한국정보과학회

제1권 제1호
Vol. 1 No. 1

2005

한국 소프트웨어공학 학술대회 논문집

Proceedings of 2005 Korean Conference on Software Engineering

- 일시 : 2005년 2월 21일(월)~23일(수)
- 장소 : 무주리조트

주최 : 한국정보과학회, 한국정보처리학회

주관 : 한국정보과학회 소프트웨어공학연구회
한국정보처리학회 소프트웨어공학연구회
한국전자통신연구원

후원 : 한국소프트웨어진흥원
KAIST 소프트웨어프로세스 개선센터
건국대 소프트웨어 연구센터
(주)n3soft

주제 4:어키텍처 및 컴포넌트 I, II (2월 22일)

· 로봇 소프트웨어의 시맨틱스 기반 소프트웨어 컴포넌트 선택 메커니즘----233
이화연, 신현일, 고인영, 최호진 (한국정보통신대학교)
· 워크플로우 지향 도메인 분석(WODA) 도구를 이용한 컴포넌트 추출에 관한 연구----242
김윤정, 김영철(홍익대)
· 독립형 임베디드 RFID리더 시스템을 위한 통신 소프트웨어 설계 및 구현----250
탁성우, 김종덕, 백창훈, 이동건, 이현지, 장수완, 신동우 (부산대)
· 아키텍처 모델의 정제 시 만족되어야 할 행위적 일관성----258
김경래, 전태웅(고려대)
· 아키텍처의 행위명세를 위한 UML 2.0 프로파일----271
신혜선, 전태웅(고려대),최유희(한국전자통신연구원)
· 레고 로봇을 위한 코드 자동 생성 기법----285
이정림, 권기현(경기대)

주제 5: 프로세스 I, II (2월 22일)

· 소프트웨어 제품 라인 공학을 적용한 공정 제어 시스템 개발 : 사례 연구----295
김근배(포스코), 강교철, 김경석, 김상목, 이재준(포항공대),고은만(LG CNS),양강선(포스코)
· 효과적인 프로세스 개선을 위한 Opportunity Tree 프레임 워크 설계----307
이민광, 송기원, 이경환(중앙대)
· CMMI를 기반으로 확장된 SPEM 프로파일을 이용한 프로세스 모델링과 테일러링----316
박승훈, 윤경아, 배두환(한국과학기술원)
· 정보보호제품 결함관리도구 개발 지원 프로세스----328
김행곤(대구가톨릭대), 양해술(호서대)
· 웹 애플리케이션을 위한 UI 중심의 경량 개발 프로세스----333
이기열, 박상현, 이춘우, 이욱진, 정우성, 박영주, 우치수(서울대), 이병정(서울시립대), 김희천(한국방통대)
· 대용량 생물정보 문헌으로부터 단백질 상호작용 분석을 위한 바이오 데이터 마이닝 시스템----344
여은주, 이현철, 조완섭(충북대)

주제 6: 소프트웨어 설계 (2월 23일)

· 메트릭 기반의 디자인 패턴 결정 방법----359
서광익, 최은만(동국대)
· P2P 프로토콜 패턴분석 및 서비스 차단 알고리즘----369
황선명, 김상영, 이재영(대전대)
· 자바가상머신을 지원하는 컴포넌트 의존 인터페이스 알고리즘----377
최상균(김포대), 송영재(경희대)

주제 7: 소프트웨어 응용 (2월 23일)

· 홈 서버 미들웨어를 위한 규칙 관리자와 디바이스 관리자 설계----387
황도현, 이현정, 정기원(숭실대)
· 상황인식형 자가구성 시스템의 설계----397
오제환, 이승희, 이은석(성균관대)
· 객체지향 데이터베이스를 이용한 SBML 문서 관리기----407
임정곤, 김대경, 정태성, 조완섭(충북대)

워크플로우 지향 도메인 분석(WODA) 도구를 이용한 컴포넌트 추출에 관한 연구

김윤정, 김영철

홍익대학교 일반대학원 전자전산공학과
충남 연기군 조치원읍 신안리
{simile, bob}@won.hongik.ac.kr

요약: 본 연구에서는 레거시 시스템에 대한 기존 도메인 분석의 문제점을 해결하기 위하여 동적 모델링을 기반으로 하는 확장된 워크플로우 메커니즘을 이용한 도메인 분석 방법론을 제안한다. 이 방법론을 워크플로우 지향 도메인 분석(WODA)이라 명명한다. 본 논문에서 제안하는 절차를 통해 공통/비공통 컴포넌트들을 식별할 수 있고 컴포넌트들의 클러스터와 중요한 컴포넌트들을 추출할 수 있다. 이를 통해 새로운 시스템을 개발 시 효율적으로 재사용하려 한다. 특정한 시스템에서 발생할 수 있는 모든 시나리오들을 식별한 후, 컴포넌트 가중치 측정 매트릭스를 이용하여 재사용성이 가장 높은 컴포넌트와 컴포넌트 시나리오를 결정한다. 컴포넌트 가중치 측정 매트릭스를 통하여 재사용 가능한 컴포넌트들의 중요성과 빈도수 인식 및 시스템에 있는 유용한 컴포넌트 시나리오 식별을 할 수 있다. 또한 재사용 가능한 컴포넌트들을 자동으로 추출하기 위해 구현한 자동화 모델링 도구인 WODA를 소개한다.

핵심어: 도메인 분석, 워크플로우, WODA, 컴포넌트 추출, 컴포넌트 가중치 측정 매트릭스

1. 서론

현재 소프트웨어 기술의 근간은 컴포넌트 기반 개발일 것이다. 양질의 소프트웨어를 제공하기 위해 업계와 학계에서 활발하게 연구가 진행 중인 것이 컴포넌트 기반 소프트웨어 개발 방법론이다.

컴포넌트는 재사용, 재배치가 가능하다는 장점을 갖고 있으며 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 최근 IT 업계의 화두인 BPM(Business Process Management)의 실제 근간이고 시스템적으로 구현하는 핵심 수단[9]이기도 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 식별한 후, 컴포넌트들마다 할당된 가중치 값을 통해 재사용성이 높은 컴포넌트를 결정하고자 한다. 이는 컴포넌트 가중치 측정 매트릭스를 이용하여 결정 가능하다[1]. 컴포넌트 가중치 측정 매트릭스는 컴포넌트 대스트 플랜 매트릭스[6][7]를 이용하였다.

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 모델링 도구로 구현한 WODA(Workflow Oriented Domain Analysis) 도구를 통해 컴포넌트의 추출을 확인한다. 마지막으로 6장에서 결론과 향후 연구방향에 대하여 기술한다.

2. 확장된 워크플로우 메커니즘

본 논문은 현존하는 시스템인 레거시 시스템에 대한 도메인 분석을 제안하고자 한다. 레거시 시스템에 대한 연구와 분석을 통해 재사용할 수 있는 자산들(컴포넌트)을 추출할 수 있다. 각각의 재사용 가능한 컴포넌트를 품질 평가하여 양질의 컴포넌트로 고품질의 새로운 시스템을 생산하는데 목적을 두고 있다.

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, Collaboration Diagrams, Sequence Diagrams, Use-Case Diagrams, Activity Diagrams, Colored Petri-Net 등은 객체나 함수 레벨과 같이 너무 작은 규모의 레벨에서 시스템을 모델링 하는 단점이 있다. 객체를 추출하고 후에 적절한 단위의 컴포넌트로 객체들을 묶는 방식인 상향식의 분석일 경우 모델의 복잡화를 초래할 수 있고, feature 개념을 이용하는 경우에는 의미 없는 feature가 도출될 수 있다. 대형 시스템을 상향식으로 분석하는 경우는 feature를 도출하는데 긴 시간이 소요되는 단점이 있고, 새로운 제품을 개발하기 위한 경우 공통성과 가변성 분석 및 식별의 객관적인 기준이 필요하다. 이에 하향식으로 분석하였고 수준이나 규모를 다르게 정의할 수 있는 워크플로우 개념을 도입하여 확장하였다.

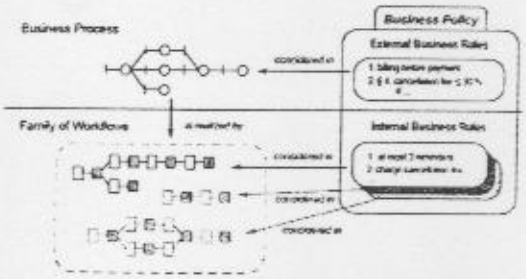


그림 1 2-schema 구조의 기본적 개념

워크플로우란 (비즈니스) 프로세스 모델링의 일종으로 시스템 또는 애플리케이션에 대해 실질적인 목적 수행을 위한 동적 작업 변화의 연속적인 흐름을 나타내는 방법이다[3]. 그림 1과 같이 비즈니스 프로세스는 워크플로우들로 이루어진다. 즉 워크플로우는 어떤 조직에 있어서 비즈니스 프로세스의 실현을 이루는 요소가 될 수 있으리라 본다[4].

워크플로우 모델링의 장점은 비즈니스 프로세스의 업무 구조를 모델링 하는 데에도 적합하다는 것이다[3]. 본 논문에서는 병렬 개념을 추가한 확장된 워크플로우 메커니즘을 통해 도메인 분석을 제시하고자 한다. 이를 통해 IT 분야, 비즈니스 모델링뿐만 아니라 병렬 시스템, 실시간 시스템도 모델링 할 수 있으리라 기대한다. 그림 2는 확장된 워크플로우 메커니즘에 필요한 요소들을 도식화한 것이다.

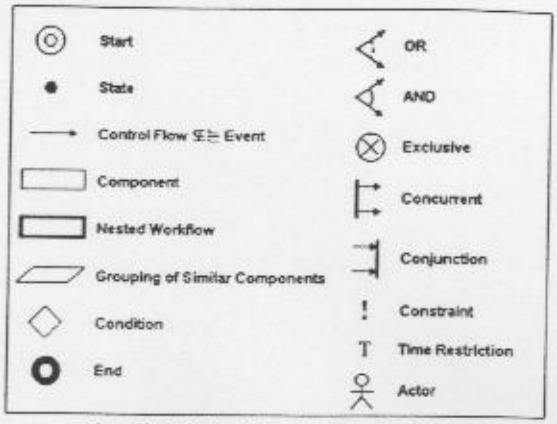


그림 2 확장된 워크플로우 메커니즘의 범례

(1) 시작(Start) : 전체 Activity들의 개시
(2) 상태(State)
(3) 이벤트(Event) : Activity를 수행하는 동안 발생하는 것
(4) 컴포넌트(Component) : 워크플로우에서 임무를 수행하는 Activity
(5) 내포된 워크플로우(Nested Workflow) : 내포 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 논리연산과 Concurrent, 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 임무 담당자

소프트웨어 시스템을 3장에서 제안한 방법론에 따라 하향식 방식으로 분석하면, 상위 레벨의 워크플로우가 반복적, 점증적 분석을 통해 하위 워크플로우로 이루어져 있음을 알 수 있다. 내포 개념으로 하위 워크플로우를 가질 수 있는 것을 'Nested Workflow'로 정의하였다. 또한 입력 값과 출력 값이 다르지만 유사한 기능을 하는 프로세스들을 그룹화 할 수 있다.

그림 2에서 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.

3. 워크플로우 메커니즘을 이용한 도메인 분석 방법론

이 장에서는 앞 장에서 언급한 것과 같이 시스템에 대해 실질적인 목적 수행을 위한 동적 작업 변화의 연속적인 흐름인 워크플로우 메커니즘을 이용한 도메인 분석 방법을 언급하고자 한다. 컴포넌트를 추출하기 위한 표준화된 절차를 제시함으로써 사용자가 원하는 크기의 컴포넌트를 찾을 수 있다. 이 방법은 확장된 워크플로우 메커니즘을 이용한 도메인 분석 방법론(Workflow Oriented Domain Analysis)으로 [5]를 확장한 것이며 아래의 절차를 따른다. 개발절차의 산출물은 [10]에 기술하였다.

1단계 : 워크플로우 메커니즘을 이용한 도메인 정의
적용할 특정한 도메인에 대해 행위자, 프로세스, 상태 등을 이용한 상위 레벨의 워크플로우 모델링을 통해 도메인 정의.

2단계 : 워크플로우 메커니즘을 이용한 도메인 분석
적용할 특정한 도메인에 대해 행위자, 프로세스, 상태 등을 적용한 워크플로우 모델링을 통해 도메인 분석.

3단계 : 워크플로우 메커니즘을 이용한 도메인 구조
전체 시스템에 대한 프로세스 모델을 제시.

4단계 : 워크플로우 메커니즘을 이용한 도메인 설계
전체 시스템에 대한 하향식 및 계층적 메커니즘을 통해 필요한 워크플로우 모델링을 설계.

5단계 : 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).

6단계 : 컴포넌트 설계

7단계 : 컴포넌트 구현

8단계 : 컴포넌트 내의 객체 추출을 위해 객체지향 방법론 적용

9단계 : 컴포넌트 테스트

이 절차를 적용하면, 동적 행위 흐름을 모델링한 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, 실행 코드를 얻어낼 수 있다. 마지막 단계에서 컴포넌트 테스트를 수행한다. 그림 3은 제안한 방법론의 절차를 시각화한 것이다.

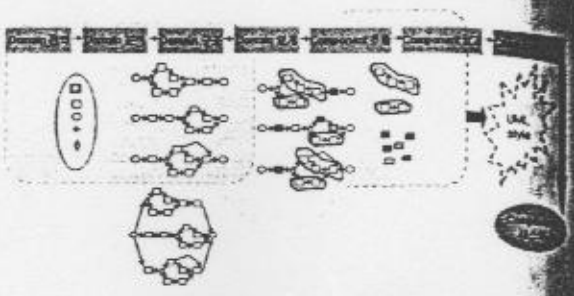


그림 3 워크플로우 메커니즘을 이용한 도메인 분석 방법론 절차

그림 4는 우리가 제안한 컴포넌트 명세 확장과 워크플로우의 구조적 모델링을 나타낸다[8][10]. 하나의 워크플로우는 하나 이상의 컴포넌트로 구성되며 각 컴포넌트는 개발 과정에 따른 여러 버전 별로 그림에서와 같은 부분들로 이루어진다.

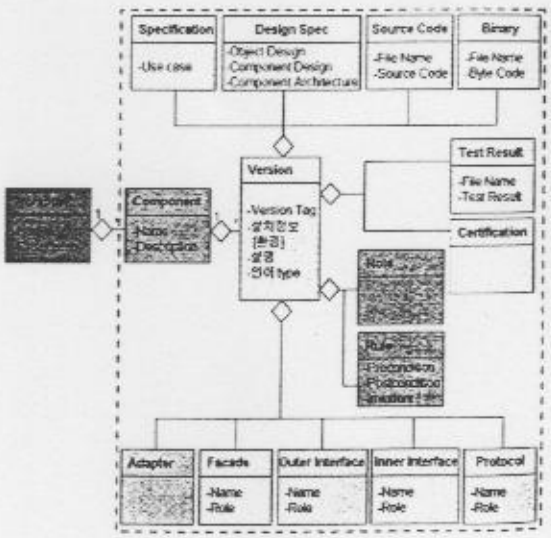


그림 4 구조적인 워크플로우 모델링

4. 확장된 워크플로우 메커니즘을 이용한 도메

인 분석 방법론의 적용 사례

이 장은 제안한 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. UPS(Uninterruptible Power Supply)를 적용 예로 하여 하나의 시스템에서 재사용성이 높은 컴포넌트를 추출하는 것을 제시한다. UPS는 입력 전원의 전압 변동과 정전 및 기타 장애의 발생에 관계없이 항상 부하에 양질의 전력을 공급하는 장치이고 모델링한 결과는 다음 그림과 같다.

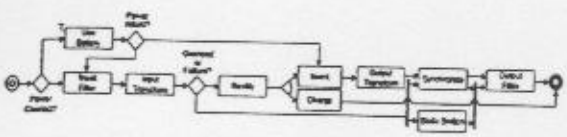
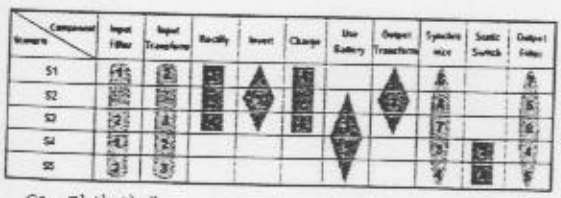


그림 5 UPS 워크플로우 모델링

UPS의 동적인 행위 분석을 위해서 발생 가능한 시나리오들을 살펴보면 그림 6에서와 같이 다섯 가지이다. 각 시나리오별로 컴포넌트에 기입된 숫자가 해당 시나리오의 경로 순서를 나타내며, 타원, 모서리가 둥근 직사각형, 마름모, 직사각형은 컴포넌트가 다섯 개의 시나리오에 동일한 횟수로 사용된 것을 나타낸다. 한 시나리오에 같은 숫자가 표시된 컴포넌트는 병렬 개념 중 AND와 Concurrent가 사용되는 것을 의미한다. 그림 5의 모델링 결과로 찾은 각각의 컴포넌트가 얼마나 사용되고 있는지 그림 6을 통해 알 수 있다.



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

그림 6 시나리오별 컴포넌트 사용

	Input Filter	Input Transform	Rectify	Invert	Charge	Bat Battery	Output Transform	Synchronize	Static Switch	Output Filter
Weighted value	1	1	1	1	1	1	1	1	1	1
Reusability weight	4	4	3	3	2	3	3	5	2	6

	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	[illegible]	20	23

그림 7 재사용성에 따른 중요 컴포넌트 측정
(#숫자 : Reusability weight)

그림 7은 UPS의 각 컴포넌트마다 가중치를 1로 가정하여 설정한 후, 재사용성에 따른 값을 계산하여 이 중 재사용성이 높은 컴포넌트들이 포함된 시나리오를 찾는 것을 보여준다(가중치 값 설정은 향후 연구가 필요한 부분이다). 하단 그림의 좌측 열은 시나리오에 공통적으로 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 6에 명기된 전체 5개의 시나리오 중 S3(정전 후 정상 상태로 복구)은 나머지 네 개의 시나리오에 비해 재사용성 가중치의 합이 커 재사용성이 높은 컴포넌트들을 사용한 시나리오로 볼 수 있다.

제안된 방법을 통해 인간의 경험이나 선호도에 따른 컴포넌트의 재사용이 아니라 시스템에서 발생할 수 있는 시나리오들을 통해 컴포넌트의 효율적인 재사용 및 중요도를 측정할 수 있으리라 기대한다. UPS 모델링의 경우에는 컴포넌트 가중치 측정 매트릭스 중 재사용성 항목에만 초점을 맞추었다.

5. 구현한 WODA 도구를 통해 컴포넌트를 추출

이 장에서는 시스템을 분석하여 모델링 한 결과를 도구를 통해 적용하였다. 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 모델링 하고자 한다.

WODA를 실행하면 그림 8과 같은 화면을 구성한다. 앞에서 보았던 확장된 워크플로우 메커니즘의 범례(그림 2)의 요소들이 메뉴 항목 아래에 아이콘으로 구성되어 있다. 시스템을 모델링하기 위해 새로운 프로젝트를 생성한 후 그림 9와 같이 WODA 메인의 'Workflow' 하에 시스템을 위해 필요한 'Flow'들을 추가하고, 그림 10에서와 같이 컴포넌트들을 생성하면서 이 컴포넌트들을

드래그 하여 시스템을 모델링 한다. 상위 레벨의 시스템 모델링을 통해 컴포넌트가 하위 워크플로우로 구성되는 내포된 컴포넌트라면, 작업창의 'Nested' 항목 내에 하위 워크플로우를 모델링 하면 되고 이 내포된 컴포넌트도 개개의 컴포넌트와 마찬가지로 드래그 하여 모델링에 적용하면 된다.

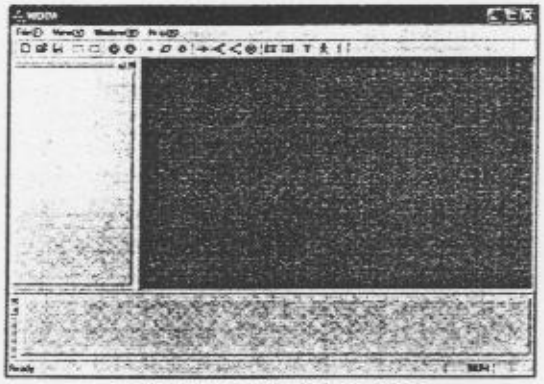


그림 8 WODA의 실행 초기화면

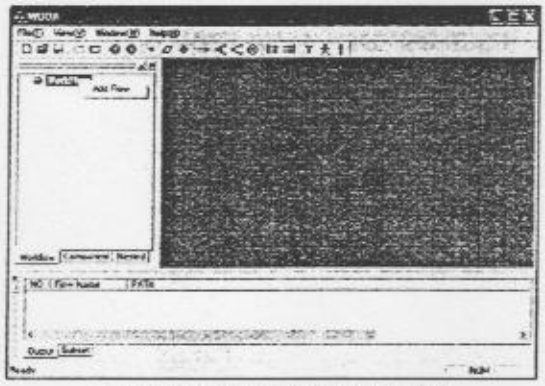


그림 9 시스템을 구성하는 Flow 생성

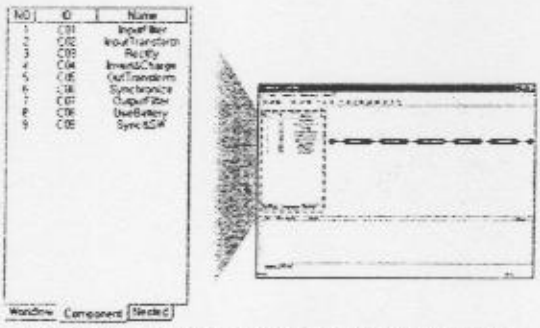


그림 10 Flow를 구성하기 위해 필요한 컴포넌트 목록

컴포넌트에 대한 정보는 각 컴포넌트를 더블 클릭하여 화면에 새로이 생성되는 명세 창에 입력하며 이는 수정 가능하다. 컴포넌트의 ID, 이름, 역할, 룰, 인터페이스 정보, 각 산출물의 유무 등을 표기하여 컴포넌트를 이용하고자 하는 사용자가 해당 컴포넌트에 대한 정보를 확인할 수 있다. 추후 레파지토리에 저장된 컴포넌트를 체계적으로 관리하기 위해서 명세 창에 입력하는 컴포넌트 ID와 DB에 저장될 ID를 동일하게 해야 할 것이다.

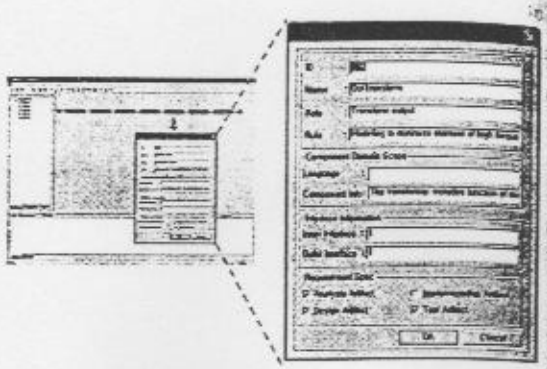


그림 11 컴포넌트의 명세에 정보 입력

모델링 후, 시스템 설계 시 고려했던 시나리오와 마찬가지의 경로가 나타나는지 확인 과정을 거친다. 이것은 메뉴 아이콘 중 'T'를 클릭하여 도구의 하단 'Output' 부분에 표시되는 'Path'를 통해 확인 가능하다. 발생할 수 있는 시나리오와 마찬가지로 UPS의 운영을 이루는 'Flow'들은 그림 12에서 볼 수 있는 것과 같다. 그림 13은 Flow들을 하나로 통합했을 때 나올 수 있는 Output 즉, 발생 가능한 시나리오들을 표시한다.

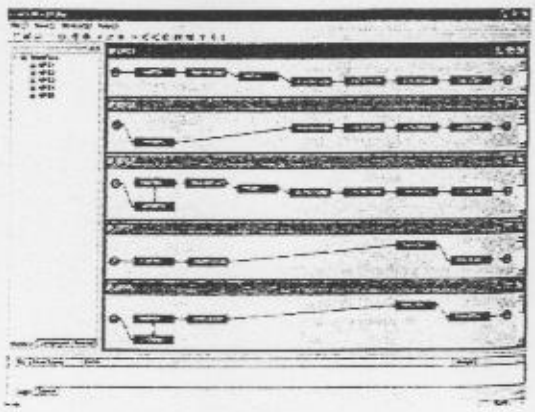


그림 12 UPS를 모델링하기 위한 Flow 구성

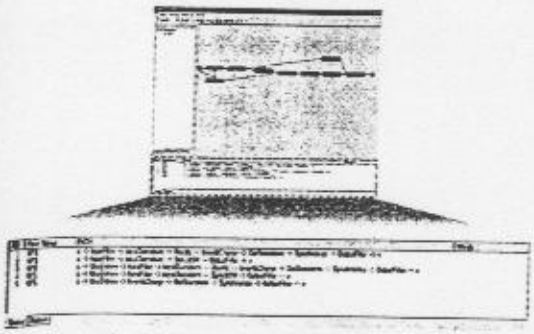
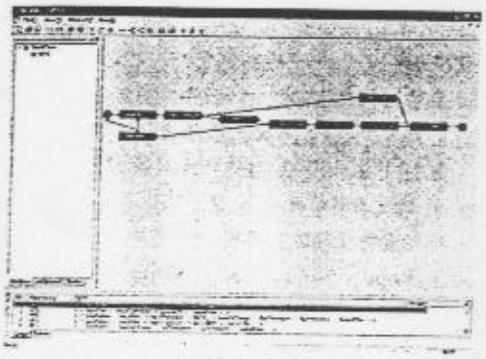
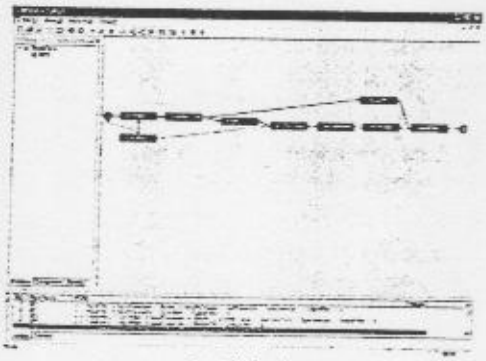


그림 13 발생 가능한 경로(Output) 표시

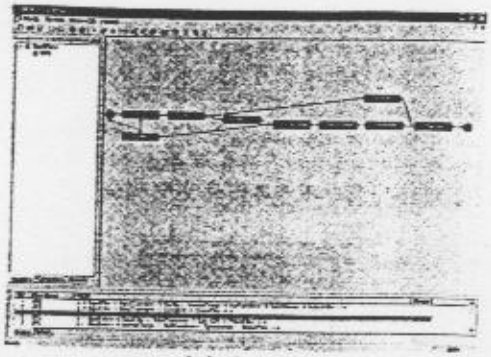
각 경로는 그림 14와 같이 이루어지며 하단의 해당 경로를 클릭하면 그 경로에 사용되는 컴포넌트가 타 컴포넌트에 비해 진하게 표시된다.



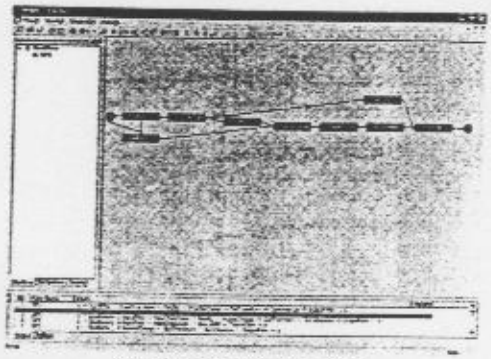
(a) 정상 상태



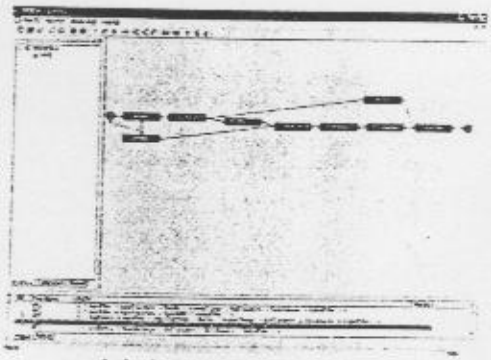
(b) 정전 시



(c) 정상 상태로 복구



(d) 과부하 및 인버터 이상



(e) 정상 상태로 복구된 뒤 인버터 이상

그림 14 UPS의 각 시나리오별 경로

그림 15는 UPS 모델링 결과로 나타난 경로의 부분 집합(Subset)을 보여준다. UPS를 구성하는 Flow들의 부분 집합과 Count 값을 통해 재사용성이 높은 컴포넌트들을 추출할 수 있다. WODA 도구로 UPS를 모델링 했을 때 InputFilter와 InputTransform 컴포넌트 Subset이 가장 높은 Count = 4를 나타냄으로써 재사용성이 높은 컴포넌트 클러스터임을 확인할 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 시스템들간 공통의 컴포넌트들을 도구를 이용함으로써 객관적으로 추출할 수 있다. 따라서 이 자동화 도구를 통해 적당한 컴포넌트의 클러스터를 빠르고 안정적으로 찾아 시스템 개발 시 시간과 비용을 줄일 수 있을 것이다.

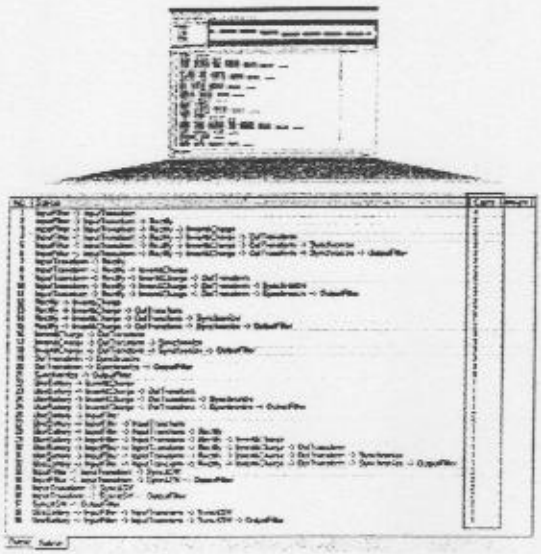


그림 15 UPS 모델링의 부분집합들

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

6. 결 론

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, UML 기법 적용으로 컴포넌트 내의 객체를 추출할 수 있으며 더 나아가 실행 코드를 얻을 수 있을 것이다. 또한 모델링을 자동화하여 컴포넌트를 손쉽게 추출하고 수작업을 통한 모델링의 문제점들을 줄이고자 WODA 도구를 구현하였다. 이 도구를 통해 시스템에서 재사용되는 컴포넌트에 대한 개발자의 주관적 판단이 아닌 객관적인 판단을 할 수 있으며, 보다 빠르고 안정적으로 시스템을 모델링 할 수 있을 것이다. 본 논문에서 제안한 확장된 워크플로우 메커니즘을 통해 IT 분야, 비즈니스 모델링뿐만 아니라 병렬 시스템, 실시간 시스템도 모델링 할 수 있으리라 기대한다. 뿐만 아니라 최근 새롭게 대두되고 있는 BPM의 핵심기술 요소가 바로 워크플로우이다. 따라서 이 논문의 방법론 적용을 통해 업계 BPM에 접근할 수 있을 것이다.

향후 연구 과제로 워크플로우 메커니즘을 이용한 도메인 분석을 통하여 추출된 공통/비공통 컴포넌트를 검증하기 위한 인증 연구 및 WODA의 병렬개념을 지원하는 요소, 제약사항, 가중치 값을 통한 재사용성 값 계산에 대한 추가 보완 그리고 방법론의 기본이 되는 컴포넌트의 정규화(Normalization)를 들 수 있다. 현재 본 논문에서 제안한 방법론을 임베디드 시스템 개발에 적용할 방법과 WODA 도구에서 컴포넌트의 자동 실행 코드 생성에 관한 연구를 진행 중이다.

참고문헌

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

[2] S. Fraser, J. Coplien, 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, M. Schrefl, "Workflow Transparency", Conference on Advanced Information Systems Engineering, 1997.

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

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

[7] Y. Kim, J. Kim, 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] 문희철, 오래된 워크플로우? 새로운 BPM! -2, 마이크로소프트웨어, 2004년 6월호.

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