Modeling and Verification of Change Processes in Collaborative Software Engineering

Phan Thi Thanh Huyen, Kunihiko Hiraishi, and Koichiro Ochimizu

School of Information Science, Japan Advanced Institute of Science and Technology
Nomi-shi, Ishikawa, 923-1292 Japan
{huyenttp,hira,ochimizu}@jaist.ac.jp

Abstract. In collaborative software engineering, many change processes implementing change requests are executed concurrently by different workers. However, the fact that the workers do not have sufficient information about the others’ work and complicated dependencies among artifacts can lead to unexpected inconsistencies among the change-impacted artifacts. By focusing on the contexts of the changes, i.e. the change processes containing the changes, rather than the concurrent changes only like the previous works, we have proposed an approach that helps the workers detect and resolve the inconsistencies more effectively [1]. Our approach is to build a Change Support Environment (CSE) that represents the change processes explicitly as the Change Support Workflows (CSWs) and manages their execution based on our patterns of inconsistency, including many patterns besides the conflict patterns mentioned in the previous works. To evaluate the feasibility of our proposed approach, this paper presents a formal model of CSE using Colored Petri Nets (CPN) to model the artifacts, and both data flow and control flow of CSWs. CPN Tools is used to edit, simulate, and verify the CPN model of CSE to detect data-related abnormalities, in particular the patterns of inconsistency. Differently from the previous works in workflow modeling, our method for modeling CSWs using CPN can represent many aspects of a workflow, including data flow, control structure, and execution time, in one single model. Data and changes on the value of data are also represented explicitly. In addition, our modeling and verification method can be applied to other types of workflow.

Keywords: Inconsistency awareness, Change process, Collaborative software engineering, Colored Petri Nets, Modeling and verification.

1 Introduction

Collaboration is indispensable in software engineering along with the increased scale and complexity of projects. In a collaborative work, software artifacts are created based on the collaboration of many workers. However, because of the fact that the workers do not always have sufficient information about the others’ work, and complicated dependencies among the artifacts, changes to some artifacts of a worker may affect the changes of other workers. Therefore inconsistencies may happen among the change-impacted artifacts.
There were several attempts to detect conflicts, a type of inconsistency caused by concurrent changes to the same artifact (direct-conflict) or to dependency-related artifacts (indirect-conflict) [2]. Version control systems (VCSs) [3], widely used in collaborative software development, could detect only direct-conflicts at check-in time after the changes have been finished. To detect conflicts earlier, when the changes are being implemented, some recent studies [2, 4–7] have concentrated on workspace awareness, namely continuously sharing information of ongoing parallel activities across the workspaces.

The abovementioned studies [2–7] focused on conflicts and concurrent changes. However, because of the dependencies among artifacts, when a requirement changes, it is often necessary to implement a change process that is a sequence of tasks applying changes to a set of artifacts to implement a change request. Ignoring the context of the change, i.e. the change process containing the change, may neglect some inconsistencies that may only be detected much later in the development process (Sec. 2.1). Also, in these studies, when a (potential) conflict is detected, workers are aware of concurrent changes and the related artifacts only. Nevertheless, to resolve an inconsistency, a worker will need to consider the contexts of the changes causing the inconsistency rather than the changes only. Based on these considerations, in [1], we have presented an approach to dealing with the inconsistency problem more effectively. We have proposed a Change Support Environment (CSE) where the change processes are represented explicitly as the Change Support Workflows (CSWs) and their execution is managed based on the patterns of inconsistency. We have also identified and categorized these patterns, including the conflict patterns, considering both concurrent and non-concurrent changes and the contexts of the changes. If the change processes in the system can be managed, a worker could have a clearer view of the contexts of the changes of his and the related workers, and hence inconsistencies, including conflicts, can be detected earlier and resolved more easily.

To prove the feasibility of our proposed approach in [1], this paper presents a formal model of CSE and verifies this model to detect the patterns of inconsistency. As mentioned earlier, inconsistencies among CSWs are directly related to data, and hence, data flow verification is required in CSE. However, previous works in workflow verification mostly concentrated on structure verification, temporal verification, and resource verification [8]. Although data are an important aspect of workflows, only little research in data-flow verification can be observed and they focused on data-flow errors in a single workflow instance. Unlike previous works, we model the essential behaviors of CSE using Colored Petri Nets (CP-nets or CPN) [9]. Then, we use CPN Tools [10] to edit, simulate, and verify the CPN model of CSE to detect data-flow abnormalities including the patterns of inconsistency. We make the following contributions:

– Using Colored Petri Nets to model workflows instead of extending existing languages to represent the data factor like the works in [12–15].
– Being able to represent data and changes on the properties of data explicitly.
– Being able to represent both control flow and data flow in one single model.