Study of Support Environment for Information Sharing in Distributed Cooperative Software Development

Kazuyoshi Nishida (110097)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 13, 2004

Keywords: Distributed Cooperative Software Development, Support Environment for Information Sharing, E-mail communications, Deliberation Structure Model.

1 Background and Purpose

Today, we can use a lot of softwares what high performance and multifunction. On the other hand software development is increasing a its scale. As a result, Distributed Cooperative Software Development is getting commonly applied to softwares development. This method can help developers to handle many tasks, and developers can deal the tasks simultaneously and easily communicate by using networks. However, Distributed Cooperative Software Development causes several problems.

The problems are as follows:

- Conflicts and Duplications by increase source codes
- Increasing tasks of the project leader
- Not be able to determine about processes of the software development

In this paper, we define the information which needs to be treated in Distributed Cooperative Software Development and we research to support the method for these information sharing. Eventually, we design the support environment for the information sharing.

2 Problem Research and Analysis

In our distributed cooperative software development, due to the dynamical changes of its group structure and object function, there are inevitable liabilities such as source codes expansion and increasing project manager's tasks. In order to solve these problems, we define some terms about distributed cooperative software development. The treated information are as follows:

Information of deficient products: It is one of the most important aspects of software developments to write down the source codes and create its documents. However, there might be conflicts or duplications of tasks. For instance, it is likely that we edit the same place at the same time or create similar features in different places. They are caused by our insufficient recognition of what other developers are doing and how we are related to the others. The source codes expansion mostly brings these difficult problems to us. Therefore ,we need to share the information about deficient products such as the source codes and the documents with other implementers and be aware of our responsibility at current work.

Information of communications: In our distributed cooperative software development, we are often restricted to communicate directly, because our laboratories can be located different areas in the world. We are sometimes not efficient enough to develop any sort of softwares, because the restriction affects the procedure and result of the software development. Therefore, we need to handle network communication to support each other and recognize the update of the information of their current arguments.

Then, there is another argument that is how to share our information. If we started sharing our information, the project leader must face much more tasks. Consequently, we need to make the supportive idea to cut down the enormous tasks, and this is the purpose of our distributed cooperative software development.

3 Support Environment for Information Sharing

As we said above, it is required to support information sharing and reduce the project leader's tasks. Gforge is a sophisticated system which can establish the ideal environment for us to overcome these problems.

Support environment for information sharing can function within the existing systems such as a web browser. Developers are able to access by web browser and they can acquire the information which we are sharing. Another advantage of its use is the system linkage which can reduce the leader's tasks.

4 Problem of Support Communication for Information Sharing

GForge's environment enable us to share information which Distributed Cooperative Software Development needs. But GForge's environment causes problems about E-mail archive method from the mailing list. The developers have to read a lot of E-mail when they search old discussions. We approve that the problem of Deliberation Structure Model builds in GForge's environment.

5 Conclusion and Future Works

We researched about problems of Distributed Cooperative Software Development. And we designed support environment for information sharing by GForge. But GForge's environment had problem that E-mail archives method in Mailing List. This problem cause much worse effects on developers. Therefore Deliberation Structure Model is applied to Gforge's environment for remedy its problem. As a result we create support environment for information sharing which can handle software development smoothly.

The following shows future works.

- Investigation of new communication tools

  We deal with problems of Mailing List on support environment for information sharing. On the other hand developers can use another network communication tool which is different from E-mail. So we need to explore into other communication tools on the network.

- Continuous Administration and Evaluation Instrument

  The test administration was very short time and developers do not distributed because software development environment is the class of university. So we need other support more than long time software develop and large distribute environment. Moreover, we have to research for the method of the evaluation instrument and find out new issues.

- English Communication Support

  English is a very important language in the network communication. But now our Deliberation Structure Model makes only the deliberation structure tree of E-mail communication in Japanese. We need to research and design about the automatic system to make the deliberation structure tree of English communication for more effective information sharing environment.