	v	v	v	v	v	v	67		Construction
v	v	v	v	v	v	v	v	68		Transition
								70	9	{Discipline Set/Sub Phase}
v	v	v	v	v	v	v	v	12		Analysis & Design
								7	8	{Analysis and Design}
								1		Start
t								2		Define a Candidate Architecture
t								2		Perform Architectural Synthesis
t	t							3		Refine the Architecture
t	t							3		Analyze Behavior
								2		Design Components
								2		Design the Database
								4		End
R	R	R	R	R	R	R	R	176	15	{RUP Tools}
			v					23		Rational ClearCase
					v	v		15		Rational Purify
			v		v	v		16		Rational TestFactory
	v	v	v	v	v	v	v	56		Rational Rose Tool Set
								65	33	{Role}
		v						3		User-Interface Designer
			v					1		Architecture Reviewer
					v			2		Code Reviewer
						v		1		Database Designer
					v	v	v	3		Design Reviewer
	v				v	v	v	4		Designer
					v	v	v	4		Implementer
					v			8		Integrator
	v	v	v	v	v			6		Software Architect
S	S	S	S	S	S	S	S	176	112	{RUP Tools/RealTime/Scenarios}
			v					2		Designing with Active Objects in Rational Rose RealTime
			v					5		Accessing Rational ClearCase from Rational Rose

		V	V		V			4		Capturing the Results of Use-Case Analysis Using Rational Rose
		V	V					3		Creating Use-Case Realizations Using Rational Rose
							V	2		Designing and Modeling Databases Using Rational Rose Data Modeler
			V					2		Documenting the Deployment Model Using Rational Rose
			V					2		Documenting the Process View Using Rational Rose
					V	V		4		Generating Elements from a Model Using Rational Rose
			V		V	V		4		Managing Classes Using Rational Rose
	V			V	V			4		Managing Collaboration Diagrams Using Rational Rose
			V					2		Managing Interfaces Using Rational Rose
	V			V	V			4		Managing Sequence Diagrams Using Rational Rose
			V		V			3		Managing Subsystems Using Rational Rose
	V		V	V	V			5		Managing the Design Model Using Rational Rose
	V	V		V	V	V		10		Publishing Web-based Rational Rose Models Using Web Publisher
			V					3		Structuring the Implementation Model Using Rational Rose
	V	V		V	V	V		10		Finding Actors and Use Cases Using Rational Rose
			V					5		Setting Up the Implementation Model with UCM Using Rational ClearCase
			V					6		Updating Your Project Work Area Using Rational ClearCase
					V	V		4		Profiling Java Memory Usage Using Rational Purify and Rational PurifyPlus (Windows)
					V	V		4		Profiling Memory Usage in Managed Code Using Rational Purify® and Rational PurifyPlus (Windows)
					V	V		4		Implementing Generated Test Scripts Using Rational TestFactory
			V		V	V		6		Structuring the Test Implementation with Rational TestFactory
			V		V	V		6		Setting Up the Test Environment in Rational TestFactory
S	S	S	S	S	S	S	S	176	151	(Activities) - 176
		V	V					6		Architectural Analysis
			V					1		Assess Viability of Architectural Proof-of-Concept
					V	V		3		Class Design
			V					1		Construct Architectural Proof-of-Concept
						V		2		Database Design
			V					2		Describe Distribution
			V					3		Describe the Run-time Architecture
	V	V						3		Develop Design Guidelines
			V					2		Develop Programming Guidelines
			V					5		Identify Design Elements
			V					2		Identify Design Mechanisms
					V	V		9		Implement Component
					V			2		Incorporate Existing Design Elements
					V			2		Plan System Integration
		V	V					2		Prioritize Use Cases
		V						2		Prototype the User-Interface
					V			2		Review Code
			V					1		Review the Architecture

					v	v	v	5		Review the Design
				v				8		Structure the Implementation Model
						v		3		Subsystem Design
	v				v			6		Use-Case Analysis
						v		4		Use-Case Design
B	B	B	B	B	B	B	B	233	90	{Artifact} {Phase}
					u			23		Analysis Model
								5		Architectural Proof-of-Concept
								10		Business Case
			u			u	u	45		Change Request
								11		Data Model
			u					30		Deployment Model
		u	r	u	u	u		43		Design Model...
								27		Glossary
			u		u	u		41		Review Record
	u	B	u					27		Software Architecture Document
								33		Supplementary Business Specification
								18		Supplementary Specifications
								52		Use-Case Model(Specification)
								14		User-Interface Prototype
		u						31		Vision
								237	9	{Artifact} {Phase}
v	v	v	v	v	v	v	v	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
v	v	v	v	v	v	v	v	73		Map © by Yaozhong Chen 2002
v	v	v	v	v	v	v	v	73		Map Editing WMJ
v	v	v	v	v	v	v	v	113		Map © by Yaozhong Chen 2002/02/18

4. Discipline— Implementation view

29	30	31	32	33	34	239	240	241		
						237	1	{Context Tuple Unique Id}		
29	30	31	32	33	34	237		id		
						237	13	{Context View}		
v	v	v	v	v	v	6		Implementation		
						70	4	{Phase}		
v	v	v	v	v	v	66		Inception		
v	v	v	v	v	v	67		Elaboration		
v	v	v	v	v	v	67		Construction		
v	v	v	v	v	v	68		Transition		
						70	9	{Discipline Sub Discipline}		
v	v	v	v	v	v	11		Implementation		
						6	7	{Implementation}		
						1		Start		
						2		Structure the Implementation Model		
						2		Plan the Integration		
						2		Implement the Components		
						2		Integrate Each Subsystem		
						2		Integrate the Subsystem		

						3		End
R	R	R	R	R	R	176	15	(RUP Tools)
	V	V				23		Rational ClearCase
			V			15		Rational Purify
	V		V			16		Rational TestFactory
	V		V	V	V	56		Rational Rose Tool Set
						65	33	(Role)
			V			2		Code Reviewer
			V			4		Implementer
		V	V	V	V	8		Integrator
	V					6		Software Architect
S	S	S	S	S	S	176	112	(RUP Tools/Model/Scenario)
	V					5		Accessing Rational ClearCase from Rational Rose
				V	V	3		Comparing and Merging Rational Rose Models Using Model Integrator
			V			4		Generating Elements from a Model Using Rational Rose
	V					3		Structuring the Implementation Model Using Rational Rose
		V				4		Setting Policies Using Rational ClearCase
	V					5		Setting Up the Implementation Model with UCM Using Rational ClearCase
	V					6		Updating Your Project Work Area Using Rational ClearCase
			V			4		Profiling Java Memory Usage Using Rational Purify and Rational PurifyPlus (Windows)
			V			4		Profiling Memory Usage in Managed Code Using Rational Purify® and Rational PurifyPlus (Windows)
			V			3		Executing a Test Suite Using Rational TestFactory
			V			4		Implementing Generated Test Scripts Using Rational TestFactory
	V		V			6		Structuring the Test Implementation with Rational TestFactory
	V		V			6		Setting Up the Test Environment in Rational TestFactory
S	S	S	S	S	S	176	151	(Activities)
			V			1		Fix a Defect
			V			9		Implement Component
			V			2		Implement Test Components and Subsystems
				V		2		Integrate Subsystem
				V	V	3		Integrate System
			V			2		Plan System Integration
		V				4		Promote Baselines
			V			2		Review Code
	V					8		Structure the Implementation Model
R	R	R	R	R	R	233	90	(Artifact)
					U	5		Build
						10		Component
						43		Design Model...
						9		Implementation Model
					U	11		Implementation Subsystem
					U	11		Integration Build Plan
						22		Iteration Plan
					U	41		Review Record
						27		Software Architecture Document
						17		Test Component
						237	9	(Author)
V	V	V	V	V	V	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
V	V	V	V	V	V	73		Map © by Yaozhong Chen 2002
V	V	V	V	V	V	73		Map Editing WMJ
V	V	V	V	V	V	113		Map © by Yaozhong Chen 2002/02/18

5. Discipline— Test view

35	36	37	38	39	40	41	239	240	241
							237	1	(Context) Tools/Utilities (id)
35	36	37	38	39	40	41	237		id
							237	13	(Context) Tools/Utilities (id)
V	V	V	V	V	V	V	7		Test
							70	4	(Phase) Inception
V	V	V	V	V	V	V	66		Inception
V	V	V	V	V	V	V	67		Elaboration
V	V	V	V	V	V	V	67		Construction
V	V	V	V	V	V	V	68		Transition
							70	9	(Discipline) Set/Sub Phase
V	V	V	V	V	V	V	12		Test
							7	8	(Test) Start
							1		Start
							2		Define Evaluation Mission
							2		Verify Test Approach
							3		Validate Build Stability
							2		Test and Evaluate
							2		Achieve Acceptable Mission
							3		Improve Test Assets
							2		End
R	R	R	R	R	R	R	176	15	(RUP Tools) Rational ClearQuest
	V	V		V	V	V	20		Rational ClearQuest
	V	V		V	V	V	6		Rational PureCoverage
	V	V		V	V	V	15		Rational Purify
				V	V	V	3		Rational QualityArchitect
	V	V		V	V	V	5		Rational Quantify
					V	V	10		Rational Robot
					V	V	5		Rational Test RealTime
					V	V	16		Rational TestFactory
	V	V		V	V	V	15		Rational TestManager
							65	33	(Role) Test Analyst
	V	V	V	V	V	V	6		Test Analyst
	V	V		V		V	4		Test Designer
		V	V	V	V	V	5		Tester
	V	V	V	V	V	V	5		Test Manager
S	S	S	S	S	S	S	176	112	(RUP Tools/Memory/Scenario) Reporting Defect Trends and Status Using Rational ClearQuest
	V	V		V	V	V	6		Reporting Defect Trends and Status Using Rational ClearQuest
				V		V	5		Reporting Review and Work Status Using Rational ClearQuest
	V	V		V	V	V	8		Submitting Change Requests Using Rational ClearQuest
	V	V		V	V	V	5		Evaluating Code Coverage Using Rational PureCoverage and Rational PurifyPlus (UNIX)
	V	V		V	V	V	5		Evaluating Code Coverage Using Rational PureCoverage and Rational PurifyPlus (Windows)
	V	V		V	V	V	6		Detecting Run-Time Errors Using Rational Purify and Rational PurifyPlus (UNIX)
	V	V		V	V	V	6		Detecting Run-time Errors Using Rational Purify and Rational PurifyPlus (Windows)
					V	V	3		Implementing an Automated Component Test using Rational QualityArchitect
	V	V		V	V	V	5		Finding Performance Bottlenecks Using Rational Quantify and Rational PurifyPlus
					V	V	3		Creating an Automated Performance Test Script Using Rational Robot

6. Discipline— Deployment view

42	43	44	45	46	47	48	49	239	240	241
								237	1	(Content Tuple Unique Id)
42	43	44	45	46	47	48	49	237		id
								237	13	(Content View)
v	v	v	v	v	v	v	v	8		Deployment
								70	4	(Phase)
v	v	v	v	v	v	v	v	66		Inception
v	v	v	v	v	v	v	v	67		Elaboration
v	v	v	v	v	v	v	v	67		Construction
v	v	v	v	v	v	v	v	68		Transition
								70	9	(Discipline Set/Sub Phase)
v	v	v	v	v	v	v	v	13		Deployment
								8	9	(Deployment)
								1		Start
								2		Plan Deployment
								2		Develop Support Material
								2		Produce Deployment Unit
								3		Beta Test Product
								2		Manage Acceptance Test (At Development Site)
								2		Package Product
								2		Provide Access to Download Site
								3		End
								176	15	(RUP Tools)
								65	33	(Role)
			v					4		Implementer
	v		v	v	v	v	v	6		Deployment Manager
		v						1		Course Developer
						v		1		Graphic Artist
		v						1		Technical Writer
s	s	s	s	s	s	s	s	176	112	(RUP Tools Mentors/Scenario)
s	s	s	s	s	s	s	s	176	151	(Activities)
						v		1		Create Product Artwork
	v							1		Define Bill of Materials
	v							1		Develop Deployment Plan
			v					1		Develop Installation Artifacts
		v						1		Develop Support Materials
		v						1		Develop Training Materials
					v			1		Manage Acceptance Test
					v			1		Manage Beta Test
							v	1		Provide Access to Download Site
							v	1		Release to Manufacturing
							v	1		Verify Manufactured Product
							v	1		Write Release Notes
								233	90	(Activities)
								6		Bill of Materials
								5		Build
								45		Change Request
								8		Deployment Plan
								7		Deployment Unit
								2		End-User Support Material

								1		Installation Artifacts
								22		Iteration Plan
								3		Manual Styleguide
								1		Product (The Product Build)
								3		Product Acceptance Plan
								1		Product Artwork
								2		Release Notes
								16		Software Development Plan
								17		Software Requirements Specification
								20		Test Environment Configuration
								20		Test Results
								2		Training Materials
								14		User-Interface Prototype
								237	9	Authoring Tools
								73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
								73		Map © by Yaozhong Chen 2002
								73		Map Editing WMJ
								113		Map © by Yaozhong Chen 2002/02/18

7. Discipline— Configuration & Change Management view

50	51	52	53	54	55	56	239	240	241
							237	1	{Context Tuple Unique Id}
50	51	52	53	54	55	56	237		id
							237	13	{Context View}
V	V	V	V	V	V	V	7		Configuration & Change Management
							70	4	{Phase}
V	V	V	V	V	V	V	66		Inception
V	V	V	V	V	V	V	67		Elaboration
V	V	V	V	V	V	V	67		Construction
V	V	V	V	V	V	V	68		Transition
							70	9	{Discipline Set/Sub Phase}
V	V	V	V	V	V	V	12		Configuration & Change Management
							7	8	{Configuration/Change Management Process}
							1		Start
							2		Plan Project Configuration and Change Control
							2		Create Project CM Environments
							2		Change and Deliver Configuration Items
							2		Manage Baselines and Releases
							2		Manage Change Requests
							2		Monitor and Report Configuration Status
							4		End
							176	15	{Rational Tool Set}
							23		Rational ClearCase
							20		Rational ClearQuest
							6		Rational Process Workbench
							28		Rational RequisitePro
							10		Rational Robot
							16		Rational TestFactory
							15		Rational TestManager
							56		Rational Rose Tool Set

							65	33	(Role)
							8		Integrator
							4		Change Control Manager
							4		Configuration Manager
							2		Any Role
S	S	S	S	S	S	S	176	112	(RUP Tools/Mentor/Scenario)
							5		Accessing Rational ClearCase from Rational Rose
							4		Checking Out and Checking In Configuration Items Using Rational ClearCase
							4		Creating a Development Workspace Using Rational ClearCase
							2		Creating an Integration and Building Workspace Using Rational ClearCase
							2		Creating Baselines Using Rational ClearCase
							2		Creating Multiple Sites Using Rational ClearCase
							2		Delivering Your Work Using Rational ClearCase
							4		Setting Policies Using Rational ClearCase
							2		Setting Up the Implementation Model Using Rational ClearCase
							5		Setting Up the Implementation Model with UCM Using Rational ClearCase
							6		Updating Your Project Work Area Using Rational ClearCase
							2		Using UCM Change Sets with Rational ClearCase
							2		Defining Change and Review Notifications Using Rational ClearQuest
							2		Establishing a Change Request Process Using Rational ClearQuest
							6		Reporting Defect Trends and Status Using Rational ClearQuest
							5		Reporting Review and Work Status Using Rational ClearQuest
							8		Submitting Change Requests Using Rational ClearQuest
							3		Viewing the History of a Defect Using Rational ClearQuest
							3		Setting Up and Managing the Rational Process Workbench Workspace
							2		Archiving Requirements Using Rational RequisitePro
							2		Baselining a Rational RequisitePro Project
							2		Viewing Requirement History Using Rational RequisitePro
							4		Evaluating Test Coverage Using Rational TestFactory
							3		Evaluating the Results of Executing a Test Suite Using Rational TestFactory
							3		Executing a Test Suite Using Rational TestFactory
							4		Using Rational TestFactory to Measure and Evaluate Code-based Test Coverage on Rational Robot Test Scripts
							3		Viewing Logs and Evaluating the Execute Test Suite Using the Rational TestManager
S	S	S	S	S	S	S	176	151	(Activities)
							3		Confirm Duplicate or Rejected CR
							3		Create Baselines
							1		Create Deployment Unit
							3		Create Development Workspace
							3		Create Integration Workspaces
							4		Deliver Changes
							5		Establish Change Control Process
							13		Establish Configuration Management (CM) Policies
							4		Make Changes
							1		Perform Configuration Audit
							4		Promote Baselines
							4		Report on Configuration Status
							3		Review Change Request
							6		Set Up Configuration Management (CM) Environment
							2		Submit Change Request

										2		Update Change Request
										3		Update Workspace
										3		Write Configuration Management (CM) Plan
R	R	R	R	R	R	R	R	R	R	233	90	{Artifact}
										6		Bill of Materials
										45		Change Request
										3		Configuration Audit Findings
										24		Configuration Management Plan
										7		Deployment Unit
										5		Project Measurements
										11		Project Repository
										13		Work Order
										14		Workspace
										237	9	{Artifact}
V	V	V	V	V	V	V	V	V	V	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
V	V	V	V	V	V	V	V	V	V	73		Map © by Yaozhong Chen 2002
V	V	V	V	V	V	V	V	V	V	73		Map Editing WMJ
V	V	V	V	V	V	V	V	V	V	113		Map © by Yaozhong Chen 2002/02/18

8. Discipline— Project Management view

										237	1	{Context Tuple Unique Id}
57	58	59	60	61	62	63	64	65		237		id
										237	13	{Context View}
V	V	V	V	V	V	V	V	V	V	9		Project Management
										70	4	{Phase}
V	V	V	V	V	V	V	V	V	V	66		Inception
V	V	V	V	V	V	V	V	V	V	67		Elaboration
V	V	V	V	V	V	V	V	V	V	67		Construction
V	V	V	V	V	V	V	V	V	V	68		Transition
										70	9	{Discipline Set/Sub Phase}
V	V	V	V	V	V	V	V	V	V	14		Project Management
										9	10	{Project Management}
										1		Start
										2		Conceive New Project
										3		Evaluate Project Scope and Risk
										4		Develop Software Development Plan
										5		Plan for Next Iteration
										3		Manage Iteration
										2		Monitor and Control Project
										2		Close-out Phase
										2		Close-out Project
										5		End
R	R	R	R	R	R	R	R	R	R	176	15	{Project Management}
										15		Rational TestManager
										65	33	{Role}
	V	V	V	V	V	V	V	V	V	8		Project Manager
	V	V	V	V	V	V	V	V	V	7		Project Reviewer
S	S	S	S	S	S	S	S	S	S	176	112	{Project Management}
										5		Performing Test Activities Using Rational TestManager

S	S	S	S	S	S	S	S	S	S	176	151	Activities
					V					2		Acquire Staff
					V					1		Assess Iteration
			V							1		Compile Software Development Plan
			V							1		Define Monitoring & Control Processes
			V							1		Define Project Organization and Staffing
	V	V		V						3		Develop Business Case
			V							1		Develop Iteration Plan
			V							1		Develop Measurement Plan
			V							1		Develop Problem Resolution Plan
			V							1		Develop Product Acceptance Plan
			V							1		Develop Quality Assurance Plan
			V							1		Develop Risk Management Plan
			V			V				2		Handle Exceptions and Problems
	V	V								2		Identify and Assess Risks
					V					1		Initiate Iteration
	V									1		Initiate Project
					V					1		Iteration Acceptance Review
					V					1		Iteration Evaluation Criteria Review
				V						1		Iteration Plan Review
							V			1		Lifecycle Milestone Review
							V			1		Monitor Project Status
								V		1		Prepare for Phase Close-Out
								V		1		Prepare for Project Close-Out
								V		1		Project Acceptance Review
	V									1		Project Approval Review
			V							1		Project Planning Review
							V			1		Project Review Authority (PRA) Project Review
							V			1		Report Status
							V			1		Schedule and Assign Work
R	R	R	R	R	R	R	R	R	R	233	90	Artifacts
		U	U	U						10		Business Case
					U			U		45		Change Request
						U	U	U	U	15		Issues List
							U	U	U	5		Iteration Assessment
				U						22		Iteration Plan
										1		Measurement Plan
										2		Problem Resolution Plan
										3		Product Acceptance Plan
									U	5		Project Measurements
										2		Quality Assurance Plan
					U					20		Requirements Attributes
			U	U	U	U	U	U	U	41		Review Record
		U	U	U	U				U	6		Risk List
										2		Risk Management Plan
			U	U	U	U	U			16		Software Development Plan
						U	U			5		Status Assessment
										19		Test Evaluation Summary
										26		Test Plan
										31		Vision
										13		Work Order
										237	9	Patterns
V	V	V	V	V	V	V	V	V	V	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
V	V	V	V	V	V	V	V	V	V	73		Map © by Yaozhong Chen 2002

V	V	V	V	V	V	V	V	V	73	Map Editing WMJ
V	V	V	V	V	V	V	V	V	113	Map © by Yaozhong Chen 2002/02/18

9. Discipline— Environment view

					237	1	(Context Tuple Unique Id)
66	67	68	69	70	237		id
					237	13	(Context View)
V	V	V	V	V	5		Environment
					70	4	(Phase)
V	V	V	V	V	66		Inception
V	V	V	V	V	67		Elaboration
V	V	V	V	V	67		Construction
V	V	V	V	V	68		Transition
					70	9	(Discipline Set/Sup Phase)
V	V	V	V	V	10		Environment
					5	6	(Environment)
					1		Start
					2		Prepare Environment for Project
					3		Prepare Environment for an Iteration
					3		Prepare Guidelines for an Iteration
					3		Support Environment During an Iteration
					1		End
R	R	R	R	R	176	15	(RUP Tools)
		V			23		Rational ClearCase
	V	V			6		Rational Process Workbench
		V			28		Rational RequisitePro
		V			10		Rational Robot
		V			56		Rational Rose Tool Set
					65	33	(Role)
			V		8		Business-Process Analyst
			V		6		System Analyst
			V		3		User-Interface Designer
	V	V			2		Process Engineer
				V	1		System Administrator
S	S	S	S	S	176	112	(RUP Tools/Mentors/Scenario)
		V			2		Setting Up Version Control using Rational Rose RealTime with Rational ClearCase
		V			2		Setting Up Rational Rose for a Project
	V	V			3		Setting Up and Configuring the Rational Process Workbench Tool
		V			2		Setting Up Rational RequisitePro for a Project
		V			2		Setting Up the Test Environment in Rational Robot
S	S	S	S	S	176	151	(Activities)
	V				1		Assess Current Organization
			V		3		Develop Business-Modeling Guidelines
			V		3		Develop Design Guidelines
	V	V			2		Develop Development Case
			V		1		Develop Manual Styleguide
			V		2		Develop Programming Guidelines
	V	V			3		Develop Project-Specific Templates
			V		3		Develop Test Guidelines
			V		1		Develop Tool Guidelines
			V		1		Develop Use-Case Modeling Guidelines

			V		1		Develop User-Interface Guidelines
			V		1		Launch Development Case
		V			1		Select and Acquire Tools
			V		6		Set Up Tools
				V	1		Support Development
			V		1		Verify Tool Configuration and Installation
R	R	R	R	R	233	90	(Artifact)
					2		Business Modeling Guidelines
					2		Design Guidelines
					9		Development Case
					1		Development Infrastructure
					8		Development-Organization Assessment
					3		Manual Styleguide
					2		Programming Guidelines
					7		Project-Specific Templates
					18		Test Guidelines
					1		Tool Guidelines
					8		Tools
					15		Use-Case Modeling Guidelines
					1		User-Interface Guidelines
					237	9	(Artifact)
V	V	V	V	V	73		Syntax and Patterns © by W.M. Jaworski, 1988-2002
V	V	V	V	V	73		Map © by Yaozhong Chen 2002
V	V	V	V	V	73		Map Editing WMJ
V	V	V	V	V	113		Map © by Yaozhong Chen 2002/02/18

-----End of Appendix-----