제 27 회 한국정보처리학회 춘계학술발표대회 논문집 제 14 권 제 1 호 (2007. 5)

소프트웨어 프로세스를 위한 워크플로우 작성 엔진의 구현

카델 프라카쉬, 박준영, 최호진
한국정보통신대학교 공학부
e-mail : {prakash,nitra,hjchoi}@icu.ac.kr

Implementation of Workflow Composition Engine for Software Process

Prakash Kadel, JunYoung Park, Ho-Jin Choi
School of Engineering, Information and Communications University, Republic of Korea
{prakash,nitra,hjchoi}@icu.ac.kr

Abstract.

Workflow composition and management is a crucial part of Component Based Development (CBD). Workflows can be composed reusing existing workflows to improve efficiency during the various stages of software development process. Reusability of software process activities is improved by using existing workflows. This paper proposes a Workflow Composition Engine that generates workflow which consists of existing process activities and describes design of The Workflow Composition Engine. A prototype of the Workflow Composition Engine implemented is described.

1. Introduction[1]

CBD aims to enhance reusability, efficiency, and productivity but has several constraints. Mass of components is smaller than expected to leverage the use of the components and development environment to share the components efficiently is not provided.

One of main research issues in CBD is to retrieve components or artifacts of software development process that are stored in distributed repositories and use it to compose other applications or artifacts.

As one of artifacts, process activities used to produce software can be reused to form a new workflow for other software and an efficient and effective approach is needed.

This paper design and implement a Workflow Composition Engine to produce workflows for other software. This Workflow Composition Engine builds an ontology model for software processes described in RDF and infers new workflows from the ontology model.

The rest of this paper is organized as follows: section 2 describes related works and section 3 describes features and structure of Workflow Composition Engine we designed. Implementation of the Workflow Composition Engine is described in section 4 and section 5 shows results of the implementation. Section 6 summarizes the paper and addresses further research.

2. Related Works

This section describes related technologies for the Workflow Composition Engine.

2.1 Workflow

Workflow is a computerized facilitation or automation of a business process in whole or part [1]. Workflow describes how tasks are structured, who performs them, what their order is, how they are synchronized, how information flows to support them, and how they are tracked.

A workflow basically can be defined by three parameters:

- input description

[1] This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement).

- transformation rules, algorithms
- output description

Workflows can only be plugged together if the output of one previous (set of) workflow(s) is equal to the mandatory input requirements of the following workflow.

2.1 Software Process

Software process can be considered as a procedure developers follow to produce software in an organization. There are lots of software processes available for developers. However, processes have common structure that is comprised of several tasks and activities.

Figure 1 shows steps of ISPW-6[2] software process example.

- Develop Change and Test Unit
- Schedule and Assign Tasks
- Modify Design
- Review Design
- Modify Design
- Modify Test Plans
- Modify Unit Test Package
- Test Unit
- Monitor Progress

Figure 1 tasks to perform in ISPW-6 software process example

Software processes are composed of tasks, which has prerequisite tasks, outputs, and activities as properties. Figure 3 depicts the concept of process and task.

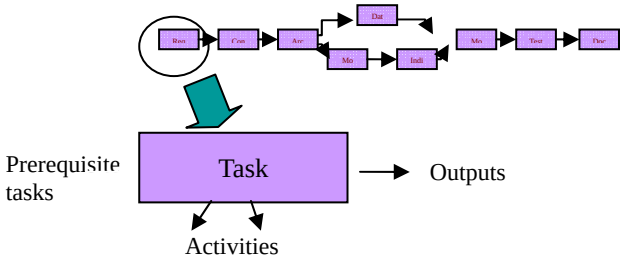

Figure 2 concept of task

2.2 CBD

Objective of Component-based Software Development (CBD) is to develop independent software components and compose new applications using the components developed in the previous projects or acquired from suppliers. As a result of this approach, it was expected that reusability, efficiency, and productivity would improve but significant changes to solve the software crisis have not shown.

The reason of this result is that the number and the amount of the components were not large to achieve "Critical Mass" [3]. In other words, there was not enough components shared among developers and development environment to help developers share those components was not provided.

Processes are useful asset that can be reused by developers. Using CBD methodology, developers can reuse not only executable components but also artifacts that are produced in the other projects to produce artifacts that are similar to previous ones.

2.2 Ontology

Ontology is a data model that represents a domain and is used to reason about the objects in that domain and the relations between them. Ontology generally describes individuals (instances), classes (concepts), attributes, and relations.

Several researches on software component repositories based on ontology are ongoing and semantic web languages like RDF and OWL enables to access ontology and ontology repository [4]. RDF provides common structures that are used for interoperable XML data exchange [5].

For example, tasks of software process can be represented in RDF. Information of a task that has name, outputs, prerequisite tasks, and activities as properties can be described as the following example.

```
…
</kb:Task>
<kb:Task rdf:about="&kb;Task_Instance_2"
kb:Name="Requirement Analysis"
kb:outputs="Requirement Specification
Document"
rdfs:label="Task_Instance_2">
<kb:prerequisite_tasks
rdf:resource="&kb;Task_Instance_0"/>
<kb:activities>Cost Benefit
Analysis</kb:Activities>
<kb:activities>Interactions with End
User</kb:Activities>
</kb:Task>
…
```

Figure 3 example of RDF description

3. Workflow Composition Engine

Objective of this research is to propose a way to better compose workflow for CBD process. To achieve the objective we made a prototype of Workflow Composition Engine that can produce (plan) workflows for developing specific software. This Chapter explains structure and functionality, features, data and control flow, and data structure of the Workflow Composition Engine.

The structure of the workflow engine is shown as below.

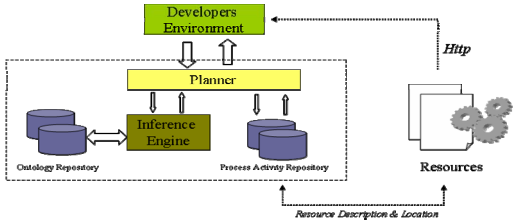

Figure 4 Structure of Workflow Composition Engine

The Workflow Composition Engine consists of Ontology Engine, Inference Engine Process Activity Repository, and

Developer Environment. In addition, Web Resources described in Process Activities Repository is required.

Process activities of software processes are described and stored in Process Activity Repository. Ontology for the software processes are constructed in Ontology Repository by Inference Engine from Process Activity Repository. Development Environment is workplace that user withdraws workflows asking Inference Engine. When a new workflow was suggested by Inference Engine, user can retrieve details of process activities which are described in RDF and stored in Process Activity Repository accessing Web Resources.

3.1 Features of Workflow Composition Engine

The Workflow Engine we made has several features as the following:

- Modular Architecture
- Generic parameter for adaptation to changes
- Standard ADT
- Extensibility
- Up-to-date with recent related researches

3.2 Data and Control Flow

Data and control flow of the Workflow Composition Engine is defined as the following.

First, user chooses Project Name that he/she wants to generate workflow. Second, user selects Project Goal to see the workflow for the goal. Third, if there is a Choice Point (CP) that user should decide which one to follow; the system shows the CP and gets user input. Once the CP is determined, the engine constructs a graph for the model. Fourth, the Workflow Composition Engine processes the graph for inconsistencies using planning, DFS, and backtracking. Finally, the workflow in graph is displayed.

Figure 5 shows procedures and data and control flow of workflow generation.

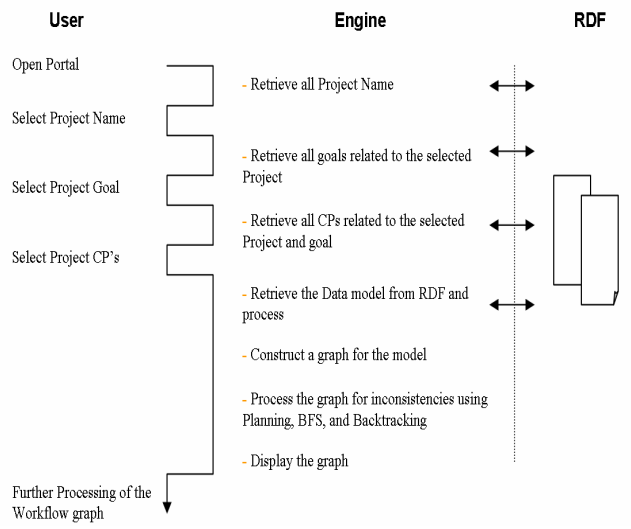

Figure 5 Interactions between user and Workflow Engine

Figure 6 has two different paths from start to end. The paths are different after "Requirement Analysis" and "Requirement Analysis" is the Choice Point of this workflow.

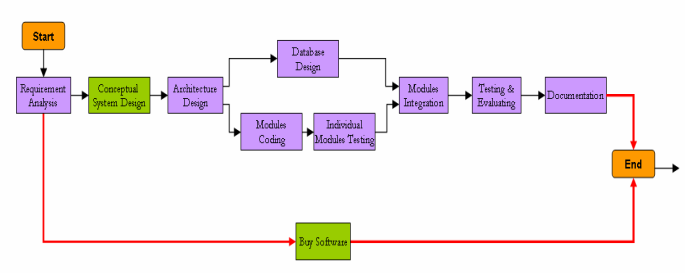

Figure 6 Example of Choice Point Selection

3.3 Data Structure

It is important to construct a graph that shows the path from start to end of the path to achieve specific goal. The Workflow Composition Engine constructs a graph represented in symbol table of vertices. To figure out a path in the workflow is broken or not, we used DFS in implementing backtracking technique.

4. Implementation

The Workflow Composition Engine was implemented by Java and Jena was used to access Ontology Repository constructed in RDF. Jena provides Java APIs for ontology process and was used in Eclipse IDE together.

To construct Ontology Repository for processes, Protégé was used. An example repository that has sample process activities was created using Protégé. Because constructing Process Activity Repositories is out of the scope of this project and a set of process activities in a file was given as a repository to the Workflow Composition. RDF was used to describe the location and properties of the process activities stored in the file.

Figure below is a sample scenario that user gets a new workflow produced by the Workflow Composition Engine.

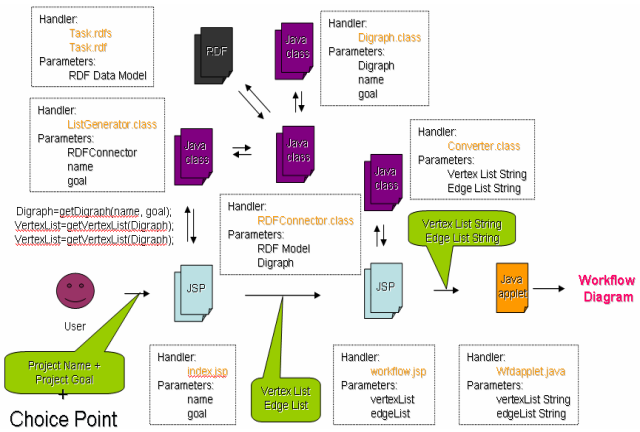

Figure 8 Sample Scenario of using Workflow Engine

The Workflow Engine is run on http server. When user connects the system, it shows a web page that requires user inputs related to workflow. If user chooses a project name

and project goal, system finds the goals that match that user choose. It builds the process activity graph and finds Choice Point to determine which path to follow. If user selects the Choice Point, the Workflow Engine prepares new workflow.

5. Results

We implemented prototype of the Workflow Composition Engine we designed. This chapter explains how to use the system implemented as a result of this research project.

First, user connects the web server that runs the Workflow Composition Engine and selects three kinds of information in turn. The information is "Project ", "Project Goal", and "Choice Point". The system requires selection for "Choice Point" only when the "Project Goal" can be achieved by flowing more than one path.

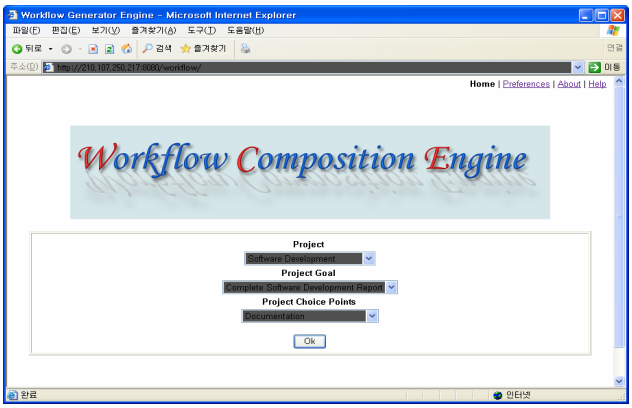

Figure 9 User selects three options needed to generate workflow

After user finished selecting information, the system displays workflow that generated as what user input. Figure 10 is the result of user input shown in Figure 9. It shows 9 tasks are required to produce "Complete Software Development Report" when user selected "Documentation" as Choice Point. In case user input another Choice Point, the system generates alternative workflow for the Choice Point. If user selects "Buy Software" instead of "Documentation" for Choice Point, there are only 2 tasks generated by the Workflow composition Engine (see Figure 11). Workflows of these Figures 10 and 11 are different from each other because of the Choice Points "Documentation" and "Buy Software" after "Requirement Analysis".

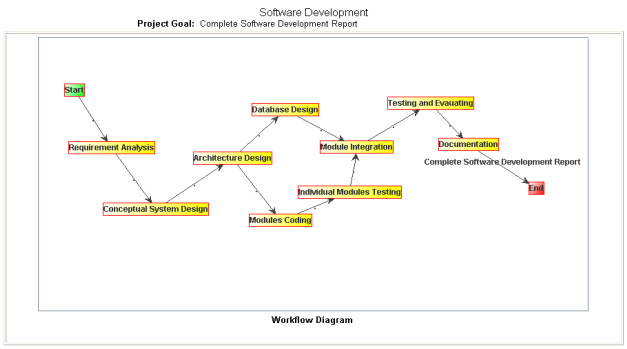

Figure 7 Workflow generated when user choose Choice Point "Documentation"

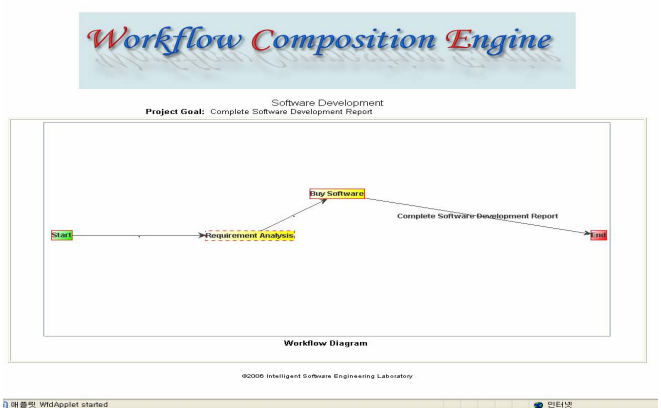

Figure 8 Workflow generated when user choose Choice Point "Buy Software"

6. Conclusion and Future Works

In this paper, a Workflow Composition Engine that can be used to generate a new workflow reusing existing process activities was designed and implemented. The result of implementation shows that the Workflow Composition Engine can help make new workflows needed for developers. The Workflow Composition Engine composes workflows that fit user requirements by finding paths of software tasks to the Project Goal.

The system asks Choice Point to user when more than one paths exist to the Project Goal. However, there are some limitations in determining Choice Points when lots of alternatives are identified.

Further improvement is necessary in implementation of the Workflow Composition Engine and further research on the reusability of components is required.

References

[1] David Hollingsworth, "The Workflow Reference Model", Workflow Management Coalition, 19 Jan 1995.

[2] Mark I. Kellner, et al., "ISPW-6 Software Process Example", IEEE, 1991.

[3] Szypersky, C., Gruntz D., Murer S., "Component Software: Beyond Object-Oriented Programming", 2nd edition, ACM Press, 200

[4] Regina M., M. Braga et al. "The Use of Mediation and Ontology Technologies for Software Component", Information Retrieval, ACM, 2001.

[5] "Practical RDF", O'Reilly, 2003.