Studies in Workflow Structure and Correctness

Abstract

Meda Hema Sundari
Indian Institute of Management Calcutta, July 2008

Joint Thesis Supervisors: Prof. Amitava Bagchi & Prof. Anup Kumar Sen

A workflow is a graphical representation of a business process. It shows the tasks in the process, the predecessor-successor ordering of the tasks, the agents (*i.e.*, resources) that execute the tasks and the data items that are inputted to the tasks or outputted by the tasks. Special nodes called connectors are employed to depict the process logic. These indicate whether two tasks execute in parallel or in sequence. The tasks pertain to various business functions such as accounting, manufacturing, marketing or operations. The graphical nature of workflows provides a useful mechanism for users to describe and understand business processes and to redesign them for better performance. Business analysts can use workflows to construct scenarios of system behavior and capture the functionality of the systems they want to develop. Process consultants can use them to detect and eliminate bottlenecks in processes. Significant benefits are realized from workflows when they are used to automate processes. Specialized applications packages called workflow management systems provide functionality for deploying workflows.

Two important aspects of workflow management are modeling and verification. No universally accepted standard for modeling workflows exists as yet. Several formalisms are in use, each having its own notational and semantic conventions. An important research issue is the formulation of proper guidelines to help users to choose suitable formalisms that are most appropriate for their requirements.

After a workflow is modeled, it must be verified for correctness before it is put to use. Improper selection of modeling constructs can introduce design defects that can result in incorrect execution. This can lead to financial loss and customer dissatisfaction. Most of the current commercial workflow management systems do not provide much support for workflow verification. When workflows are small and have few tasks, errors can be detected through manual inspection. But workflows that arise in practice sometimes have hundreds of tasks and connectors. To detect errors in such cases, a computer based approach is needed to ensure that all possible scenarios have been taken into account.

Workflows must be verified irrespective of which modeling formalism has been employed to construct them. In this thesis, we consider two aspects of workflow verification, namely control-flow verification and dataflow verification. Control flows from one task to another from the start to the end of the workflow. Control-flow errors result from incorrect design that causes control not to flow smoothly or to flow more freely than desired. For workflows constructed using AND and XOR connectors, we derive analytical results for detecting control-flow errors such as deadlock, lack of synchronization, indefinite looping and inactive AND-join connector. We consider workflows that are loop-free and also those that have loops. The loop-free workflows have nested, non-nested mesh or standard overlapping structures. A graph traversal algorithm that makes use of the analytical results has also been implemented and test run successfully on workflows of arbitrary structure.

The set of connectors has been extended to make workflow representation more expressive, natural and compact. Existing verification approaches do not consider the problem of detection of control-flow errors in workflows that use these connectors. We generalize the analytical results derived for AND/XOR connectors and address correctness issues of workflows that make use of connectors like Inclusive OR, Synchronizing OR and Multiple Execution OR.

Dataflow verification has not yet received much attention from researchers. Tasks need data to execute. They also generate data after execution. The output data of a task can be fed to subsequent tasks as input. Errors sometimes arise in the flow of data causing data to be unavailable, inconsistent or redundant. There is a need to develop techniques for identifying and detecting all possible types of dataflow errors. We present an algorithm for detecting dataflow errors such as missing data, lost data, redundant data and redundant data in loops. We allow the revision of data by tasks. Our algorithm successfully detects errors in workflows with interconnected loops. It has been test run on many different workflows. Our study implies that it might be possible to develop a unified method for detecting control-flow and dataflow errors at the same time. We expect our error detection methods to prove useful to different categories of users including business analysts and process consultants.