

도메인 산출물의 효과적인 재사용을 지원하기 위한 도메인 Asset 저장소 개발

조정희⁰ 염근혁
부산대학교 컴퓨터공학과
(jh_cho⁰, yeom}@pusan.ac.kr

Development of Domain Asset Repository to Reuse Domain Artifacts Effectively

Junghee Cho⁰ Keunhyuk Yeom
Dept. of Computer Engineering, Pusan National University

요약

재사용을 목적으로 하는 도메인 공학과 재사용을 통한 컴포넌트 기반 소프트웨어 개발에 있어 재사용 저장소는 필수적 요소이며 재사용의 대상도 다양하다. 그러나 기존의 컴포넌트 저장소들은 소프트웨어 개발 주기의 다양한 산출물을을 컴포넌트의 일부분으로써 종속적으로 다루고 있음으로 인해 요구사항 수집에서 구현에 이르는 전 단계에서 도메인 산출물들의 재사용을 지원하기에 부족하다. 또한 기존의 도메인 저장소들은 도메인 산출물들 간 연관성 정의와 도메인 공통성 및 다양성 속성에 대한 기술이 부족하다.

따라서 본 논문에서는 소프트웨어의 전 개발 주기에서 재사용될 수 있는 도메인 산출물의 의미 있는 단위인 도메인 Asset들을 그것과 연관된 다른 도메인 Asset들과 함께 제공하고, 도메인 Asset이 가지는 특성에 대한 기술을 통해 도메인 산출물들의 효과적인 재사용을 지원하는 도메인 Asset 저장소 개발 방안을 제시한다.

1. 서 론

최근의 많은 소프트웨어는 기존 소프트웨어 컴포넌트의 조립을 통한 컴포넌트 기반으로 개발되고 있으며 재사용은 단지 기회적인 성공의 결과가 아니라 소프트웨어 개발 프로세스의 분리할 수 없는 부분이 되었다. 재사용은 그 방법이 조직적이고 계획적이어야 하며 이는 도메인 분석 및 아키텍처 개발의 결과로부터 식별되어지는 소프트웨어 컴포넌트에 대한 요구로부터 유도된다[1]. 이 때 재사용 저장소는 성공적인 재사용을 실현하는 중요한 요소로서 개발된 컴포넌트들에 대한 효과적인 정보 공유를 가능하게 한다. 도메인 공학은 도메인 분석 및 설계를 통해 소프트웨어의 재사용을 지원하는데, 이 과정에서 생성되는 산출물들은 서로의 연관성을 유지하며 저장되고 컴포넌트 기반 개발 시 유용한 정보가 된다[2].

소프트웨어 개발에서 재사용의 대상은 컴포넌트뿐만 아니라 아키텍처, 요구사항 명세, 방법, 도구 등 다양하지만[3], 기존의 컴포넌트 저장소들은 컴포넌트에 초점을 두고 그 외 산출물을은 컴포넌트의 일부로써 종속적으로 다루고 있어 컴포넌트 기반 개발의 요구사항 수집에서부터 구현에 이르는 전 단계에서 다양한 도메인 산출물들의 재사용을 지원하기에 부족하다. 또한 기존의 도메인 저장소들은 특정 도메인에 한정되어 있으며 도메인 공학 프로세스로부터의 산출물들 간 연관성 정의 및 도메인이 가지는 공통성 및 다양성 속성에 대한 기술이 부족하다.

따라서 본 논문에서는 소프트웨어 전 개발 주기에 재사용될 수 있는 도메인 산출물의 의미 있는 단위인 도메인 Asset들을 일반적인 전체 도메인에서 분류하고 연관성을 정의하여 하나의 도메인 Asset과 traceability를 갖는 다른 도메인 Asset들을 함께 제공하는 도메인 Asset 저장소 개발 방안을 제시한다. 그리

고 도메인 Asset의 공통성 및 다양성 속성에 대해 구체적으로 기술함으로써 도메인 산출물들의 효과적인 재사용을 지원하도록 한다.

2. 관련 연구

2.1 도메인 공학

도메인 공학은 유사한 시스템 집합(family)에 대하여 어플리케이션 개발의 중요한 Asset을 생성하기 위한 프로세스로서 이를 위한 모든 활동을 포함하고 있다[4]. 이 활동들은 유사한 시스템의 범위와 목적을 정하는 도메인 정의로부터 시작하여 도메인의 공통성과 다양성을 분석하는 도메인 모델링, 도메인 컴포넌트를 식별하고 도메인 아키텍처를 개발하는 도메인 설계, 그리고 구현 단계로 구성된다. 그 결과 도메인 명세, 도메인 모델, 도메인 아키텍처, 도메인 컴포넌트 등의 다양한 Asset들을 산출해낸다. 도메인 공학은 여러 어플리케이션에서의 재사용성을 위한 것인 만큼 도메인의 공통성과 다양성을 추출하는 것이 매우 중요하다.

2.2 컴포넌트 저장소

컴포넌트 저장소는 재사용 가능한 컴포넌트를 위치시키고 검색하며 관리를 지원하는 라이브러리 시스템으로서 소프트웨어의 신뢰성 및 생산성을 향상시키고 중복 개발을 최소화한다. 컴포넌트 저장소는 컴포넌트의 브라우징과 검색을 위해 사용자 중심의 GUI를 지원하는 자동화된 시스템이어야 하며 표준적인 컴포넌트 프레임워크를 지원한다. 또한 각 도메인에 대한 효과적인 분류 방안과 완전한 시스템 및 컴포넌트 문서를 포함한다[1]. 현재 상용적인 컴포넌트 저장소로는 ComponentSource[5]와 국내의 컴포넌트 정보 시스템[6], 컴포넌트 뱅크[7] 등이 있으나 대부분 소프트웨어 개발 주기의 다양한 산출물을 컴포넌트의 일부분으로써 종속적으로 다루고 있다.

3. 도메인 Asset 저장소

도메인 Asset 저장소는 도메인 분석 및 설계의 결과인 도메인 산출물을 도메인 Asset으로 정의하고 저장, 관리하는 수단이자 방법이다(그림 1). 이를 통해 재사용을 위한(for reuse) 소프트웨어 개발과 재사용을 통한(with reuse) 소프트웨어 개발을 지원한다.

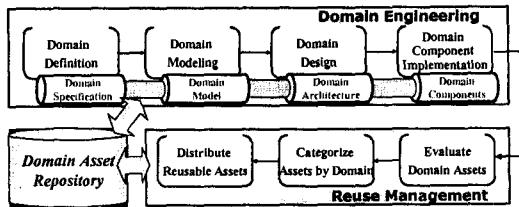


그림 1 도메인 Asset 저장소와 소프트웨어 재사용 활동들

3.1 도메인 Asset 정의

도메인 공학의 분석 및 설계 과정에서 다양한 도메인 산출물들이 생성되는데, 이들은 어플리케이션 개발 시 재사용될 수 있는 단위로써 저장소에서 저장, 관리되어야 한다. 이와 같은 의미 있는 재사용 단위를 도메인 Asset이라 하고 어플리케이션 개발, 특히 컴포넌트 기반 개발에서의 재사용성을 근거로 하여 도메인 요구사항, 도메인 컴포넌트, 도메인 아키텍처를 도메인 Asset으로 식별한다(그림 2). 도메인 요구사항은 도메인의 범위 및 목적을 기술하는 도메인 명세와 도메인 요구사항의 공통성 및 다양성을 분석한 도메인 유즈케이스 모델을 포함한다. 이는 특정 어플리케이션의 개발을 위해 요구사항 수집 및 정의 시 재사용의 범위를 정하고 관련 도메인 내 공통적인 개념 및 요구사항 정보를 활용할 수 있게 한다. 도메인 컴포넌트는 그것의 기능성 및 역할을 명확하게 기술하는 도메인 컴포넌트 명세와 도메인 인터페이스 명세를 필수적으로 포함하고 보다 더 자세한 정보를 위하여 도메인 컴포넌트 클래스 다이어그램과 도메인 객체 상호작용 다이어그램을 부가적으로 포함할 수 있다. 어플리케이션 개발자는 관련 도메인 내 주요 컴포넌트를 식별하고 컴포넌트에 기술된 다양한 선택 사항의 커스터마이징을 통해 특정 어플리케이션 맞도록 도메인 컴포넌트를 이용할 수 있다. 마지막으로 도메인 아키텍처는 이를 구성하는 도메인 컴포넌트 간의 상호작용을 보여주는 도메인 컴포넌트 상호작용 다이어그램과 정적 구조를 보여주는 도메인 컴포넌트 개발 다이어그램을 포함한다. 도메인 아키텍처는 어플리케이션 개발자가 도메인 내 여러 도메인 컴포넌트들 중 필요한 컴포넌트를 식별하거나 새로운 컴포넌트를 생성할 때 전체적인 구조를 제시하고 소프트웨어 아키텍처 개발의 참조를 제공한다.

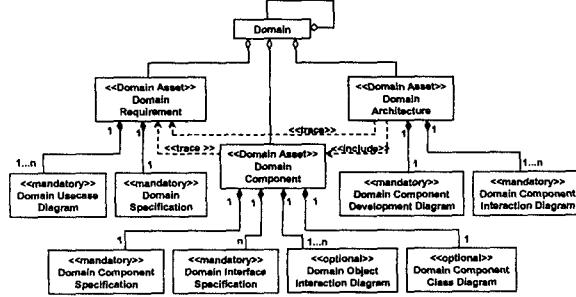


그림 2 도메인 Asset 저장 구조

식별된 도메인 Asset들은 일련의 도메인 공학 프로세스를 거쳐 생성된 것이므로 그들 사이에는 연관성을 가지고 있다. 본

도메인 Asset 저장소는 도메인 Asset을 검색할 때 이와 연관된 다른 도메인 Asset도 함께 제공한다. 이는 도메인 Asset의 재사용 시 그 도메인 Asset이 생성된 도메인 공학 프로세스 내 연관된 다른 종류의 도메인 Asset 정보를 참조할 때 더욱 효과적인 재사용이 가능하기 때문이다. 도메인 아키텍처와 도메인 컴포넌트는 도메인 요구사항을 trace하고 도메인 요구사항은 도메인 아키텍처와 도메인 컴포넌트를 realize하는 관계를 가진다. 그리고 도메인 아키텍처는 도메인 컴포넌트를 포함하는 관계를 가진다.

3.2 도메인 Asset 저장 구조

소프트웨어를 분류하는 것은 재사용하고자 하는 개발자들이 컴포넌트의 집합을 구조화할 수 있도록 하며 이는 컴포넌트의 검색을 용이하게 하므로[8] 도메인 Asset 저장소에 저장하는 도메인 Asset들을 위한 분류 체계가 필요하다. 도메인 Asset이 산출되는 도메인은 그 특성 및 범주에 따라 그림 3과 같이 분류된다.



그림 3 도메인 분류

먼저 도메인은 어플리케이션 도메인과 기반 서비스 도메인으로 나누어진다. 어플리케이션 도메인은 가장 일반적으로 다루어지는 비즈니스 도메인으로서 실세계의 도메인을 포함하며 고객 수준의 도메인이 된다. 금융의 계좌관리 · 고객관리, 의료의 예약관리 등이 이 도메인 분류에 속한다. 어플리케이션 도메인에 속한 도메인들 중에는 서로 공통적인 부분이 존재하게 되는데 이러한 부분을 공통 어플리케이션 도메인으로 분리시킴으로써 어플리케이션 도메인은 특정 어플리케이션 도메인과 공통 어플리케이션 도메인으로 다시 나누어지게 된다. 기반 서비스 도메인은 어플리케이션 도메인을 지원하는 서비스 도메인으로 개발자의 관점에서 볼 때 반드시 필요한 도메인이다. 통신 프로토콜, GUI, 장치 관리와 같은 전통적인 컴퓨터 과학 분야가 이 도메인 분류에 속한다.

분류된 각 도메인은 그 도메인의 특수한 분야에 따라 계층 구조를 가질 수 있고 따라서 상위 수준의 도메인은 다시 하위 수준의 도메인으로 분해 될 수 있다. 도메인 Asset 저장소는 이와 같이 계층화된 도메인에 도메인 Asset들이 서로 trace(역관계는 realize) 및 include의 관계를 가지며 포함되는 저장 구조를 가진다(그림 2). 이때 도메인 Asset들은 같은 수준의 도메인 상에서 서로 연관성을 가지며 서로 다른 도메인 수준 간의 도메인 Asset에 대한 연관성은 존재하지 않는다. 그리고 도메인 컴포넌트는 일련의 도메인 공학 프로세스의 결과 도메인 컴포넌트 집합으로써 하나의 도메인 수준에서 저장되며 그 중 완전히 독립적으로 하위 수준의 도메인에서 재사용 가능한 것들은 하위 수준의 도메인에 저장된다.

3.3 도메인 Asset Description

도메인 Asset 저장소에 저장되어 있는 도메인 Asset을 검색하기 위해서는 이를 도메인 Asset의 특징을 기술하는 도메인 Asset Description이 필요하다. 도메인 Asset Description은 도메인 Asset을 검색하는 수단인 동시에 검색된 도메인 Asset에 대한 자세한 정보를 제공하는 역할을 한다. 도메인 Asset Description은 도메인 Asset이 속한 도메인에 대한 정보를 비

롯하여 각 도메인 Asset의 종류 별 기능성 정보를 기술해야 한다. 또한 한 도메인 Asset이 그것과 연관된 다른 도메인 Asset에 대한 정보 및 유통, 관리, 유지 보수를 위한 부가 정보의 기술도 필요하다. 따라서 도메인 Asset Description은 표 1과 같이 정의된다.

표1 도메인 Asset Description

도메인 Asset 공통		도메인 Asset 공통	
분 류 정 보	• 도메인 Asset type • 도메인 계층	<ul style="list-style-type: none"> • 도메인 Asset 등록일 • 도메인 Asset 제작일 • 도메인 Asset 버전 • 도메인 Asset 풍급자 • 도메인 Asset 라이센스 제한 • 기술 및 개발 표준 	
		<th>도메인 요구사항</th> <th>도메인 아키텍처</th> <th>도메인 컴포넌트</th>	도메인 요구사항
의 미 정 보	<ul style="list-style-type: none"> • 도메인 범위 • 도메인 목적 • 도메인 개념 • 도메인 Usecase common • variable optional • Usecase with variable • 도메인 actor 	<ul style="list-style-type: none"> • 도메인 아키텍처 목적 • 기능성 정보(포함한 도메인 컴포넌트) <ul style="list-style-type: none"> - user service - business service - data service • 구조적 정보(포함한 도메인 컴포넌트) <ul style="list-style-type: none"> - common - variable - optional • 지원하는 품질 요소 	<ul style="list-style-type: none"> • 도메인 컴포넌트 목적 • 도메인 컴포넌트 솔루션 제공하는 인터페이스 • 사용하는 인터페이스 • 가변성
연 관 성 정 보	<ul style="list-style-type: none"> • realize하는 도메인 아키텍처 • realize하는 도메인 컴포넌트 	<ul style="list-style-type: none"> • trace하는 도메인 요구사항 • 포함되는 도메인 아키텍처 • Multiple version의 도메인 아키텍처 	<ul style="list-style-type: none"> • trace하는 도메인 요구사항 • 포함되는 도메인 아키텍처 • Multiple version의 도메인 컴포넌트

특히, 도메인 Asset이 가지는 도메인의 공통성과 다양성은 도메인 Asset Description의 의미 정보에서 자세히 기술되며 연관성 정보의 Multiple version의 도메인 아키텍처 및 도메인 컴포넌트는 다양성으로 인해 같은 타입의 도메인 Asset에서 다양하게 구현된 도메인 Asset들을 기술한다(필줄 표시). 웹 환경의 도메인 Asset 저장소는 XML을 통해 도메인 Asset Description의 문서 정보를 표준화하고 저장 정보의 관리를 효율적으로 한다.

3.4 도메인 Asset 검색

도메인 Asset 저장소는 도메인 Asset Description을 기반으로 사용자가 원하는 도메인 Asset 및 연관된 도메인 Asset을 검색하고, 검색된 도메인 Asset에 대한 재사용 형상 정보 분석 결과를 제공한다. 도메인 Asset을 검색하는 방법에는 사용자의 요청에 따른 분류 정보 기반 검색과 의미 정보 기반 검색이 있다. 그리고 검색된 도메인 Asset과 연관된 다른 도메인 Asset 검색을 위한 연관성 정보 기반 검색이 있다. 전체 도메인 Asset 검색 프로세스는 그림 4와 같다.

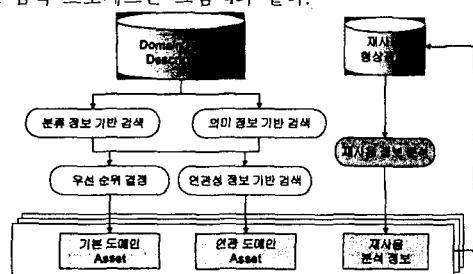


그림 4 도메인 Asset 검색 프로세스

분류 정보 기반 검색은 도메인 Asset이 분류되어 있는 카테고리를 통해 도메인 Asset을 검색하며 도메인 Asset Description의 분류 정보를 이용하여 사용자에게 도메인 Asset type과 도메인 계층을 facet으로 제시한다. 의미 정보 기반 검색은 도메인 Asset의 기능성과 같은 의미적 요소를 통해 도메인 Asset을 검색한다. 이때는 도메인 Asset 별로 도메인 요구사항의 도메인 개념 및 도메인 유즈케이스, 도메인 아키텍처의 기능성 정보 및 도메인 구조 정보, 그리고 도메인 컴포넌트의 제공·사용하는 도메인 인터페이스 요소에서 사용자가 입력한 도메인 Asset의 핵심 키워드와 일치하는 도메인 Asset들을 검색한다. 분류 정보 기반 검색과 의미 정보 기반 검색은 개별적으로 또는 함께 사용될 수 있다. 연관성 정보 기반 검색은 검색된 각 도메인 Asset과 연관성을 가지는 다른 도메인 Asset을 함께 검색하며 그 결과를 브라우징 방식으로 제공한다. 도메인 Asset 저장소의 재사용 형상 정보에는 사용자의 요청에 따른 도메인 Asset 검색 결과 및 로그(다운로드된 도메인 Asset, 도메인 Asset 사용자, 검색 키워드, 날짜 등)가 저장되는데 이를 통해 현재 검색된 도메인 Asset의 다운로드 횟수와 같은 도메인 Asset 사용 분석 정보를 제공한다. 그리고 도메인 Asset의 검색 결과가 하나 이상일 경우, 의미 정보 기반 검색에서 사용자의 요구 조건과 가장 많이 일치하는 도메인 Asset에 높은 순위를 부여함으로써 검색된 도메인 Asset에 대한 사용자의 이해를 돋는다.

4. 결론 및 향후 연구

본 논문에서는 소프트웨어 개발에서의 효과적인 재사용을 지원하기 위하여 컴포넌트뿐만 아니라 전체 소프트웨어 개발주기에서 재사용될 수 있는 도메인 Asset들을 도메인 공통성 및 다양성 속성을 반영하여 기술하고 도메인 공학 프로세스에서 연관성을 갖는 다른 도메인 Asset과 함께 저장, 검색하는 도메인 Asset 저장소 개발 방안을 제시하였다. 향후 연구 과제로서 제시한 저장소의 실질적 구현과 함께 저장소 관리 및 도메인 Asset의 검증에 대한 연구가 필요하다.

참고 문헌

- [1] Guo, J., Luqi, "A Survey of Software Reuse Repositories", Seventh IEEE International Conference and Workshop on Engineering of Computer Based Systems(ECBS2000), pp. 92-100, 2000
- [2] M.K., Moon, K.H., Yeom, "Domain Design Method to Support Effective Reuse in Component-Based Software Development", 1st International Conference on Software Engineering Research & Applications(ICSER'03), 2003.
- [3] Wei, L.K., Jarzabek, S., "A Generic Discretionary Access Control System for Reuse Framework", Twenty-Second Annual International Computer Software and Application Conference(COMPSAC'98), pp. 356-361, 1998
- [4] SEI in Carnegie Mellon University, Domain Engineering, http://www.sei.cmu.edu/domain-engineering/domain_eng.html
- [5] ComponentSource, <http://www.componentsource.com>
- [6] 컴포넌트 정보 시스템, <http://www.componis.com>
- [7] 컴포넌트 뱅크, <http://www.component-bank.com>
- [8] Poulin, J.S., Yglesias, K.P., "Experiences with a Faceted Classification Scheme in a Large Reusable Software Library (LSL)", Seventh Annual International Computer Software and Application Conference(COMPSAC'93), pp. 90-99, 1993.