

확장된 워크플로우 메커니즘을 통한 공통/비공통 컴포넌트 식별 및 공통 컴포넌트의 클러스터링에 관한 연구

김윤정*, 김영철

*홍익대학교 일반대학원 전자전산공학과
e-mail:simile@won.hongik.ac.kr

A Study on identifying Common/Uncommon Components and clustering Common Components through Extended Workflow Mechanism

Yun-Jeong Kim*, R. Young-Chul Kim

*Dept. of Electronics and Computer Science Engineering,
Hong-Ik University

요 약

레거시 시스템을 위한 기존의 도메인 분석의 문제점을 해결하기 위하여 동적인 모델링인 확장된 워크플로우 메커니즘을 기반으로 하는 도메인 분석 방법을 제안하고자 한다. 이 도메인 분석을 통해 공통/비공통의 프로세스 컴포넌트 식별 및 공통 프로세스 컴포넌트들의 클러스터를 추출하고 마지막 단계에서 UML 기법으로 컴포넌트 내의 객체를 추출할 수 있다. 또한 제안한 컴포넌트 가중치 측정 매트릭스에 적용해 사용 빈도수가 많거나 중요한 컴포넌트 및 컴포넌트 클러스터를 찾는 방법을 제시하고자 한다.

1. 서론

정보기술이 발전하면서 소프트웨어 산업이 차지하는 비중이 점점 커지고 있다. 이에 활발하게 연구가 진행 중인 것이 컴포넌트 기반 소프트웨어 개발(Component Based Software Development : CBSD) 방법론이다. CBSD는 기존의 시스템 개발 방식과는 달리 기 개발된 컴포넌트들을 조합하여 새로운 시스템을 개발하는 방식이다. 재사용, 재배포가 가능하고 변화에 대하여 관리가 가능한 컴포넌트의 장점을 통하여 개발 시간을 단축시키고 개발비용을 절감시켜 소프트웨어 생산성을 향상시킨다.

본 논문은 시스템들의 개발과 유지관리 효율성을 높이기 위한 재사용 그리고 시스템들과 애플리케이션들에서의 정적 구조와 동적 행위 추출에 목적을 둔 도메인 분석을 이용하여 컴포넌트를 추출하고자 한다. 도메인이란 시스템을 내의 공통 기능 영역이나 그룹을 말한다. 소프트웨어 시스템의 도메인 분석을 통해

공통/비공통의 컴포넌트를 추출하고, 그룹화된 컴포넌트와 중요도에 따른 컴포넌트를 통해 시스템 개발 시 효과적으로 재사용할 수 있는 컴포넌트를 식별하는 방법에 대해 기술하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 관련 연구로서 도메인 분석에 대해 언급하고, 기존의 도메인 분석 방법에 대해 기술한다. 3장에서는 확장된 워크플로우 메커니즘, 4장에서는 워크플로우 메커니즘을 이용한 도메인 분석에 관하여 소개하고, 5장에서는 이를 적용한 사례를 언급한다. 마지막으로 결론과 향후 연구방향에 대하여 기술한다.

2. 관련 연구

2.1. 도메인 분석

도메인 분석과 도메인 공학이라는 용어는 서로 일치하지는 않지만 호환성 있게 사용된다. 도메인 분석이 도메인 공학보다 앞서 나왔지만, 도메인 공

학이 좀 더 포괄적인 용어다[2].

도메인 공학은 재사용을 지원할 시스템들을 위해 '도메인 정의 → 도메인 분석 → 도메인 구조 → 도메인 설계 → 컴포넌트 추출 → 컴포넌트 설계 → 컴포넌트 구현'의 단계를 갖는다[1]. 이 중 도메인 분석의 산출물이 도메인 모델이며, 도메인 모델은 기존 시스템의 유지계획과 유사한 시스템들에 대해서 향후에 개발을 위한 재사용을 발견하고 추출한다.

2.2. 기존의 도메인 분석 방법론

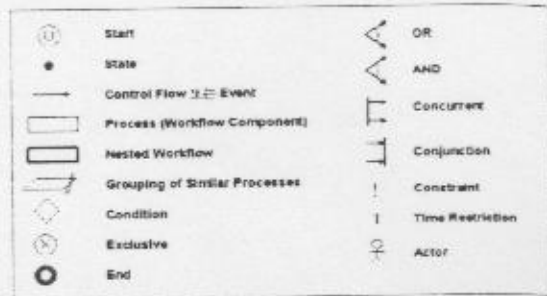
FODA(Feature Oriented Domain Analysis)는 feature(시스템에 대한 기능 및 특성) 중심의 도메인 분석방법을 말한다. 1990년 Software Engineering Institute를 통하여 소개된 이후, 산업계와 학계에서도 도메인 분석과 프로덕트 라인을 분석하는 방법으로 사용되어 왔다. FODA를 기반으로 한 프로덕트 라인 방법론인 FORM(Feature-Oriented Reuse Method)은 소프트웨어 아키텍처 설계와 객체 지향 컴포넌트 개발을 지원한다. FORM의 핵심은 프로덕트 라인을 feature 중심으로 분석하고 이러한 feature를 이용하여 재사용 가능하고 변화에 적응시킬 수 있는 프로덕트 라인 자산을 개발하는데 있다[3].

이 외에도 ODM(Organization Domain Modeling), FeatureRSEB(Feature Reuse-Driven Software Engineering Business), FODacom 등이 있지만 이들의 문제점을 볼 때 상향식 방식으로써 공통성과 가변성에 대해 언급을 했을 뿐 비공통 컴포넌트에 대해서는 언급이 없다. 그래서 우리는 하향식의 공통/비공통 컴포넌트 식별 및 컴포넌트의 클러스터를 찾는 방법에 대해 언급하고자 한다.

3. 확장된 워크플로우 메커니즘

본 논문은 레거시 시스템에 대한 도메인 분석을 제안하고자 한다. 레거시 시스템과 새로운 시스템의 도메인 분석에는 차이점이 있다[4]. 레거시 시스템에서 도메인 분석을 위해 보편화되어 있는 모델링 기술 방법들은 너무 작은 규모의 레벨에서 시스템을 모델링 하는 단점이 있다. 새로운 시스템에서는 도메인 분석 단계 입력이 요구사항이기 때문에 요구사항 분석 정도의 수준을 도메인 활동으로 볼 수 있다. 수준이나 규모를 다르게 정의할 수 있는 개념이 필요한데 이것이 바로 워크플로우이다. 워크플로우란 (비즈니스) 프로세스 모델링의 일종으로 시스템

또는 애플리케이션에 대해 실질적인 목적 수행을 위한 동적 작업 변화의 연속적인 흐름을 나타내는 방법이다[6]. 워크플로우 모델링의 장점은 비즈니스 프로세스의 협업 구조를 모델링 하는데 적합하다는 것이다. 따라서, 우리는 확장된 워크플로우 메커니즘을 통해 도메인 분석을 제시하고자 한다. (그림 1)은 확장된 워크플로우 메커니즘에 필요한 요소들을 도식화한 것이다.



(그림 1) 확장된 워크플로우 메커니즘의 범례

4. 워크플로우 메커니즘을 이용한 도메인 분석

이 장은 컴포넌트를 추출하기 위한 표준화된 절차를 제시함으로써 언제, 누구든지 간에 원하는 크기의 컴포넌트를 찾을 수 있는 방법을 제시하고자 한다. 이 방법은 워크플로우 메커니즘을 이용한 도메인 분석으로 아래의 절차에 의해 수행된다[7].

- 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 이용해 도메인 분석. 적용할 특정한 도메인에 대해 행위자, 프로세스 컴포넌트, 상태 등을 적용한 워크플로우 모델링을 통해 도메인 분석.
- 3단계 : 워크플로우 메커니즘을 이용해 도메인 구조. 전체 시스템에 대한 프로세스 모델을 제시.
- 4단계 : 워크플로우 메커니즘을 이용해 도메인 설계. 전체 시스템에 대한 하향식(top-down) 및 제층적 메커니즘을 통해 필요한 워크플로우 모델링을 설계.
- 5단계 : 프로세스 컴포넌트 추출.
 - 5-1단계 : 4단계를 통해, 공통/비공통의 프로세스들을 추출.
 - 5-2단계 : 각각의 프로세스 컴포넌트에 대해 하위 워크플로우(또는 네포된 워크플로

우)를 찾는다.

1단계부터 5단계 반복수행(원하는 크기의 컴포넌트를 찾을 때까지).

6단계 : 컴포넌트 설계.

7단계 : 컴포넌트 구현.

8단계 : 컴포넌트 내의 객체 추출을 위해 객체지향 방법론 적용.

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. 마지막으로 기존의 UML(Unified Modeling Language) 분석 단계를 수행하여 객체를 추출할 수 있다. 이 과정에서 공통/비공통의 컴포넌트(또는 프로세스)를 추출하고 이를 그룹화할 수 있는데, 이는 컴포넌트 경로를 추정함으로써 가능하다.

컴포넌트 경로 추출		컴포넌트 가중치(w)
길이 (Length)	최단 경로 (simple path) - 최소의 스텝	$w = 1$
	최장 경로 (hardest path) - 최대의 스텝	$w = 1$
중요성 (Criticality)	중요 컴포넌트	$w \geq 1$
	중요 컴포넌트	$w \geq 0$
재사용성 (Reusability)	공통 컴포넌트	$w > 1$
	비공통 컴포넌트	$w \geq 0 \text{ and } \neq 1$
	대부분 재사용되는 컴포넌트 클러스터	$w = 1$

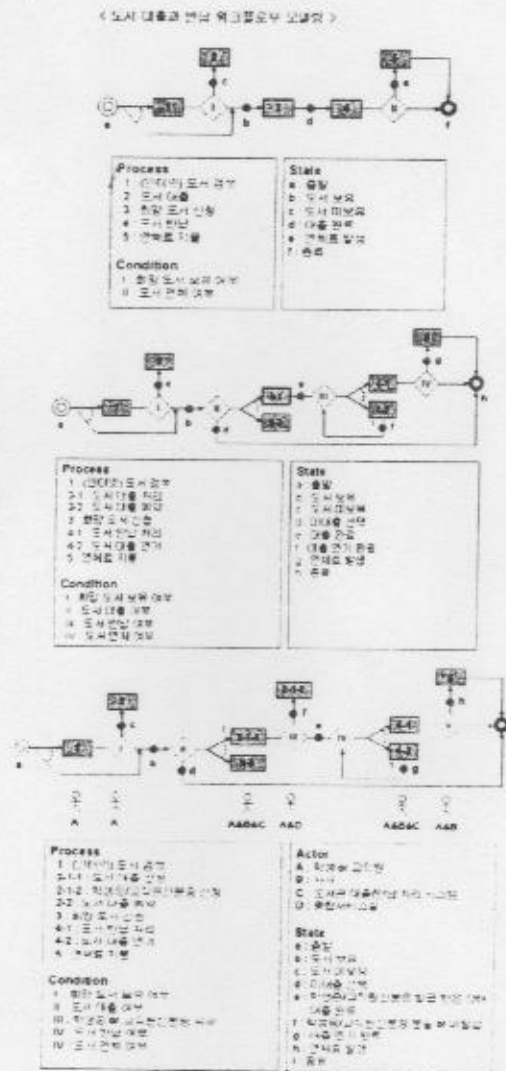
<표 1> 컴포넌트 가중치 추정 매트릭스

컴포넌트 경로의 길이, 중요성, 재사용성별로 컴포넌트 가중치를 두어 가중치가 큰 컴포넌트를 채택하여 시스템을 개발할 때 효율성을 높일 수 있다. <표 1>이 컴포넌트 가중치 추정 매트릭스로서 컴포넌트 테스트 플랜 매트릭스[8][9]를 이용하였다.

5. 확장된 워크플로우 메커니즘 적용 사례

앞 장에서 언급한 방법론을 이용하여 상위 레벨 워크플로우 기반의 컴포넌트 모델링을 통해 상위 레벨의 프로세스 컴포넌트를 정의하고 추출한 후, 하위 레벨의 모델링을 통해 점진적, 반복적으로 사용자(개발자, 설계사, 시험자 등)들에 맞는 크기의 컴포넌트를 추출한다. [10]에서는 학생 학위 허가 시스템에서 단지 컴포넌트의 식별에만 제한하였지만, 이 논문에서는 한 시스템의 동적 시나리오들에서의 공통 컴포넌트의 클러스터링 및 식별에 대한 적용 예로서 대학 도서관의 업무 중 대출과 만남이라는 제

한된 부분을 모델링에 적용하였다. (그림 2)에서 상위 레벨 워크플로우 모델링은 5개의 프로세스 컴포넌트(도서 검색, 도서 대출, 희망 도서 신청, 도서 반납, 연체료 지불)와 2개의 조건 결정(희망 도서 보유 여부, 도서 연체 여부)으로 구성된다. 이 중 프로세스 컴포넌트 2(도서 대출)와 4(도서 반납)를 분석하면 하위 워크플로우들로 구성되어 있음을 알 수 있다.



(그림 2) 도서관 대출&반납 워크플로우 모델링

도서관 대출&반납 워크플로우 모델링을 통해 찾은 프로세스 컴포넌트들을 <표 1>의 컴포넌트 가중치 추정 매트릭스를 이용해 공통/비공통 컴포넌트 및 클러스터에 대한 중요도 및 재사용성을 체크할 수 있다. <표 2>는 시나리오별로 프로세스 컴포넌트들의 경로를 작성하고 그 결과로 공통의 컴포넌트 클러스터를 찾을 수 있음을 보여준다.

Process Component	C1										C2		C3		C4		C5
	S1	S2	S3	S4	S5	S6	S7	S8	C2-1		C2-2	C3	C4-1	C4-2	C5		
									C2-1-1	C2-1-2							
S1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
S2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
S3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
S4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
S5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
S6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
S7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
S8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

- S1: 인터페이스 도식 입력 후 도서를 등록, 저장 내역 도서를 연립하는 경우
- S2: S1의 시나리오에 연속요구 변경되는 경우
- S3: S1의 시나리오에 연속 연계를 한 경우
- S4: (등록 연거 후 연계를 변경한 경우)
- S5: 등록 시점 시 확보시간/등록시간으로 등록이 불가능한 경우(확보시간/등록시간으로 산정)
- S6: 도서 보유 중이나 대출되어 있는 경우
- S7: 복원 도서 (대량) 산정(연립 도서 산정)
- S8: S1의 시나리오에서 인터페이스 변경에 의한 경우(시나리오에서 새로 도서를 찾는 경우)

<표 2> 시나리오별 프로세스 컴포넌트 클러스터

	C1	C2-1-1	C2-1-2	C2-2	C3	C4-1	C4-2	C5
Weighted value	1	1	1	1	1	1	1	1
Reusability weight	7	6	1	1	1	5	2	2

	S1	S2	S3	S4	S5	S6	S7	S8
#7	1	1	1	1	1	1	1	0
#6	1	1	1	1	1	0	0	1
#5	1	1	1	1	0	0	0	1
#2	0	1	1	1	0	0	0	0
#1	0	0	0	0	1	1	1	0
Sum	16	20	20	20	14	8	8	11

<표 3> 재사용성에 따른 중요 컴포넌트 측정

<표 3>은 각 프로세스 컴포넌트마다 가중치를 두고 재사용성에 따른 값을 계산하여 사용 빈도수가 많은 중요 컴포넌트를 찾는 것을 보여준다. 8개의 시나리오 중 S2~S4는 재사용성 값이 다른 시나리오에 비해 키 높은 재사용성을 보인다. 이 표에서 재사용성 값이 1인 프로세스 컴포넌트들이 있는데 이들은 재사용되지 않은 비공통 컴포넌트이지만 시스템에서 없어서는 안 될 컴포넌트일 수 있다.

제안된 방법을 통해 인간의 경험이나 선호도에 따른 컴포넌트의 재사용이 아니라 시스템에서 발생할 수 있는 시나리오들을 통해 효율적인 재사용 및 중요도를 측정하고자 한다.

6. 결론

이 논문은 레거시 시스템에 대한 도메인 분석을 위한 방법 제안으로써, 동적 작업 변화의 연속적인 흐름을 나타내는 워크플로우 기반의 도메인 모델링을 통해 공통/비공통 프로세스 컴포넌트를 추출하기

위한 방법을 제안하고자 한다. 도메인 분석 절차를 따라 점진적, 반복적으로 분석함으로써 각각의 사용자에 적당한 크기의 프로세스 컴포넌트를 추출하고 마지막 단계에서 UML 기법으로 컴포넌트 내의 객체를 추출할 수 있다. 또한 컴포넌트 가중치 측정 매트릭스를 통해 사용 빈도수가 많거나 중요한 공통/비공통 컴포넌트를 찾는 방법을 제시한다. 확장된 워크플로우 메커니즘을 통해 IT 분야, 비즈니스 모델링뿐만 아니라 병렬 시스템, 실시간 시스템도 모델링할 수 있으리라 기대한다.

향후 연구 과제는 도메인 분석을 통하여 추출된 공통/비공통 컴포넌트를 검증하기 위한 인증 연구와 워크플로우 모델링 언어 및 도구의 개발이다.

참고문헌

- [1] Katz S. et al, "Glossary of Software Terms", Gaithersburg, MD: National Institute of Standards and Technology, 1994.
- [2] 문미경, 엄근혁, "컴포넌트 기반 개발(CBD)을 위한 도메인 공학," 정보처리학회지, 2003.
- [3] 이재준, 강교철, "프로덕트 라인 소프트웨어 개발 프로세스," 정보과학회지, 2002.
- [4] S. Fraser, J. Coplien and J. White, Application of Domain Analysis to Object-Oriented System, ACM Press, 1995, pp 46-49.
- [5] Prieto-Diaz, "Domain Analysis : An Introduction," Software Engineering Notes 15, 1990.
- [6] K. Belhajjame, "A Flexible workflow model for process-oriented applications," IEEE, 2002.
- [7] 최은만, 김영철, 전병국, "국방 소프트웨어 컴포넌트 객체 웹 구조 및 플랫폼 기술 연구(최종보고서)," 국방과학연구소, 2003.
- [8] Y. Kim, R. Carlson, "Scenario Based Integration Testing for Object Oriented Software Development," The eighth Asian test symposium (ATS99), 1999, pp 283-288.
- [9] Y. Kim, J. Kim, R. Carlson, "Adaptive Design Based Testing," Proceedings of the ISCA 15th International Conference on Computers and their applications 2000, pp 165-168.
- [10] 김윤정, 전병국, 김영철, "워크플로우 메커니즘을 통한 소프트웨어 컴포넌트 식별에 관한 연구," 정보처리학회지, 2003.