워크플로우 지향 도메인 분석

Workflow Oriented Domain Analysis

김영철*, 김윤정**

홍익대학교 컴퓨터정보통신 소프트웨어공학연구실*, 한국과학기술정보연구원**

Young-Chul Kim(bob@selab.hongik.ac.kr)*, Yun-Jeong Kim(miso@kisti.re.kr)**

요약

본 논문에서는 레거시 시스템에 대한 기존 도메인 분석의 문제점을 해결하기 위하여 동적 모델링을 기반으로 하는 확장된 워크플로우 메커니즘을 이용한 도메인 분석 방법론을 제안한다. 이 방법론을 WODA(Workflow Oriented Domain Analysis)라 명명한다. 제안하는 절차를 통해 공통/비공통 컴포넌트를 식별 및 컴포넌트들의 클러스터를 추출할 수 있다. 이를 통해 새로운 시스템을 개발 시 효율적으로 재사용하고자 한다. 동적 분석으로 특정한 시스템에 발생 가능한 시나리오들을 식별한 후, 제안한 컴포넌트 테스트 플랜 매트릭스를 이용해 재사용성이 높은 컴포넌트와 컴포넌트 시나리오를 결정한다. 또한 컴포넌트 가중치 측정을 통해 재사용 가능한 컴포넌트들의 중요성과 빈도수를 인식하고 컴포넌트 시나리오들의 우선순위를 도출 할 수 있다. 구현한 자동화 모델링 도구인 WODA을 통해 UPS(Uninterrupted Power Supply)에 적용 사례를 소개한다.

■ 중심어 : | 도메인 분석 | 워크플로우 | 컴포넌트 기반 개발 | 컴포넌트 테스트 플랜 매트릭스 |

Abstract

In this paper we will propose a domain analysis methodology that uses an extended workflow mechanism based on dynamic modeling to solve problems of a traditional domain analysis on legacy systems. This methodology is called WODA(Workflow Oriented Domain Analysis). Following procedures on WODA, we can identify common/uncommon component, and also extract the cluster of components. It will be effectively reusable on developing new systems with these components. With our proposed component testing metrics, we can determine highly reusable component/scenario on identifying possible scenarios of the particular system. We can also recognize most critical / most frequent reusable components and prioritize possible component scenarios of the system. This paper contains one application of UPS that illustrates our autonomous modeling tool, WODA.

■ keyword : | Domain Analysis | Workflow | CBD | Component Test Plan Metrics |

* 본 논문은 2003년도 홍익대학교 학술연구 조성비에 의하여 연구되었습니다.

접수번호 : #051226-001

접수일자 : 2005년 12월 26일

심사완료일 : 2006년 01월 17일

교신저자 : 김영철, e-mail : bob@selab.hongik.ac.kr

I. 서 론

현재 유비쿼터스에 대한 연구가 활발히 진행 중이며 앞으로 우리 사회는 유비쿼터스 환경이 될 것이다. 유비쿼터스의 의미는 언제 어디서나 원하는 시간에 사용자가 네트워크에 연결되어 서비스를 지원받을 것이다. S/W 컴포넌트들로 구성된 애플리케이션이 수행하는 것이 바로 서비스가 될 수 있다고 본다. 즉, 유비쿼터스 기술에 있어서 네트워크 환경과 더불어 요구되는 것이 바로 소프트웨어 기술이다. 그 소프트웨어 기술의 근간이 컴포넌트 기반 개발일 것이다. 양질의 소프트웨어를 제공하기 위해 업계와 학계에서 활발하게 연구가 진행 중인 것이 컴포넌트 기반 소프트웨어 개발 방법론이다.

컴포넌트란 재사용, 재배치가 가능하며 변화에 대하여 관리가 가능한 독립적인 단위의 소프트웨어라 할 수 있다. 컴포넌트는 잘 정의된 인터페이스를 통해 서비스를 제공하고 응집력과 결합력을 갖는다. 그러나 컴포넌트에 대한 크기는 명확하게 정의되지 않았다. 따라서 고객이 원하는 소프트웨어의 개발을 위해서는 고객과 개발자 간의 상담을 통하여 요구사항을 분석하고 모델링 하는 기술이 필요하다.

이 논문에서는 하향식의 개발 방법을 통해 고객이 원하는 크기의 컴포넌트를 추출하고, 시나리오 분석과 같은 행위 분석을 적용하고자 한다. 즉, 컴포넌트 기반 개발의 이점과 동적인 분석, 하향식 방식으로 컴포넌트를 추출한다[1]. 컴포넌트를 추출하기 위해 시스템들의 개발과 유지관리 효율성을 높이기 위한 재사용 그리고 시스템들과 애플리케이션들에서의 정적 구조와 동적 행위 추출에 목적을 둔 도메인 분석을 이용한다. 시스템을 분석하는 기본 모델링 기술로 정보와 정보의 흐름의 표현으로 시스템을 기술하는 워크플로우 모델을 이용한다. 워크플로우는 BPM(Business Process Management)에서 시스템적으로 구성하는 핵심이기도 하다. 이 논문에서는 워크플로우 메커니즘을 통해 소프트웨어 시스템을 도메인 분석하여 공통/비공통의 컴포넌트를 추출하고, 컴포넌트 클러스터와 중요도에 따른 컴포넌트를 통해 시스템 개발 시 효과적으로 재사용할 수 있는 방법에 대해 기술하고자 한다. 이를 위해 시스템에서 발생할 수 있는 전체 시나리오를 결정한 후 컴포넌트들마다 할당된 가중치 값을 통해 재사용성이 높은 컴포넌트를 결정한다. 이는 컴포넌트 테스트 플랜 매트릭스를 이용하여 결정 가능하다.

본 논문은 다음과 같이 구성되어 있다. 2장과 3장에서 확장된 워크플로우 메커니즘과 이를 이용한 도메인 분석 방법론에 관하여 소개한다. 제안한 방법론의 적용 사례를 4장에서 언급하고, 5장에서 시나리오의 확률 값에 따른 신뢰성을 확인한다. 6장에서는 모델링 도구로 구현한 WODA(Workflow Oriented Domain Analysis) 도구를 통해 컴포넌트의 추출을 확인한다. 마지막으로 7장에서 결론과 향후 연구방향에 대하여 기술한다.

II. 확장된 워크플로우 메커니즘

본 논문은 현존하는 시스템인 레거시 시스템에 대한 도메인 분석을 제안하고자 한다. 이는 재사용할 수 있는 자산들(컴포넌트)을 추출하고, 각각의 재사용 컴포넌트를 품질 평가하여 양질의 컴포넌트로 새로운 고품질의 시스템을 생산할 수 있게 된다.

도메인 분석의 대상이 레거시 시스템과 새로운 시스템일 때 그 둘 사이에는 차이점이 있다[2]. 레거시 시스템에서 도메인 분석을 위해 보편화되어 있는 모델링 기술 방법들인 State Diagrams, State Charts, Use Case Diagrams, Sequence Diagrams, Collaboration Diagrams, Activity Diagrams, Colored Petri-Net 등은 객체나 함수 레벨과 같이 너무 작은 규모의 레벨에서 시스템을 모델링 하는 단점이 있다. 수준이나 규모를 다르게 정의할 수 있는 개념이 필요하여 워크플로우 개념을 도입하였다.

워크플로우란(비즈니스) 프로세스 모델링의 일종으로 시스템 또는 애플리케이션에 대해 실질적인 목적 수행을 위한 동적 작업 변화의 연속적인 흐름을 나타내는 방법이다[3]. 비즈니스 프로세스도 워크플로우 개념으로 이루어진다[4]. 즉 워크플로우는 어떤 조직에 있어서 비즈니스 프로세스의 실현을 이루는 요소이다.

워크플로우 모델링의 장점[3]은 비즈니스 프로세스의

협업 구조를 모델링 하는데 적합하다는 것이다. 따라서 본 논문에서는 확장된 워크플로우 메커니즘을 통해 도메인 분석을 제시하고자 한다. [그림 1]는 확장된 워크플로우 메커니즘에 필요한 요소들을 도식화한 것이다.

제안한 방법론은 하향식 방식으로 분석하면, 상위 레벨의 워크플로우가 반복적, 점증적 분석을 통해 하위 워크플로우로 이루어진다.

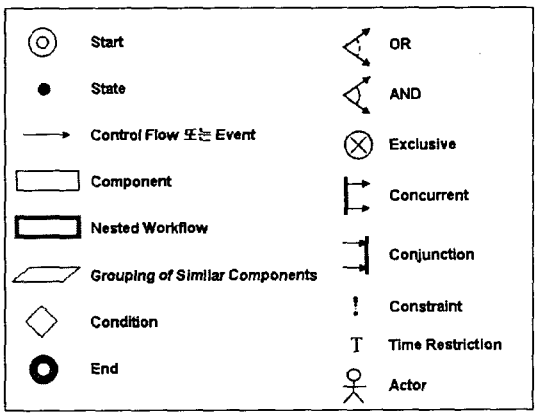


그림 1. 확장된 워크플로우 메커니즘의 Agenda

(1)시작(Start) : 전체 Activity들의 개시
(2)상태(State)
(3)이벤트(Event) : Activity를 수행하는 동안 발생하는 것
(4)컴포넌트(*Component*) :워크플로우에서 임무를 수행하는Activity
(5)내포된 워크플로우(Nested Workflow) : 내포(Nested) 개념으로 하위 워크플로우를 가질 수 있다.
(6)그룹핑 컴포넌트(Grouping Component) : 유사한 컴포넌트들의 그룹화
(7)조건(Condition) : 분기로써 다른 상태로 전이할 수 있는 조건
(8)종료(End) : 전체 Activity들의 완료
(9)전이(Transition) : 제어 흐름으로써 한 상태에서 다른 상태로 변환. 병렬 개념을 지원하는 AND, OR, Exclusive 논리연산과Concurrent, Conjunction 개념 포함.
(10)제약사항(Constraint) : 임무 수행 시 제한사항
(11)시간 제약(Time Restriction) : 시간 제약을 할 필요가 있는 임무에 설정
(12)행위자(Actor) : Activity를 수행하는 부서 또는 부서의 업무 담당자

그리고 내포(Nested) 개념으로 하위 워크플로우를 가질 수 있는 것을 'Nested Workflow'로 정의한다. 또한 입력 값과 출력 값이 다르지만 유사한 기능을 하는 프로세스들을 그룹화 할 수 있다.

[그림 1]에서 병렬 개념을 지원하는 AND는 여러 컴포넌트가 모두 수행되어야 다음 단계로 전이하고, OR는 여러 컴포넌트 중 하나 이상의 컴포넌트가 수행되어야 다음 단계로 전이하는 것을 의미한다. Exclusive는 여러 컴포넌트 중 하나의 정확한 컴포넌트가 전달되었을 때 다음 단계로 전이하며, Concurrent와 Conjunction은 바로 전 단계의 컴포넌트가 동시에 모두 수행되어야만 다음 단계로 전이한다.

III. 워크플로우 메커니즘을 이용한 도메인 분석 방법론

이 장에서는 앞 장에서 언급한 것과 같이 시스템에 대해 실질적인 목적 수행을 위한 동적 작업 변화의 연속적인 흐름을 나타내며 비즈니스 프로세스의 협업 구조에 적합한 워크플로우 메커니즘을 이용한 도메인 분석 방법을 언급하고자 한다. 컴포넌트를 추출하기 위한 표준화된 절차를 제시함으로써 사용자가 원하는 크기의 컴포넌트를 찾을 수 있다. 이 방법은 [5]를 확장된 워크플로우 메커니즘을 이용한 도메인 분석 방법론(Workflow Oriented Domain Analysis)으로 아래의 절차를 따른다.

- 1 단계 : 워크플로우 메커니즘을 이용한 도메인정의.
  적용할 특정한 도메인에 대해 행위자, 프로세스, 상태 등을 이용한 상위 레벨의 워크플로우 모델링을 통해 도메인 정의.
- 2 단계 : 워크플로우 메커니즘을 이용해 도메인분석.
  적용할 특정한 도메인에 대해 행위자, 프로세스, 상태 등을 적용한 워크플로우 모델링을 통해 도메인 분석.
- 3 단계 : 워크플로우 메커니즘을 이용해 도메인 구조.
  전체시스템에 대한 프로세스 모델을 제시.
- 4 단계 : 워크플로우 메커니즘을 이용해 도메인 설계.
  전체 시스템에 대한하향식(Top-down) 및 계층적 메커니즘을 통해 필요한 워크플로우 모델링을 설계.
- 5 단계 : 컴포넌트 추출.
  5-1단계 : 4단계를 통해, 공통/비공통의 컴포넌트들을 추출.
  5-2단계 : 컴포넌트 가중치 측정 매트릭스를 통해 컴포넌트들의 반복 및 중복을 찾음.
  5-3단계: 각각의 컴포넌트에 대해 하위 워크플로우(또는 내포된 워크플로우)를 찾는다.

1단계부터 5단계 반복 수행(원하는 크기의 컴포넌트를 찾을 때까지).

- 6 단계 : 컴포넌트 설계.
- 7 단계 : 컴포넌트 구현.
- 8 단계 : 컴포넌트 내의 객체 추출을 위해 객체지향 방법론 적용.
- 9 단계 : 컴포넌트 테스트

이 절차를 적용하면, 동적 행위 흐름을 모델링 한 하나의 상위 레벨의 워크플로우 모델링이 점진적, 반복적 분석을 통해 하위 워크플로우 또는 내포된 워크플로우들로 구성되어 있음을 알 수 있다. 기존의 UML(Unified Modeling Language) 분석 단계를 수행하여 객체를 추출할 수 있고, 마지막 단계에서 컴포넌트 테스트를 수행한다. 방법론의 절차를 시각화한 것이 [그림 2]이다.

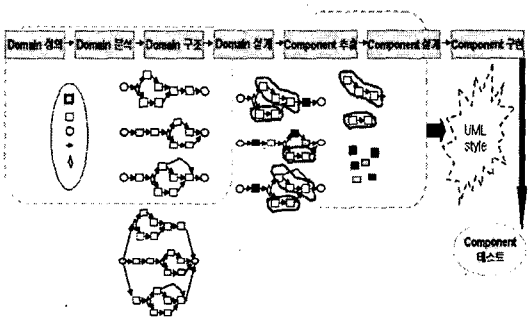


그림 2. 워크플로우 메커니즘을 이용한 도메인 분석 방법론 절차

기존의 컴포넌트 기반 방법에서는 단지 컴포넌트의 재사용에 초점을 두고 있다. 우리는 복잡한 국방 소프트웨어 컴포넌트 분석에서 많은 서브시스템들이 공통의 컴포넌트와 연속되는 컴포넌트군(이를 컴포넌트 클러스터라 함)도 함께 사용하고 있다. 그리고 단지 하나의 서브시스템에 사용되지만 없어서는 안 되는 중요한 컴포넌트를 찾게 되었다[5]. 우리는 이를 비공통 컴포넌트라 명명하였다.

그래서 우리의 개발 절차에서도 컴포넌트 추출과정에서 공통/비공통의 컴포넌트를 추출하고 이를 그룹화 하여 컴포넌트 클러스터를 찾는다. 이는 컴포넌트 시나리오 경로를 측정함으로써 가능하다. 컴포넌트 시나리오 경로는 시스템의 동적 분석을 바탕으로 작성할 수 있다.

즉 워크플로우 개념의 동적 분석으로 전체 시스템내의 컴포넌트를 식별하고, 시스템에서는 수행되는 컴포넌트들의 순차적인 순서(이를 컴포넌트 시나리오라 함)를 결정하고 시나리오들을 우선적으로 선별한다. 이런 다음 컴포넌트들마다 할당된 가중치 값과 분석시 설정, 부여한 우선순위를 통해 재사용성이 높은 컴포넌트와 중요한 컴포넌트를 찾는다.

시스템에서 발생 가능한 전체 시나리오들을 통해 산출되는 경로 상의 컴포넌트들에 길이, 중요성, 재사용성별로 컴포넌트 가중치를 설정한다.

이를 통해 가중치가 큰 컴포넌트는 그 만큼, 유사 시스템을 개발할 때 재사용의 효율성을 높일 수 있다.

표 1. 컴포넌트 테스트 플랜 매트릭스 (◆ : NOT)

	컴포넌트 경로의 측정		컴포넌트 가중치(w)
길이 (Length)	최단 경로(Shortest path) : Least steps of Components		w = 1
	최장 경로(Longest path) : Most steps of Components		
중요성 (Criticality)	중요 컴포넌트 (Most critical(frequent) path)		w ≥ 1
	최소 컴포넌트 (Least critical path)		w ≥ 0
재사용성 (Reusability)	컴포넌트	대부분 재사용되는 컴포넌트 (Most reusable steps)	w > 1
		최소로 재사용되는 컴포넌트 (Least reusable steps)	w ≥ 0 and ◆ = 1
	컴포넌트 클러스터	대부분 재사용되는 서브 컴포넌트 클러스터 (Most reusable sub-path)	w > 1

이는 [표 1]의 컴포넌트 테스트 플랜 매트릭스[6][7]를 이용하여 구할 수 있다. 가중치를 설정하기 위해 고려하는 세 항목은 다음과 같고, 이 논문에서는 컴포넌트 가중치 측정 매트릭스 중 재사용성 항목만 언급하겠다.

첫째, 길이의 핵심은 도메인 분석을 위하여 가장 짧은 클러스터와 가장 긴 클러스터의 두 가지 관점이다. 그러나 우리가 매트릭스의 다른 카테고리와 함께 이 이슈를 사용하는 경우에만 유용하다. 둘째, 중요성의 핵심은 컴포넌트 시나리오들의 리스트를 선택하기 위해 중요하다. 셋째, 재사용성의 핵심 또한 재사용 가능한 컴포넌트들을 식별하고 극대화하기 위해 중요하다.

그러나 현재의 프로덕트 라인 방법론에서의 휘처 모델링[10]은 가변성과 공통성에 초점이지만 사실 비공통 컴포넌트 대한 중요성을 간과하는 경향이 있다.

또한 [그림 3]은 워크플로우의 정적인 분석을 위해 구조적으로 모델링을 보여 준다. 이는 [8]의 컴포넌트 명세를 확장하여 작성하였다.

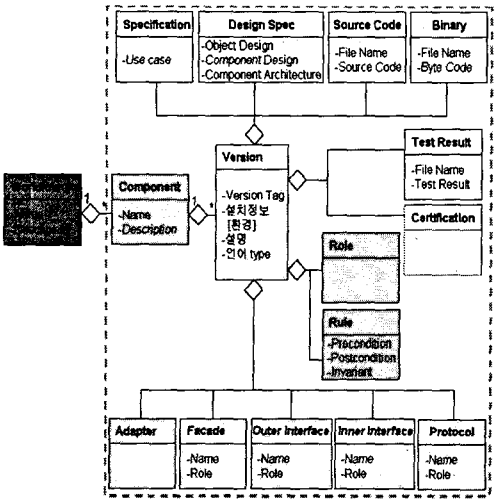


그림 3. 구조적인 워크플로우 모델링

Ⅳ. 워크플로우 지향 도메인 분석(WODA) 단계별 적용 사례

이 장은 앞 장에서 언급한 방법론을 이용하여 상위 레벨 워크플로우 기반의 컴포넌트 모델링을 통해 상위 레벨의 컴포넌트를 정의하고 추출한 후, 점진적, 반복적으로 하위 레벨의 모델링을 통해 사용자(개발자, 설계자, 시험자 등)에게 맞는 크기의 컴포넌트를 추출하는 것에 대해 설명한다. 모델링 적용 예로써 입력 전원의 전압 변동과 정전 및 기타 장애의 발생에 관계없이 항상 부하에 양질의 전력을 공급하는 장치인 UPS (Uninterruptible Power Supply)를 들어 하나의 시스템에서 재사용성이 높은 컴포넌트를 추출하는 것을 제시한다. 3장에서 설명한 개발절차의 산출물은 총 9단계 13활동 27 산출물로 구성된다[9]. 이 논문에서는 이 중 몇 단계의 산출물을 UPS 분석의 산출물로 보이고자 한다.

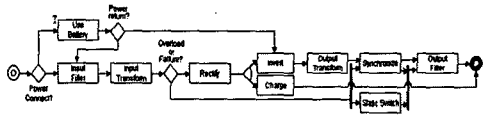
1. 도메인 설계

프로세스 명세까지 마친 후 전체 시스템에 대한 하향식 및 계층적 메커니즘을 통해 필요한 도메인을 설계한다.

◆ 도메인 설계서

설계할 도메인에 대하여 상위 레벨로 모델링한 후 하향식 분석을 통해 하위 레벨로 모델링하고 주의사항을 기입한다. [표 2]처럼 도메인 설계서를 작성한다.

표 2. 도메인 설계서

<table>
<tr><td rowspan="3">도메인 설계서</td><td>단계</td><td>도메인 설계</td></tr>
<tr><td>활동</td><td>도메인 설계</td></tr>
<tr><td>작업</td><td>하향식 도메인 설계</td></tr>
<tr><td colspan="3">프로젝트명 : UPS 모델링을 통한 컴포넌트 추출</td></tr>
<tr><td colspan="3">도메인 모델링</td></tr>
<tr><td colspan="3">상위레벨
하위레벨
  </td></tr>
</table>

UPS에서 발생 가능한 워크플로우 시나리오는 [그림 4]에서와 같이 다섯 가지이다. 각 워크플로우 시나리오별로 컴포넌트에 기입된 숫자가 해당 시나리오의 경로 순서를 나타내며, 타원, 모서리가 둥근 직사각형, 마름모, 직사각형은 컴포넌트가 다섯 개의 시나리오에 동일한 횟수로 사용된 것을 나타낸다. 모델링 결과로 찾은 각각의 컴포넌트가 얼마나 사용되고 있는지 [그림 4]를 통해 알 수 있다.

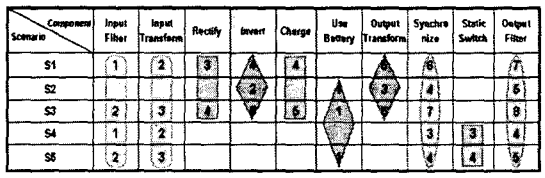


그림 4. 워크플로우 시나리오별 컴포넌트 사용 수

S1 : 정상 상태
S2 : 정전 시
S3 : 정상 상태로 복구
S4 : 과부하 및 인버터 이상
S5 : 정상 상태로 복구된 뒤 인버터 이상

[그림 5]는 각 컴포넌트마다 가중치를 1로 설정한 뒤 재사용성에 따른 값을 계산하여 이 중 사용 빈도수가 많은 컴포넌트들이 포함된 시나리오를 찾는 것을 보여준다.

	Input Filter	Input Transform	Rectify	Invert	Charge	Use Battery	Output Transform	Synchro -nize	Static Switch	Output Filter
Weighted value	1	1	1	1	1	1	1	1	1	1
Reusability weight	4	4	2	3	2	3	3	5	2	5

	S1	S2	S3	S4	S5
#5	2	2	2	2	2
#4	2	0	2	2	2
#3	2	3	3	0	1
#2	2	0	2	1	1
Sum	28	19	31	20	23

그림 5. 재사용성에 따른 중요 컴포넌트 측정
(#숫자 : Reusability weight)

[그림 5]의 좌측 열은 시나리오에 사용되는 컴포넌트의 수 즉, 재사용성 값을 나타낸다. 시나리오별로 사용된 컴포넌트들의 재사용성 값의 합으로 재사용성이 높은 시나리오를 찾는다(S1의 경우, 5*2+4*2+3*2+2*2=28). [그림 5]에 명기된 전체 5개의 시나리오 중 S3(정전 후 정상 상태로 복구)은 나머지 네 개의 시나리오에 비해 재사용성가중치의 합이 커 재사용성 값이 큰 컴포넌트들을 사용한 워크플로우 시나리오로 볼 수 있다.

2. 컴포넌트 테스트

컴포넌트 자체를 테스트하여 컴포넌트 결과서를 작성하고, 컴포넌트를 통합 테스트하기 위해서 통합 테스트 설계서를 작성한 후 테스트를 수행하여 통합 테스트 결과서를 작성한다. 시스템 통합이 필요할 때 또한 테스트 설계서를 작성하고 시스템 통합 테스트 수행 결과를 작성한다. 사용자와 운용자에게 컴포넌트를 인도하기 전 사용자 지침서와 운용자 지침서를 작성하는 것으로 컴포넌트 테스트 단계를 마친다.

◆ 컴포넌트 테스트 결과서

테스트케이스별로 테스트를 수행한 후 예상 결과 및 수행 결과를 기입하고 오류 내용과 수정 여부를 작성한다.

제안된 방법을 통해 컴포넌트의 효율적인 재사용 및 중요도를 측정할 수 있으리라 기대한다. 또한 이제까지 언급한 사례에서는 앞에서 살펴본 컴포넌트 가중치 측정 매트릭스 중 재사용성에 초점을 맞추었다. 얼마나 많이 재사용되는가에 따른 빈도수 체크보다는 얼마나 중요한 컴포넌트인가를 분석하는 것이 필요하다. 따라서 컴포넌트 가중치 측정 매트릭스의 중요성(Criticality) 항목에 대한 평가가 이루어져야 할 것이다.

V. 워크플로우 시나리오의 우선순위

이 장은 데이터 수집 또는 추측방법으로 얻을 수 있는 데이터를 통해, UPS 시스템 모델링내의 경로 상의 Control flow에 발생 확률 값을 할당하고 이를 통해 전체 시스템의 워크플로우 시나리오들[그림 6]과 그 시나리오들의 발생 가능 확률 값을 통해 모든 워크프로우 시나리오들의 우선순위를 찾고자 한다.

다음은 총 발생 확률 계산하는 공식으로써,

$\forall i$ Scenario i in Workflow $\subseteq$ A Workflow W
(조건: W는 a UPS Application임)

워크프로우내의 모든 시나리오들에 대해,

특정한 시나리오 i 는 워크플로우 W 내에 있다.

$\forall i$ component i $\subseteq$ A Workflow scenario i;

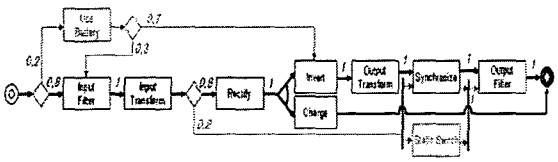
특정한 워크플로우의 시나리오 i 내의 모든 컴포넌트들에 대해,

(Π the weighed factor of Component i * probability of component i) / (Σ probability i) 식을 이용해 총 발생률을 계산한다.

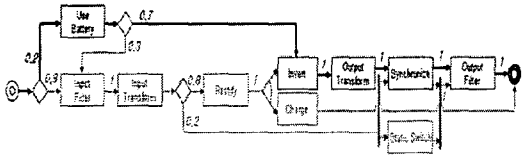
아래 [그림 6]에서와 같이 Control flow마다 할당한 값은 계산을 통해 모든 워크플로우 시나리오의 발생 가능성한 확률 값[그림 7]의 계산이 가능하다. 가장 높은 시나리오(정상 상태)로써 0.64의 높은 값을 갖는다. 다음으로 정전 시(0.14)등등을 갖는다.

이런 워크플로우 시나리오들의 우선순위를 통해 테스트 프로세스 생산성을 향상시킬 수 있다. 예를 들면, 우리는 먼저 제일 높은 워크플로우 시나리오 "정상 상태"

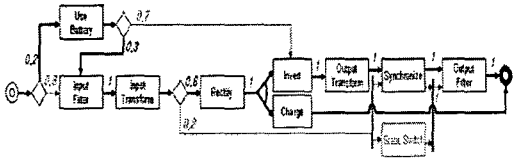
와 두 번째 높은 워크플로우 시나리오 "정전 시"를 테스트함으로써, 전체 시스템의 78%를 보장할 수 있다. 특정 시스템을 78%의 보장한다는 것은 매우 중요한 이슈이다. [그림 6]은 UPS 시스템의 가능한 워크플로우 시나리오들을 보여주고 있다. 이는 테스트 임의의 값을 할당했기 때문에 오차가 있을 수 있으나, 본 논문에서는 오차를 배제한 가정 하에서 값을 계산한다. [그림 7]은 위의 계산 공식에 의해, 가능한 UPS 내의 워크플로우 시나리오들에 대해 계산한 총 발생 확률 값들과 재사용되는 컴포넌트들의 그룹(클러스터)을 보여 주고 있다.



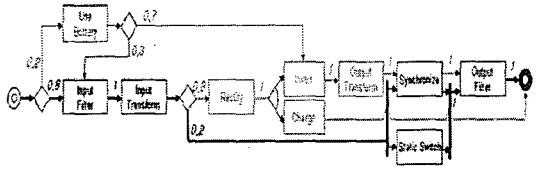
(a) 정상 상태



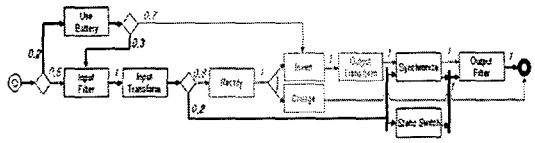
(b) 정전 시



(c) 정상 상태로 복구



(d) 과부하 및 인버터 이상



(e) 정상 상태로 복구된 뒤 인버터 이상

그림 6. 각 시나리오별 발생 확률

	Input Filter	Input Transform	Rectify	Invert	Charge	Use Battery	Output Transform	Synchro nize	Static Switch	Output Filter	전체 발생 확률
정상 상태	1	2	3	4	4		5	6		7	0.8*1*0.8*1*1*1*1 =0.64
정전 시				2		1	3	4		5	0.2*0.7*1*1*1 =0.14
정상 상태로 복구	2	3	4	5	5	1	6	7		8	0.2*0.3*0.8*1*1*1 =0.048
과부하 및 인버터 이상	1	2						3	3	4	0.8*1*0.2*1*1 =0.16
정상 상태로 복구된 뒤 인버터 이상	2	3				1		4	4	5	0.2*0.3*0.2*1*1 =0.012

그림 7. 전체 발생 확률

[그림 7]에서 컴포넌트 클러스터 (input filter, input Transform)은 4개의 시나리오에서 모두 재사용하고, 컴포넌트 클러스터 (input filter, input transform, rectify, invert, charge) 등은 2개의 시나리오에서 사용하고 있을 볼 수 있다.

VI. 구현한 WODA 도구를 통해 컴포넌트를 추출

이 장에서 살펴볼 내용은 시스템을 분석하여 모델링한 결과를 도구를 통해 확인하는 것이다. 도구를 사용함으로써 모델링을 자동화할 수 있고 수작업으로 인해 발생할 수 있는 오류를 줄일 수 있다. 이를 위해 본 논문에서는 WODA 도구를 소개하고, 앞에서 언급한 워크플로우 메커니즘의 적용 사례인 UPS를 도구를 통해 모델링하고자한다.

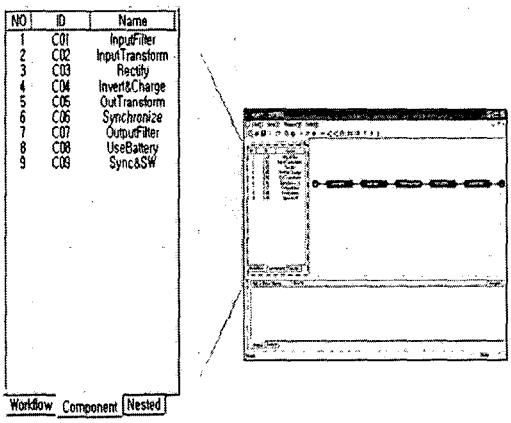


그림 8. Flow를 구성하기 위해 필요한 컴포넌트목록

[그림 8]에서와 같이 컴포넌트들을 생성하면서 이 컴포넌트들을 드래그하여 시스템을 모델링 한다. 상위 레벨의 시스템 모델링을 통해 컴포넌트가 하위 워크플로우로 구성되는 내포된 컴포넌트라면 작업창의 'Nested' 항목 내에 하위 워크플로우를 모델링 한다.

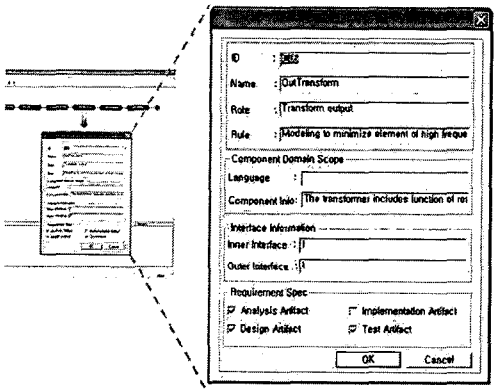


그림 9. 컴포넌트의 스펙에 정보 입력

[그림 9]에서 컴포넌트에 대한 정보는 각 컴포넌트를 더블 클릭 하여 화면에 새로이 생성되는 스펙 창에 입력하며 이는 수정 가능하다. 컴포넌트의 ID, 이름, 역할, 룰, 인터페이스 정보, 각 산출물의 유무를 표기하여 컴포넌트를 재사용하고자 하는 사용자가 해당 컴포넌트에 대한 정보를 확인할 수 있다. 추후 레파지토리에 저장된 컴포넌트를 체계적으로 관리하기 위해서 스펙 창에 입력하는 컴포넌트 ID와 DB에 저장될 ID를 동일하게 해야 한다.

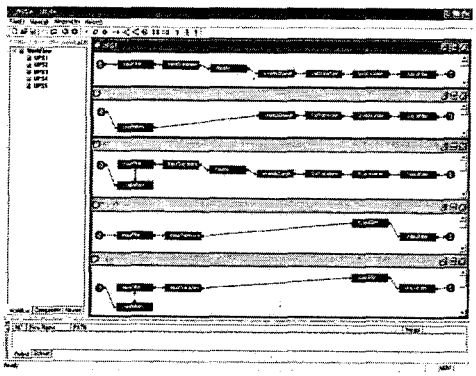


그림 10. UPS를 모델링하기 위한 Flow 구성

모델링을 마쳤다면 시스템 설계 시 고려했던 시나리오와 마찬가지의 경로가 나타나는지 확인 과정을 거친다. [그림 9에서처럼, 메뉴 아이콘 중 '!'를 클릭하여 도구의 하단 'Output' 부분에 표시되는 'Path'를 통해 확인 가능하다. 발생할 수 있는 시나리오와 마찬가지로 UPS의 운영을 이루는 'Flow'들은 [그림 10]에서 볼 수 있는 것과 같다. [그림 11]은 Flow들을 하나로 통합했을 때 나올 수 있는 Output 즉, 발생 가능한 시나리오들을 표시한다.

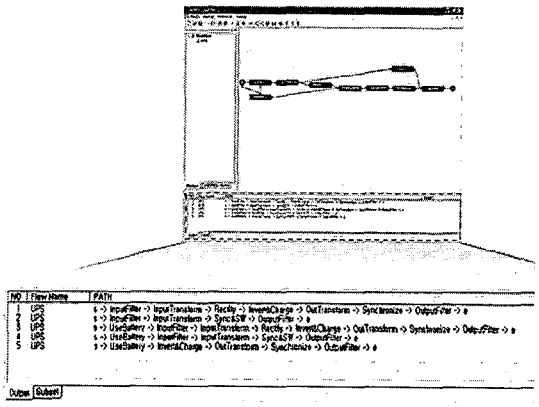
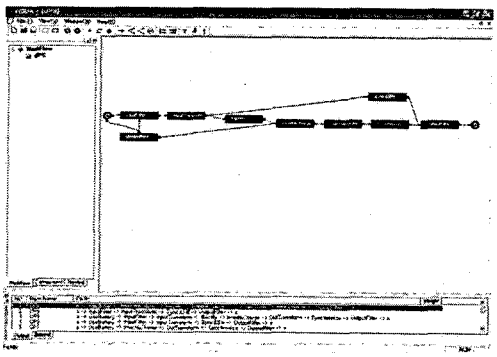
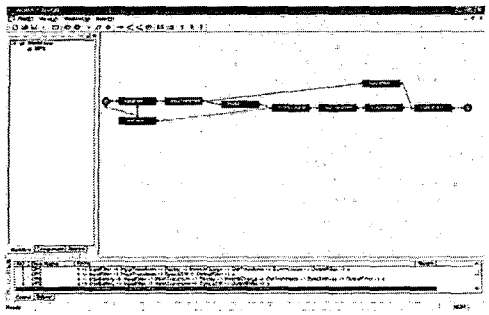


그림 11. 발생 가능한 경로(Output) 표시

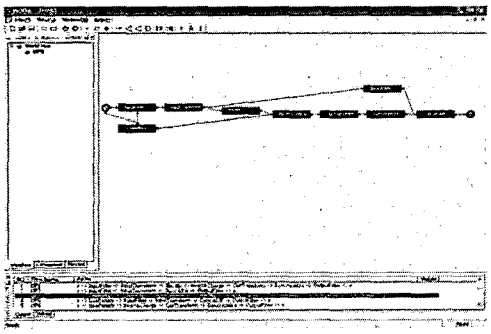
각 경로는 [그림 12]와 같이 이루어지며 여기에서 타 컴포넌트에 비해 진하게 표시된 컴포넌트들이 해당 경로를 나타낸다.



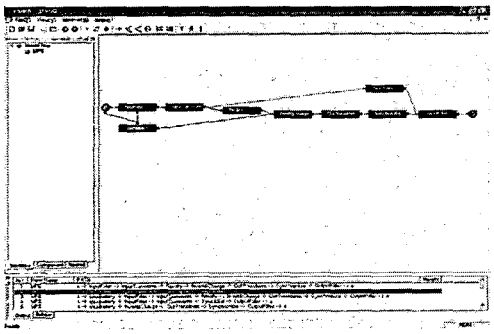
(a) 정상 상태



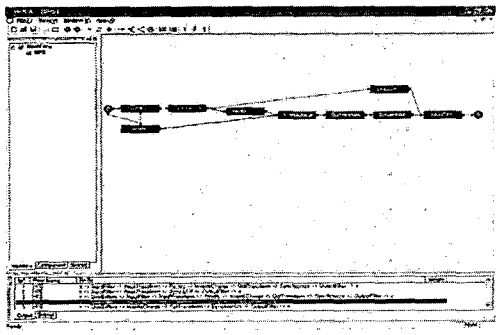
(b) 정전 시



(c) 정상 상태로 복구



(d) 과부하 및 인버터 이상



(e) 정상 상태로 복구된 뒤 인버터 이상

그림 12. UPS의 각 시나리오별 경로

[그림 13]은 UPS 모델링 결과로 나타난 경로의 부분 집합을 보여준다. UPS를 구성하는 Flow들의 부분 집합과 Count 값을 통해 재사용성이 높은 컴포넌트를 추출할 수 있다. 이 논문은 WODA 도구를 통한 모델링 적용 사례로 UPS와 같이 하나의 시스템을 대상으로 했으나, 하부 시스템을 갖는 복잡한 대형 시스템 및 유사성이 있는 시스템들 간 공통의 컴포넌트들을 쉽게 추출할 수 있다. 따라서 이 자동화 도구를 통해 적당한 컴포넌트의 클러스터를 빠르고 안정적으로 찾아 시스템 개발비용과 시간을 줄일 수 있을 것이다.

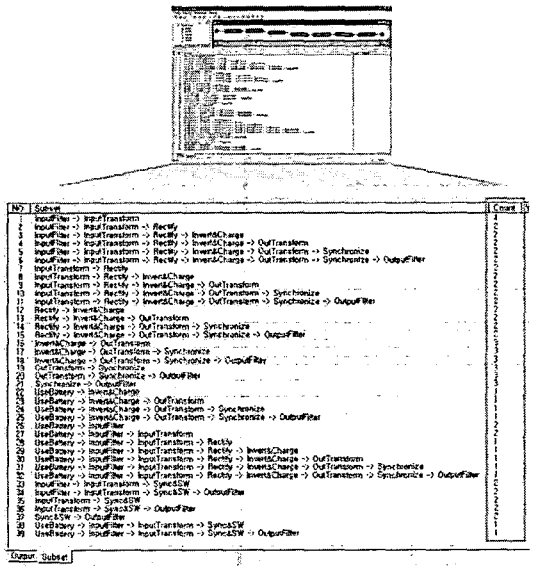


그림 13. UPS 모델링의 부분집합들

현재까지의 WODA 도구에서는 앞서 살펴본 확장된 워크플로우 메커니즘의 구성 요소들 중 병렬개념을 지원하는 요소와 제약사항, 그리고 가중치 값 할당 부분이 미비하여 향후 보완이 필요하다.

Ⅶ. 결 론

현재 유비쿼터스에 대한 연구가 활발히 진행 중이며 소프트웨어공학 면에서는 컴포넌트 기반 개발한 소프트웨어 컴포넌트를 서비스 화 그리고 이를 재사용 및 상호 운용성에 초점을 둬야 할 것이다.

이 논문은 레거시 시스템에 대한 도메인 분석을 위한 방법 제안으로써, 동적 작업 변화의 연속적인 흐름을 나타내는 워크플로우 기반의 도메인 모델링을 통해 공통/비공통 컴포넌트를 추출하기 위한 방법을 제안한다. 도메인 분석 절차를 따라 점진적, 반복적으로 분석함으로써 각각의 사용자에게 적당한 크기의 컴포넌트를 추출하고 마지막 단계에서 UML 기법으로 컴포넌트 내의 객체를 추출할 수 있다. 또한 컴포넌트 테스트 플랜 매트릭스를 통해 사용 빈도수가 많거나 중요한 공통/비공통 컴포넌트를 찾는 방법을 제시한다. 모델링을 자동화하여 컴포넌트를 손쉽게 추출하고 수작업을 통한 모델링의 문제점들을 줄이고자 WODA 도구를 구현하였다. 이를 통해 보다 빠르고 안정적으로 시스템을 모델링 할 수 있을 것이다. 뿐만 아니라 최근 새롭게 대두되고 있는 BPM(Business Process Management)의 핵심기술 요소가 바로 워크플로우이다. 따라서 이 논문의 방법론 적용을 통해 쉽게 BPM에 접근할 수 있을 것이다.

참 고 문 헌

[1] 김윤정, 김영철, "확장된 워크플로우 메커니즘을 통한 공통/비공통 컴포넌트 식별 및 공통 컴포넌트의 클러스터링에 관한 연구", 한국정보처리학회 춘계학술발표대회 논문집, 제11권, 제1호, 2004.

[2] S. Fraser, J. Coplien, and J. White, *Application of Domain Analysis to Object-Oriented System*, ACM Press, pp.46-49, 1995.

[3] K. Belhajjame, "A Flexible workflow model for process-oriented applications," Proceedings of the Second International Conference, IEEE, 2002.

[4] P. Bichler, G. Preuner, and M. Schrefl, "Workflow Transparency," Conference on Advanced Information Systems Engineering, 1997.

[5] 김영철, 최은만, 전병국, 국방 소프트웨어 컴포넌트 *객체 웹 구조 및 플랫폼 기술 연구(최종보고서)*. 국방과학연구소, 2003.

[6] Y. Kim and R. Carlson, "Scenario Based Integration Testing for Object Oriented Software Development," The 8th Asian test symposium(ATS99), pp.283-288, 1999.

[7] Y. Kim, J. Kim, and R. Carlson, "Adaptive Design Based Testing," Proceedings of the ISCA 15th International Conference on Computers and their applications, pp.165-168, 2000.

[8] 김영철, *컴포넌트 기반 체계 상호운용 적합성 평가 및 인증 기술 연구*, 국방과학연구소, 2004.

[9] 김윤정, *워크플로우 메커니즘을 통한 소프트웨어 컴포넌트 식별 방법론에 관한 연구*, 홍익대학교, 2004.

[10] 강교철, *Feature-Oriented Product Line Software Engineering: Principles and Guidelines*, Talyor & Fraincs, 2003.

저 자 소 개

김 영 철(Young-Chul Kim) 정회원

- 2000년 : Illinois Institute of Technology(공학박사)
- 2000년~2001년 : LG 산전 중앙연구소 Embedded system 부장
- 2001년~현재 : 홍익대학교 컴퓨터정보통신 조교수

<관심분야> : 소프트웨어 성숙도 모델, Use Case 방법론 및 도구 개발, CBD, BPM, 사용자 행위 분석 방법론

김 윤 정(Yun-Jeong Kim) 정회원

- 1994년~1997년 : 홍익대학교 컴퓨터정보통신 소프트웨어공학전공(공학학사)
- 2003년~2005년 : 홍익대학교 컴퓨터정보통신 소프트웨어공학전공(공학석사)
- 2005년~현재 : KISTI 연구원

<관심분야> : Workflow modeling, Use Case 방법론 및 도구 개발, CBD, BPM