Managing Inconsistency among Software Artifacts

Phan Thi Thanh Huyen, Ochimizu Lab, School of Information Science, JAIST

In collaborative software development, many change requests, such as adding new features or fixing bugs, can be implemented by different workers within a time interval. Each worker conducts his own change process (CP) that is a sequence of tasks applying changes to a set of artifacts. When a worker makes a change to an artifact, this change may affect the artifacts connected to this artifact by dependencies. However, because of communication problems and the complex and changeable nature of software, the workers do not always have enough information about the work of the others. As a result of that, a change to an artifact of a worker may unexpectedly affect the changes to other artifacts of other workers. Therefore, inconsistencies may occur from the affected artifacts.

We define inconsistency as a situation in which some artifacts are assigned values that are different from the intention of a worker, because he is unaware of the changes or the impact of the changes made by other workers, to the artifacts to which his changes apply. The goal of this dissertation is to build a Change Support Environment (CSE) that supports the workers in preventing, detecting, and resolving inconsistencies more effectively.

Differently from the existing studies in the area of inconsistency awareness that concentrated only on concurrent changes and considered them separately, this dissertation pays attention to both concurrent and non-concurrent changes, and the context of a change, i.e. the CP containing the change, rather than the ongoing changes only.

- We have defined the patterns of inconsistency with regard to the time orders of the tasks applying the changes to the artifacts, the relationships between the artifacts, and the CPs of the changes.
- We have proposed a context-based approach to solve the inconsistency problem more effectively. The CPs are managed to provide the contexts of the changes in the system. We have combined monitoring the workspaces of the workers (workspace awareness) with managing the progress of the CPs executed in the system (context awareness) to detect (potential) inconsistencies in real time. Information about the context of an inconsistency, the changes causing the inconsistency and their change processes, is provided to help the workers fully comprehend the inconsistency before resolving it.
- Based on the above approach, we have developed an inconsistency management support system that allows the workers to define, execute, and modify their CPs easily, and to receive inconsistency warnings along with the contexts of the inconsistencies to resolve the inconsistencies in advance.
- We have also given a formal method for modeling the main behaviors of CSE in Colored Petri Nets (CPN). CPN Tools is then used to verify the generated model to detect the patterns of inconsistency.

The novelty of this dissertation lies in addressing the inconsistency problem by considering the contexts of changes rather than just the changes themselves. By managing the execution of the CPs, monitoring the ongoing changes in the workspaces of the clients, and using the change history stored in the repositories of VCSs and the planned changes specified by the workers, we can detect in advance emerging inconsistencies that are not reported by the previous studies, and provide in detail the context of an inconsistency to help the workers understand the situation and have a timely decision for resolving the inconsistency before its effect goes further.

In summary, changes are indispensable in collaborative software development environments despite their high cost and risk. Our research, developing the model and the environment able to detect the (potential) inconsistencies that the workspace awareness approach cannot detect and supply the workers with the contexts of the inconsistencies, can help the workers implement changes more safely and efficiently in collaborative environments.