Process Model Combining the Artifact Centered Process with Communication Path

Saw Sanda Aye, Yi Zhou, Koichiro Ochimizu

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

Extended Abstract

In this paper, we propose a process model combining the RUP (Rational Unified Process) and organizational pattern by J. Coplien. In the software development, not only obeying the process but also communicating with other developers is important. Although the population of RUP increases, there aren't any discussions on communication. We address this issue and point out the effects of communications in development process. In order to support the communication in software development, we combined the RUP and organizational pattern by J .Coplien based on the roles of developers.

Developing a commercial software product is a large undertaking that may continue over several months to possibly a year or more. To build a successful system we must know what its prospective users want and need. Moreover, to make the development successful, the communication between developers and customers is also important. During the developing the software product, over the time, it is necessary to communicate between developers and also developers and customers. Sometimes, unnecessary communication makes the project burden. So, for the development, it's necessary to not only show the works on producing artifacts but give the communication support as well since proper communication is fundamental to organizational success.

In the global software development, the communication is the half of development activities (sometime more than half) and communication factors into the productivity of individuals and organizations. Curtis, Krasner and Iscoe discussed in detail how organizational communications is crucial to managing the project [5]. There are many attempts to support communication using groupware, database such as E-Mail, Blackboard, Design Intent, CVS. Design Intent Model [4] provides a communication infrastructure but it doesn't touch to development process. During the process, to maintain a certain overhead of communication channel, define roles and responsibilities, and it makes more clear to all stakeholders to find the correct person to contact [6, 7, 8]. Therefore, based on the role, here, we directly combined software process and communication process.

Rational Unified Process (RUP)[1] is a kind of artifact-centered software process, which defines a series of works on producing artifacts, such as diagrams and source, and the sequences of these works. 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 proposes 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.

In the organization patterns by J. Coplien, there are many patterns concerning with roles and without concerning with roles. This paper is based on the patterns which are talking about roles and communication path.

J. Coplien defines the role and communication as below.

- Pattern 5 Form Follows Function

  Group closely related activities. Name the abstractions resulting from the grouped activities, making them into roles. The associated activities become the responsibilities (job description) of the roles.

- Pattern 26 Shaping Circulation Realms

  Proper communication structures between roles are key to organizational success. Communication follows semantic coupling between responsibilities. Job responsibilities suggest the appropriate interactions between roles.

By investigating the roles of the RUP in [1], the roles are abstracted as shown in figure 1. 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 interactions 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.

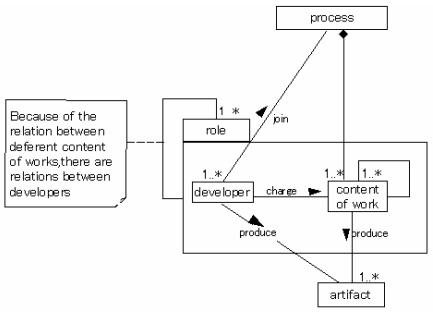

Fig 1: The meta-model of roles of the RUP

The role management model of RUP is shown in Fig. 2. RUP classified into six roles such as a system engineer, an architect, a system integrator, a user interface designer, a use case engineer, and a component engineer [1].

The relation between roles is expressed using the relation between the artifacts which each role creates. When performing detailed design or implementing artifacts of different roles, the relation between roles is realization relation. In order to realize responsibility by referring the artifacts of different roles, when creating new artifacts, a reference relation between roles occurs. Trace relation is between the same roles of different phases.

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　Fire Walls
- Pattern 25　Gatekeeper

Each role of an organization pattern by Coplien and each role of RUP are compared and if there are similar activities, they will be combined. For example, Architect and pattern 12, 13, 20: RUP's architect is responsible for the architecture of each development phases. 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.

According to the pattern 12 of J. Coplien, we added the role of project manager in RUP since any project must have a high level manager role to control the whole project's activities and project manager is also boundary spanner [5]. The communication path between each role in a role management model is shown in Fig. 3.

We found that the following plus points by combining RUP and organization pattern by J. Coplien.

- The communication which should be taken and the communication which should not be taken:

  Sometimes, unnecessary communication causes to the project implementers to confuse with conflicting emotions. By combining with organizational patterns by J. Coplien, developers are

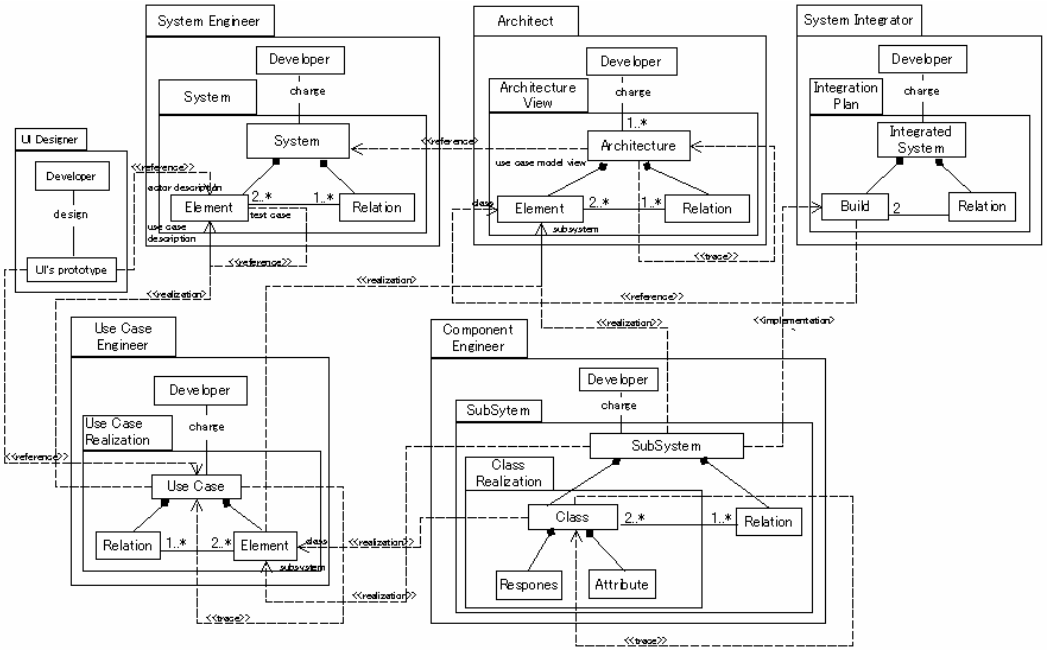

Fig.2　RUP's Role Management Model

protected by the project manager because of pattern 24 fire walls where the project manager shields the development personnel from interaction with external actors.

- The communication independent to the work flow:

  Even thought we combine RUP with organization patterns by role model, it does not effect to the core workflows of RUP. But it can support more facilities to RUP's process, for example, if there is any gap of the recognition during any iteration of core workflows, because of combination with a patron (Patron), the last conclusion is drawn with a project manager by communication.

In this paper, to be successful development, we pointed out not only producing artifacts is important but how communication is also. RUP defined how a development process should proceed the work and J. Coplien defined how organization should be controlled. By combining these two, we can solve how to control the organization to proceed the work successfully.

In this paper, the contents of communication are not touched because we are interested in constructing the process model combining the artifact centered activities and proper communication among roles. This paper can support for static communication of development process. We also already developed model and tool that can thread for each topic to represent streams of message in email communications [3] to support dynamic communication. And we are now going to combine them to promote the efficiency of collaboration work. Moreover, it is necessary to perform finer consideration about the definition of the role which J. Coplien defined, and the role of RUP, and to reexamine the combination.

References

[1] Ivar Jacobson,Grady Booch, James Rumbaugh "THE UNIFIED SOFTWARE DEVELOPMENT PROCESS" Addison Wesley Longman, Inc. 1999.

[2] Edited by Coplian and Schmidt "PATTERN LAGUAGE OF PROGRAM DESIGN" Addison Wesley Longman, Inc.1995.

[3] Hiroyuki Murakoshi, Akira Shimazu, Koichiro OCHIMIZU "Construction of Deliberation Structure in E-Mail Communication", Computational Intelligence, Volume 16, Number 4,2000.

[4] Michael E. Atwood : "Facilitating Communication in Software Development" , ACM 0-89791-673-5 /95/08, 1995

[5] Curtis, B., H.Krasner, and N. Iscoe. "A Field Study of the Software Design Process for Large Systems". Communications of ACM, 1988.31(11): pp.1268-1287

[6] Daniela Damian and others "Awareness meets requirements management: awareness needs in global software development", International Workshop on Global Software Development, Porland, Oregon, 2003

[7] Lori Kiel, "Experiences in Distribted Development: A Case Study", International Workshop on Global Software Development, Porland, Oregon, 2003

[8] Maria Paasivaara "Communication Needs, Paracties and Supporting Structures in Global Inter-Organizational Software Development Projects", International Workshop

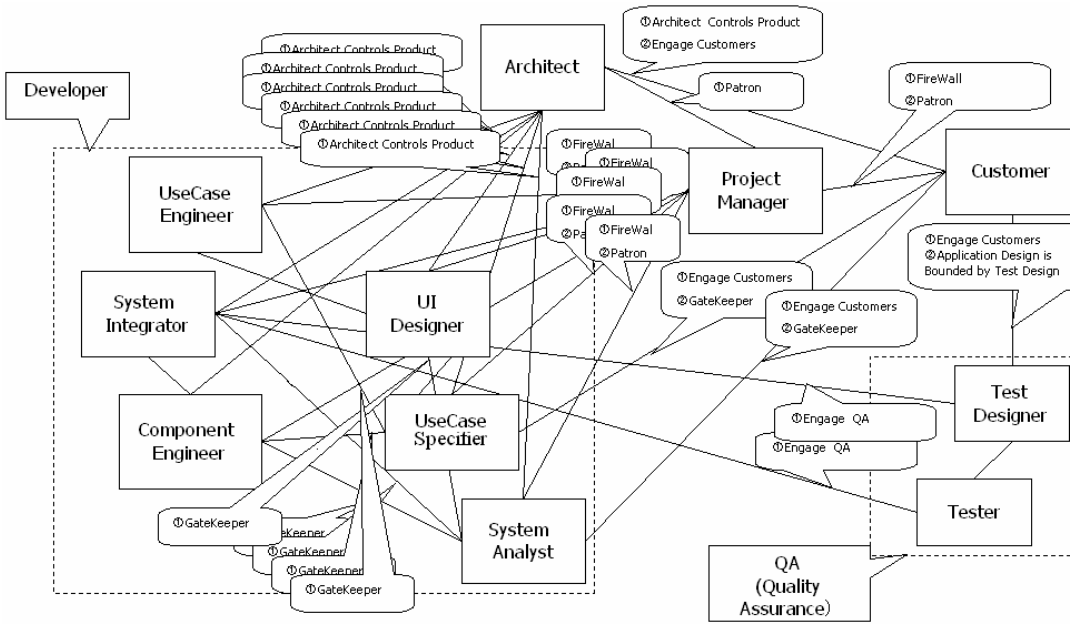

Fig. 3: The communication path between each role in a role management model

on Global Software Development, Porland, Oregon, 2003