

Title	ソフトウェア成果物間に発生する矛盾の管理法に関する研究
Author(s)	Phan, Thi Thanh Huyen
Citation	
Issue Date	2013-09
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/11552
Rights	
Description	Supervisor:落水 浩一郎, 情報科学研究科, 博士

氏名	PHAN THI THANH HUYEN		
学位の種類	博士(情報科学)		
学位記番号	博情第 282 号		
学位授与年月日	平成 25 年 9 月 24 日		
論文題目	Managing Inconsistency among Software Artifacts (ソフトウェア成果物間に発生する矛盾の管理法に関する研究)		
論文審査委員	主査	落水 浩一郎	北陸先端科学技術大学院大学 特任教授
		平石 邦彦	同 教授
		鈴木 正人	同 准教授
		青木 利晃	同 准教授
		佐伯 元司	東京工業大学 教授

論文の内容の要旨

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.

論文審査の結果の要旨

近年、ソフトウェア規模の増大につれ、ソフトウェア開発時および保守時に生じるソフトウェア変更作業のコストは膨大なものとなっており、その原因は以下の点にある。

- (1) 大規模ソフトウェアの開発は複数人の共同作業として実施され、それぞれの構成員が多くの中間成果物の作成・変更に関わっており、さらに、中間成果物間に膨大で複雑な依存関係が存在する。このため、ある変更スレッドにおける中間成果物内容の変更は、依存関係を通じて、自身の管理するまたは他人の管理する中間成果物に影響を与える。
- (2) さらに、複数個（一般には、数十個から数百個）の変更スレッドが複数個同時に存在し互いに干渉することにより、(1) の状況をさらに複雑にする。

- (3) 一般にソフトウェア変更作業はロングトランザクションとして実施され、データベースの並行制御や版管理システムの衝突検知を越える理論的枠組みを必要とし、また、作業の効率化と安全性を保証するためのツールが必要である。

本論文は、中間成果物の内、ソースコード群を対象にして、複数の変更スレッドが同時進行しているときに、変更により発生する矛盾を、変更プロセスに関する情報を利用して検出する手段を与えたものであり、また、変更作業を支援するツールを開発したものである。以下の成果を得ている。

- (1) 変更スレッドをペトリネットによる形式化した。
- (2) 複数の変更スレッドの走行による発生する矛盾に関して、3つの基準をもとに矛盾の構造的な特徴を明らかにし、矛盾パターンとして分類した。
- (3) それらの矛盾パターンを検知するアルゴリズムを与えた。
- (4) 上記の理論的成果をもとにした変更支援システムのプロトタイプを構築し、冒頭に述べた(1)から(3)の問題に対処しうる能力を有することを実験的に示した。

版管理システムのワークスペース管理をベースにした従来の方法では検出できない矛盾を、変更履歴および変更計画を利用して検出する手段を開発した点にオリジナリティと有用性がある。

以上、本論文は、大規模ソフトウェア変更制御、変更支援の方式について、理論的かつ実用的に新しい有用な展開を与えたものであり、学術的に貢献するところが大きい。よって博士（情報科学）の学位論文として十分価値あるものと認めた。