

Title	ソフトウェア共同開発における 変更作業支援ワークフローモデル
Author(s)	Phan, Huyen Thi Thanh
Citation	
Issue Date	2010-09
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/9145">http://hdl.handle.net/10119/9145</a>
Rights	
Description	Supervisor:Professor Koichiro Ochimizu, 情報科学研究科, 修士

# A Change Support Workflow Model for Cooperative Software Development

Phan, Huyen Thi Thanh (810202)

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

August 10, 2010

**Keywords:** Change Support Workflow, Unintentional Change in In-use Data, Time Data Workflow, Detection, Resolution.

## 1 Research Purpose

My research is aimed at developing a workflow model to support the software change process in a cooperative software development environment. This Change Support Workflow Model (CSW Model) is responsible for constructing and managing Change Supporting Workflows (CSWs), a sequence of activities defined to carry out a change request. Activities in CSWs take care of creating new software artifacts or modifying exiting ones. This means that data elements of CSWs are software artifacts which need to be read, modified or created in the change implementation process. In the scope of this thesis, we concentrate on synchronization of changes on shared artifacts by detecting and solving errors caused by uncontrolled access to shared data by CSWs in a CSW Model.

There are many CSWs executed at the same time in the cooperative environment. If some of these workflows modify data elements (software artifacts) being used by others, some errors such as lost data or inconsistency among data elements may occur and they could affect the correctness of the whole system. We have abstracted these problems as Unintentional Change in In-use Data (UCID). We define UCID as a situation in which

some data values are lost or some data elements are assigned values different from the intentions of workflow designers, due to non-deterministic access to shared data by different activities. First, we identify UCID patterns including intra-UCID, caused by activities in the same workflow, and inter-UCID, caused by activities in different workflow. Then, we develop algorithm to detect potential UCID patterns at build time and give some solutions to resolve UCID problem. Finally, UCID theory is applied to the CSW Model to detect and resolve errors concerning shared data among CSWs at build time.

## 2 Background

Developing CSW Model is the second part of a project on building a change support environment for cooperative software development. One unique feature of this project is to generate workflows which represent activities needed to implement change requests. Change workers can perform change activities safely and efficiently in the cooperative environment based on the generated workflows.

In the first part of the project, an information model was built. This model helps to generate automatically dependency relationships among UML model elements, among Java classes, and between a UML model element and a group of Java collaboration classes. A dependency relationship is defined as a relationship between two elements in which a change in the supplier element requires a change in the client element. The generated dependency relationships will be used in the CSW Model to generate CSWs.

## 3 Research Approach

We have abstracted problems caused by uncontrolled access to shared data by different CSWs as Unintentional Change in In-use Data (UCID). If UCID is discovered at runtime, a recovery mechanism must be performed to ensure the correctness of the whole system. However, recovery is a rather expensive work, especially in a cooperative environment with many concurrently executing workflows. Therefore, our approach is to detect

potential UCID patterns at build time to reduce risk to the target process. Early potential UCID detection will help workflow designers to have a more comprehensive view of the system, and make timely adjustments to the original workflows to avoid errors at runtime. In our approach, we assume that control flow (order of activities in workflow) and data flow can be given before workflow execution. CSW Model satisfies these assumptions because the dependency relationships generated by the information model helps us to identify the structure (control flow) and data elements of CSW at build time.

First, we identify UCID patterns including intra-UCIDs and inter-UCIDs. The former is caused by concurrent activities in the same workflow while the latter is caused by activities in different workflows. In order to detect potential UCID patterns, we observe total orders of activities with data relations and map them to UCID patterns. For activities in the same workflow, their total orders can be decided based on control flow. However, control flow is useless in the case of activities in different workflows. Assuming that Estimated Execution Time (earliest start time and latest finish time) of activities can be identified at build time, we can use activities' execution time attribute to identify total orders between activities in different workflows. Based on this assumption, we name our workflow as Time Data Workflow (TDW), an extension of WF-Net with time and data factors. Regarding UCID resolution, we take advantage of fusion features of Petri Net to create new workflows with UCIDs removed. In order to apply the UCID theory to the CSW Model, we need to model CSWs as TDWs. To implement this task, we will create a data flow skeleton of CSW and develop an algorithm to generate a CSW based on the given data flow skeleton. This draft of CSW will help workflow designers in developing the schedule of the change process. The other steps in developing change schedule, such as estimating activity resources and activity durations will be performed by workflow designers. From Activity Duration Estimates in the schedule, Estimated Execution Time of activities in this CSW can be derived. Now, we can execute potential UCID detection algorithms. If some potential UCIDs are reported, data and control structure of this CSW could be adjusted according to the suggested UCID resolutions.

## 4 Conclusion and Future Work

This thesis is our first efforts in building the CSW Model which is responsible for CSW construction and management. In this thesis, in order to synchronize changes on shared data, we have proposed an approach to solve errors caused by non-deterministic access to shared data by CSWs in a cooperative software development environment:

- We have abstracted this kind of data abnormality as Unintentional Change in In-use Data. In contrast to previous work, we have considered UCID patterns caused by not only the interactions of concurrent activities inside a single workflow, RW/WW intra-UCID, but also the mutual influences between concurrent workflows, RW/WW/UWU inter-UCID.
- We have defined a Time Data Workflow (TDW) to model workflows. TDW is an extension of WF-Net with time and data factors.
- Algorithms which help detect potential intra-UCID patterns and potential inter-UCID patterns in a Concurrent TDW Management System have been developed.
- Potential UCID resolutions have been proposed.
- Finally, we have presented how to apply UCID theory to the CSW Model to detect errors concerning shared data among CSWs at build time.

As future work, we will improve the algorithms and inter-UCID resolutions by paying attention to unique features of the CSW Model. Detecting and resolving this problem at runtime are our next targets. Then, we will continue with the unsolved problems of the CSW Model: access control, CSW construction and finally a plugin as an implementation of the CSW Model.

## References

- [1] Thi Thanh Huyen Phan and Koichiro Ochimizu: Detection of Unintentional Change on In-use Data for Concurrent Workflows. In: Pro-

ceedings of the 2010 International Conference on Software Engineering Research and Practice (SERP 10), pp 277-283. Las Vegas, Nevada, USA (2010)

- [2] Thi Thanh Huyen Phan and Koichiro Ochimizu: Detecting and Repairing Unintentional Change in In-use Data in Concurrent Workflow Management System. In: Proceedings of the International Workshop on Petri Nets and Software Engineering (PNSE10), pp 89-110. Braga, Portugal (2010)
- [3] Masayuki Kotani and Koichiro Ochimizu: Automatic Generation of Dependency Relationships between UML Elements for Change Impact Analysis. Journal of Information Processing Society of Japan, vol. 49, no.7, pp 2265-2291 (2008)