Process Model Combining the Artifact Process With Communication Path

Yi Zhou* and Koichiro Ochimizu*

September 26, 2003
IS-RR-2003-011

*School of Information Science
Japan Advanced Institute of Science and Technology (JAIST)
Asahidai 1-1, Tatsunokuchi
Ishikawa, 923-1292 Japan
yi-zhou@jaist.ac.jp, ochimizu@jaist.ac.jp

Process Model Combining the Artifact Centered Process with Communication Path

Yi Zhou† and Koichiro Ochimizu†

† Information Science, Japan Advanced Institute of Science and Technology
Asahidai 1-1, Tatsunokuchi-town, Nomi-gun, Ishikawa, 923-1292 Japan
E-mail: † {zhou-yi, ochimizu} @jaist.ac.jp

Abstract In this paper, we propose a process model combining the RUP(Rational Unified Process) and orgnizational pattern by J.Coplien. In the software development,not only obeying the process but also communicating with other developers are important.Although the population of RUP,there aren't any discussions on communication.By talking about the communication path,the orgnizaitonal pattern by J.Coplien became a family of patterns that can be used to shape a new orgnization.In order to support the communication in software development,we will combine the RUP and orgnizaitonal pattern by J.Coplien.

Keyword RUP(Rational Unified Process), orgnizational pattern by J.Coplien, process model

1. INTRODUCTION

Rational Unified Process(RUP)[1] is a kind of artifact-centered software process, which defines a series of works on producing artifacts,such as diagram and source,and the sequences of these works.But, In order to make the development successful, The communication is also important.So, for the development, it's necessary to not only show the works on producing artifacts but give the communication support as well.Based on analyzing successful development teams, Organization pattern by J.Coplien is a group of patterns talking about the way of development and organization structure with the communication path[2].

This paper discusses a process model combining the RUP and Organization pattern by J.Coplien and proposes a solution to melt the artifact-centered activities and communication.For this solution,a role management model will be defined,which maintains the sequence of producing the artifacts and arrange the RUP by the roles again.

By the process model in this paper,it's possible to (1)distinguish the necessary communication and unnecessary communication and (2)find the communications dependent from the works on producing artifacts.

The paper is organized as follows:The basic concepts of organization patterns by J.Coplien is given in the next section.the role management model is defined in Section 3.The organization patterns by J.Coplien and role management model are combined in Section 4.The paper ends with conclusions.

2. BASIC CONCEPTS OF ORGANIZATION PATTERNS BY J.COPLIEN

J.Coplien defines the role and communication below.

- Pattern 5 : Form Follows Function : Group closely related activities.Name the abstractions resulting from the grouped activities,making them into roles.
- Pattern 26 : Shaping Circulation Realms : Proper communication structures between roles are key to organizational success.Communication follows semantic coupling between responsibilities.

3. ROLE MANAGEMENT MODEL OF RUP

3.1. ARRANGE THE RUP BY ROLE

By investigating the roles of the RUP in [1],the roles are abstracted like diagram 2.

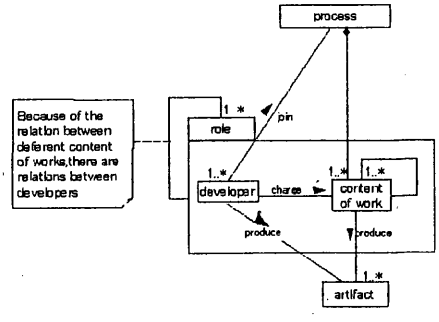

Fig 2 the meta model of roles of the RUP

RUP groups closely related activities, in order to produce artifacts, and defines this group as the contents of work. For example, in order to design a use-case, classes are identified, the interaction of objects are described and special requirements are identified. These activities are defined as contents of work named " use case design ".

RUP defines the member who participates in development, a sub development team, or the whole development team as a developer. Since the contents of work are performed by the developer, there is responsibility relation between the developer and the contents of work. The contents of work and the developer who is responsible for these contents of work are packaged and defined as a role. For example, the developer who plays the role as a use case engineer is responsible for the contents of work of "use case design".

There are sequences between each contents of work. For example, "use case analyze" consults the definition sentences created by the contents of work named "use case. definition ".

3.2. THE ROLE MANAGEMENT MODEL OF RUP

The role management model of RUP is shown in Fig. 3. All roles of RUP are classified into six kinds, a system engineer, an architect, a system integrator, a user interface designer, a use case engineer, and a component engineer.[1] The contents of work of each role are expressed using the role's artifact and the responsibility for the artifact.

3.2.1.SYSTEM ENGINEER

The system engineer represents the system analyst of requirement phase, and the test designer of test stage. The system analyt of requirement phase defines the boundary of the system, finds out actor and use cases, and be responsible for guaranteeing the consistency of use case model. The test designer of test stage is also responsible for deciding the schedule of test, describing test cases and test procedure,evaluating the performed.

ROLE	SYSTEM	ELEMENT	RELATION
S analyst	UM	· U · A	· UandA · Us
Tdesigner	TM	· TC · TP	TC and TP

S:system U:usecase A:actor T:test UM:usecase model TM:test model TC:test case TP:test procedure

table1 system engineer's artifacts

3.2.2.ARCHITECT

An architect represents the architect of requirement phase, analysis phase, design phase, implementation stage, and test stage. The architect of requirement stage describes important use cases and gives priority to each use case. The architect of analysis phase is responsible for adjusting the analysis model, and identifying analysis packages. The architect of design phase is responsible for adjusting the design model and identifying the subsystems and notes. The architect of implementation phase is responsible for adjusting implementation model and

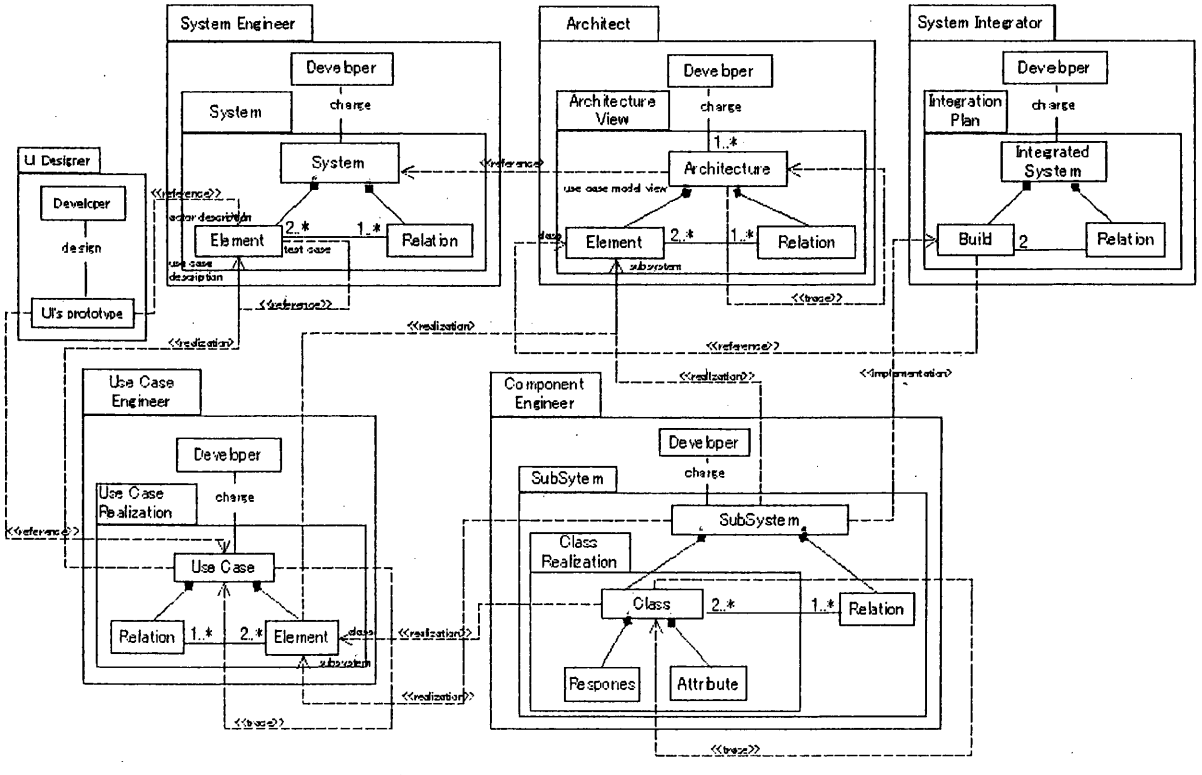

Fig.3 RUP's Role Management Model

identifying the important components and maping the component to the notes which can be performed.

architect

PHASE	ARCHITECTURE VIEW	ELEMENT	RELATION
requirement	UM'V	importantU	Us
Analysis	Analysis M'V	· analysisP · U	· analysis P and U · Us
Design	· designM ∅ V · deploymentM'V	· SB · I	· SBs · SB and I · SB and N · Ns
Implement	· implementM'V · deploymentM'V	· C · I · N	C and N

U:usecase M:model V:view P:package SB:sub system I:interface C:component N:note

table2 architect's artifact

3.2.3.SYSTEM INTEGRATOR

The system integrator represents the system integrator of implement stage. In the responsibility of the system integrator of implement stage, integration of each build when mounting the plan is included.

ROLE	ELEMENT	RELATION
S integrator	B	builds

S:sysem B:build

Table3 System integrator's artifact

3.2.4.USER INTERFACE DESIGNER

The user interface designer represents the user interface designer of requirement stage. The user interface designer of requirement stage designs the appearance of a user interface, and develops the prototype of a user interface.

ROLE	UI's prototype
UI designer	UI 's prototype

UI: user interface

Table4 user interface designer's artifact

3.2.5.USE CASE ENGINEER

A use case engineer represents the use case definitor of requirement stage, and the use case engineer of analysis phase and a design stage, and the tester of test phase. The use case definior of requirement phase has the responsibility of describing the details of the use cases, and does work closely with the actual user. The use case engineer of analysis phase is responsible for adjusting several " use case realization-analysis " and identifying the interaction of analysis classes and analysis objects. The use case engineer of a design phase is responsible for adjusting several " use case realization-design" and identifying the interaction of design classes and design objects. The testor of a test phase performs an integrated test required for each created build and performs a system test.

PHASE	U REALIZATION	ELEMENT	RELATION
Requirement	U definition	_	_
Analysis	U realization	Analysis C	Analysis Cs
Design	U realization	· design C · I · SB	· design Cs · I and SB
test	Test case	bug	Bugs

U:usecase C:class I:interface SB:subsystem TC:test case

Table5 use case engineer's artifact

3.2.6.COMPONENT ENGINEER

A component engineer represents the component engineer of analysis phase, design phase, and implement phase. In analysis phase, a component engineer defines and maintains one or more analysis classes, classes' attribute, classes' relation, and special demands. In design phase, a component engineer defines and maintains operation of one or more design classes, classes' method, classes' attribute, classes' relation, and a deployment demand. A component engineer of the implement stage writes the source code of one or more components.

PHASE	SB	CLASS	ATT	RES	RELATION
Analysis	Analysis P	Analysis C	ATT	RES	Analysis C
Design	SB	Design C	ATT	RES	Design Cs
Implement	SB	COM	ATT	OPE	COMs

C:class SB:subsystem COM:component P:packege

Table 6 complonent engineer's artifact

3.3. THE DEFINITION OF THE RELATION BETWEEN EACH ROLES

The relation between roles is expressed using the relation between the artifacts which each role creates.

3.3.1. REALIZATION RELATION

When performing detailed design or implementing artifacts of different roles, the relation between roles is realization relation.

- The use case engineer of requirement phase starts with the use case diagram relevant to the use case model which the systems engineer created, and describes a use case in detail.
- The component engineer of analysis phase defines analysis classes, analysis classes' attribute, and

analysis classes' special requirements based on " use case realization-analysis" which the use case engineer created.

- The component engineer of analysis phase checks that the analysis packages which the architect identified are independent as much as possible to each other.
- The component engineer of design phase identifys operation of design classes, classes' an attribute, and special requirements based on " use case realization-design" which the youth case engineer created.
- The component engineer of design phase checks that the subsystem which the architect identified are independent as much as possible to each other.
- The use case engineer of design phase describes the object of the class contained in subsystem level, when there is an relation betweem subsystems of " use case realization-design."
- A tester performs test procedure of the test case which the test designer created.

3.3.2.REFERENCE RELATION

In order to realize responsibility by refering the artifacts of different role, when creating new artifacts, a reference relation between roles occurs.

- The user interface designer of require phase designs a user interface by refering use case detailed description.
- The architect of require phase takes in architecture view of a use case model by refering use case model.
- A test designer builds a test case and test procedure by refering the description sentence of use case diagram.
- System integrated person plans build.

3.3.3.TRACE RELATION

Trace relation is between the same roles of a different phases.

- Based on a use case definition, the use case engineer of analysis phase identify analysis classes.
- Based on "use case realization-analysis", the use case engineer of design phase identify design classes.

Organization Patterns		Roles of RUP
Name	Explanation	Name
Engage Customers	Closely couple the Customer role to the Developer and Architect, not just to QA or marketing	Customer Architect Test Designer System Analyst Use Case Definitor
Developer Controls Process	Make the Developer the process information clearinghouse. Place the Developer role at a hub of the process .Responsibilities of Developers include understanding requirements, reviewing the solution structure and algorithm with peers, building the implementation, and unit testing	UI Designer Use Case Engineer Component Engineer System Integrator System Analyst Use Case Definitor
Patron	Give the project access to a visible, high-level manager, who champions the cause of the project. The patron can be the final arbiter for project decisions, which provides a driving force for the organization to make decisions quickly. The patron is accountable to remove project-level barriers that hinder progress	Projecto Manager Architect Customer Developer
Architect Controls Product	Create an Architect role as an embodiment of the architectural principles that define an architectural style for the project, and of the broad domain expertise that legimitimizes such a style. The Architect role should advise and influence Developer roles, and should communicate closely	Architect Customer Developer
Firewalls	Create a ManagerRole, who shields other development personnel from interaction with external roles	Projecto Manager Customer Developer
GateKeeper	One project member, a PublicCharacter with an engaging personality, rises to the role of GateKeeper_. This person disseminates leading-edge and fringe information from outside the project to project members, "translating" it into terms relevant to the project.	System Analyst Use Case Definitor Customer Developer
Engage QA	Make QA a central role. Couple it tightly to development as soon as development has something to test.	Test Designer Tester System Integrator
Application Design Is Bounded by Test Design	Scenario-driven test design starts when the customer first agrees to scenario requirements. Test design evolves along with software design, but only in response to customer scenario changes: the source software is inaccessible to the tester.	Test Designer Customer

Table.7 Combination of Roles of RUP and Organization Pattern

- The component engineer of design phase identifys operation of a design class, and design class's attributes.
- The component engineer of implement phase define classes based on operation of design classes.

4. combination of the role management model of RUP and the organization pattern by Coplien

Combination with the organization pattern of Coplien is realized by using the role management model of RUP which Chapter 3 defined.

4.1. THE ORGANIZATION PATTERN BY COPLIEN ON ROLES

There are several patterns about roles and several patterns without regarding roles in organization pattern by Coplien. In order to combine with RUP, the organization pattern of Coplien about roles is observed.

- Pattern11 : Developer Control Process
- Pattern 12 : Patron
- Pattern 13 : Architect Control Product
- Pattern 18 : Application Design Is Bounded by Test Design
- Pattern 19 : Engage QA
- Pattern 20 : Engage Customers
- Pattern 24 : FireWalls
- Pattern 25 : GateKeeper

4.2. COMPARISON OF ROLE MANAGEMENT MOLDE AND THE ORGANIZATION PATTERN BY COPLIEN

Each role of an organization pattern by Coplien and each role of RUP are compared. If there are similar activities, they will be combined. The following combination is possible.

- Architect and pattern12,13,20:RUP's architect is responsible for the architecture of each development phase, we consider that there are common activities with the architect of the organization pattern of Coplien. As a result of combination, the same effect as a patron's role in a project is expected to the architect of RUP.
- Tester and patterns18,19,20:RUP's tester design test cases based on use cases, we consider that there are QA of the organization pattern by Coplien has common activities.
- Tester and pattern 19:RUP's tester performs a test case and test build, we consider that there are QA of the organization pattern by Coplien has common activities.
- System analyst and patterns 11, 12, 13, 20 24,25:RUP's analyst identify the use case of the customer, we consider that there are common activities with the developer of the organization pattern by Coplien. As a result of combination, the system analysis person of RUP is a developer. Other developers communicate with a customer through a system analyst.
- Use case definitor and patterns 11, 12, 13, 20, 24, 25:RUP's use case definitor detail the use case of the customer who is present in the exterior of a project, we consider that there are common

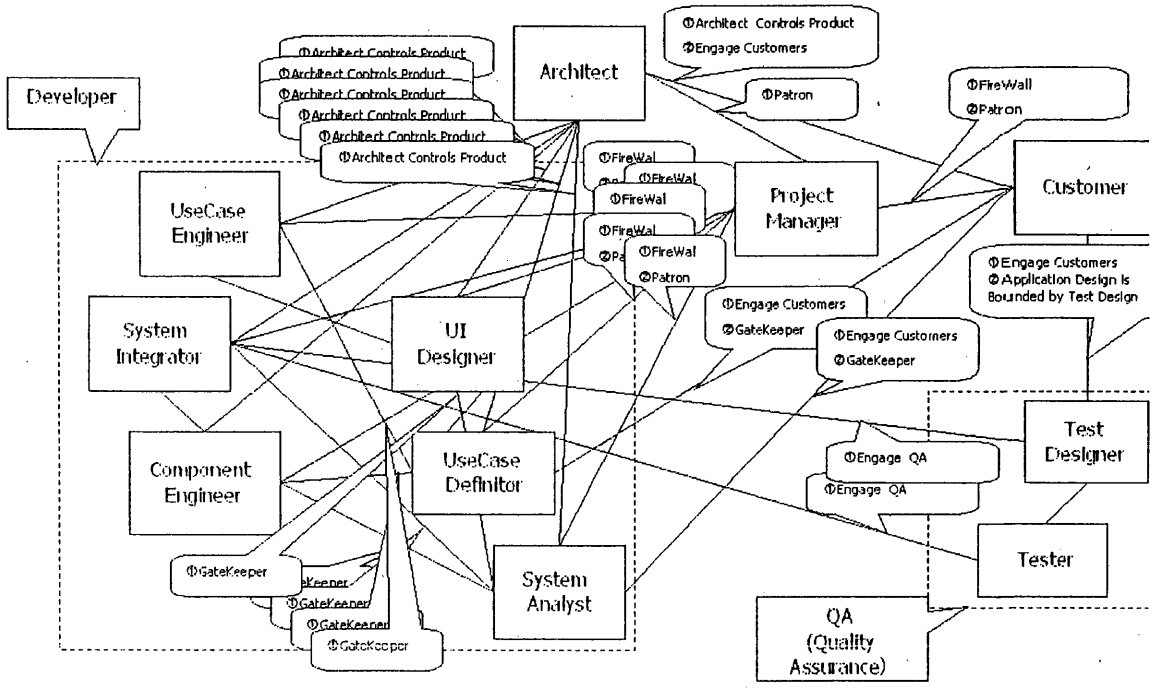

Fig.4 communication path

activities with the developer of the organization pattern by Coplien and use case definitor. As a result of combination, use case definitor of RUP are developers.Other developers communicate with a customer through a use case definitor.

- Use case engineer (analysis, design) and patterns 11, 12, 13,24,25:RUP's use case engineer analyze use case, we consider that there are common activities with the developer of the organization pattern by Coplien. As a result of combiation, in a development organization, the use case engineer (analysis, design) of RUP becomes a developer.
- Component engineer (analysis, a design, mounting), and patterns 11, 12, 13, 24 ,25:RUP's component engineer perform implement and a simple substance test, we consider that there are common activities with the developer of the organization pattern by Coplien. As a result of combination, in a development organization, the component engineer (analysis, a design, mounting) of RUP is a developer.
- System integrator and patterns 11, 12, 13, 19, 24,25:RUP's system integrator integrate builds, we consider that there are common activities with the developer of the organization pattern byCoplien. As a result of combining, in a development organization, the system integrator of RUP is a developer.

The result above is summarized in Table 7. The communication path between each role in a role management model is shown in Fig. 4.

4.3. Check of Communication Support in creation of artifacts

- The communication which should be taken, and the communication which should not be taken : example (Fig. 5): Developers are protected by the project manager because of pattern 24 firewall .

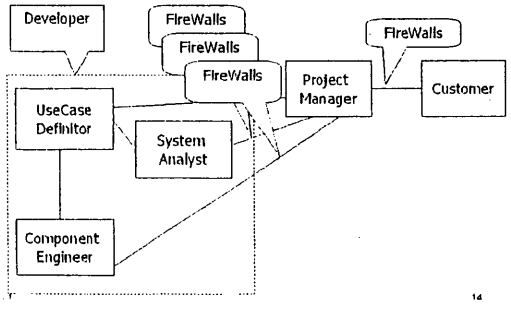

Fig5 : firewall : The communication which should be taken, and the communication which should not be taken

- the communication independent to the work flow: example (Fig. 6): when gap of the recognition about a design arises, because of combination with a patron (Patron), the last conclusion is drawn with a project manager by communication

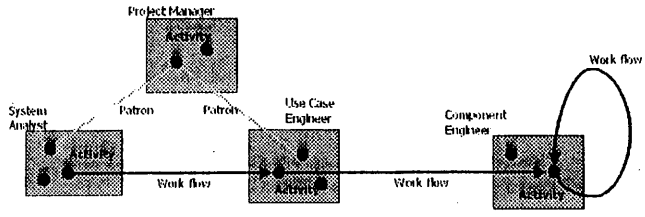

図 6 : Patron : the communication independent to the work flow

5. CONCLUTION

In this paper, The contents of communication are not touched although the role and the communication path are defined as the process model in detail. Considering that communication occurs centering on artifact, it is required to define the artifact object expressing the detailed information on artifact, and it is the subject in future[3] a. Moreover, it is necessary to perform finer consideration about the definition of the role which Coplien defined, and the role of RUP, and to reexamine the combination.

REFERENCES

[1] Ivar Jacobso,Grady Booch,James Rumbaugh : "UNIFIED PROCESS WITH UML"、2000.

[2] PloPD Editor : "The pattern language for a program design"、software bank publishing、2001.

[3] Koichiro OCHIMIZU : "Consideration on a software process model for a distributed co-operative software development"、THE INSTITUTE OF ELECTRONICS、SE-131、2001.