

SOA환경에서 레거시 시스템 컴포넌트의 재활용도 측정

이성환[ⓐ], 김우주[ⓑ]

[ⓐ] 연세대학교 정보산업공학과 박사과정
120-749, 서울 서대문구 신촌동 134

Tel: +82-2-2123-7754, Fax: +82-2-364-7807, E-mail:sunghwan@yonsei.ac.kr

[ⓑ] 연세대학교 정보산업공학과 교수
120-749, 서울 서대문구 신촌동 134

Tel: +82-2-2123-7754, Fax: +82-2-364-7807, E-mail:wkim@yonsei.ac.kr

Abstract

SOA(Service Oriented Architecture)는 컴포넌트와 서비스의 재활용을 최대화하도록 설계된 아키텍처이다. 서비스와 컴포넌트의 재활용뿐만 아니라 기존 레거시 시스템의 재활용도 주요한 이익의 하나이다. 레거시 시스템의 활용도는 프로젝트 내에서 레거시 컴포넌트를 재사용한 횟수와 레거시 컴포넌트를 다른 시스템이나 프로젝트에서 재사용한 횟수를 더하고 이를 프로젝트의 전체 서비스 컴포넌트 수로 나눠서 계산할 수 있다. 본 연구에서 도출한 레거시 컴포넌트의 재활용도 측정항 후 SOA기반의 프로젝트를 수행하고 난 후 기존의 레거시 시스템을 어느정도 재활용하였는지 측정할 수 있게 되어 IT투자 분석의 기초 자료로 활용될 수 있을 것이다.

Keywords:

레거시 시스템 재사용, SOA, Service Oriented Architecture, legacy system reuse, measurement of legacy system reuse

1. 서론

소프트웨어의 재사용은 생산성을 높이고 시간을 절약할 수 있는 효율적인 방법이다. 그러나 소프트웨어의 재사용은 많은 요소가 포함되어 있는 개념이며, 그 효과를 측정하는 것에 대해서는 여전히 많은 연구가 필요한 주제이다. SOA(Service Oriented Architecture)는 컴포넌트와 서비스의 재활용을 최대화하도록 설계된 아키텍처이다. 현재 SOA에서 서비스의 재활용을 측정하는 방법은 제시되어 있으나 기존 레거시 컴포넌트의 재활용을 측정하는 방법은 아직 제시되고 있지 않다. 본 연구에서는 레거시 컴포넌트의 재활용을 측정하는

방법을 제시하고자 한다.

2. 관련 연구

2.1 컴포넌트(Component)의 정의

많은 산업계에서 컴포넌트 기반 패러다임을 발전시켰다. 그 중에 자동차 산업을 예로 들면 차를 생산할 때 컴포넌트(모듈이나 조립부품으로 알려진)를 사용한다. 사실 최종 생산의 단계는 알고 있는 것처럼 "조립"이다. 건축 산업에서도 조립과 유사한 방법을 쓰는데 창문과 문의 조립은 미리 만들어진 것을 현장에 설치한다. PC 하드웨어 산업에서도 컴포넌트 방법론을 도입하여 컴퓨터의 개별 부품을 업그레이드 할 수 있다. 컴포넌트 방법론은 설계와 생산의 관점에서뿐만 아니라 소비의 측면에서도 효과적이며 능률적이다. 예를 들어 자동차 산업에서 새로운 엔진을 각 모델 별로 디자인 하는 것보다 여러 자동차 모델을 위한 엔진 관리 컴포넌트를 디자인하는 것이 더 효과적이다. 또한 컴포넌트를 조립하여 차를 생산하는 것이 수작업 방식보다 더 효과적이다.[1]

2.2 소프트웨어 컴포넌트(Component)의 정의

소프트웨어 산업계에서 컴포넌트 방법을 시도해왔으나 아직까지 정해진 산업표준은 없다. 약 10년 전에 컴포넌트 기반 개발 (component based development (CBD)) 운동이 특정한 경우에 적용할 수 있는 표준을 만들기 위한 운동이 일어났으나 그 표준은 다양하게 적용하기에는 불충분하였다.

소프트웨어 컴포넌트란 비즈니스나 기술적 기능을 제공하는 소프트웨어의 단위를 말한다. 소프트웨어

컴포넌트는 어플리케이션 개발을 위한 재사용 가능한 블록이다. 소프트웨어 컴포넌트는 일반적으로 다음 4가지로 구성된다.

- (1) 컴포넌트가 무엇을 하는지 정의하는 설명 (사용자나 소비자 관점에서의 프로세스와 데이터 모델을 기술)
- (2) 컴포넌트의 내부 디자인 구현,
- (3) 배포할 수 있는 실행 가능한 모듈,
- (4) 컴포넌트의 기능에 접근을 제공하는 하나 이상의 인터페이스

컴포넌트의 핵심 컨셉은 구현이 인터페이스 뒤에 숨어있어서 컴포넌트의 사용자는 컴포넌트의 자세한 내용은 알지 않아도 된다. [1]

다음 그림1은 컴포넌트 기반 재사용의 예시이다.

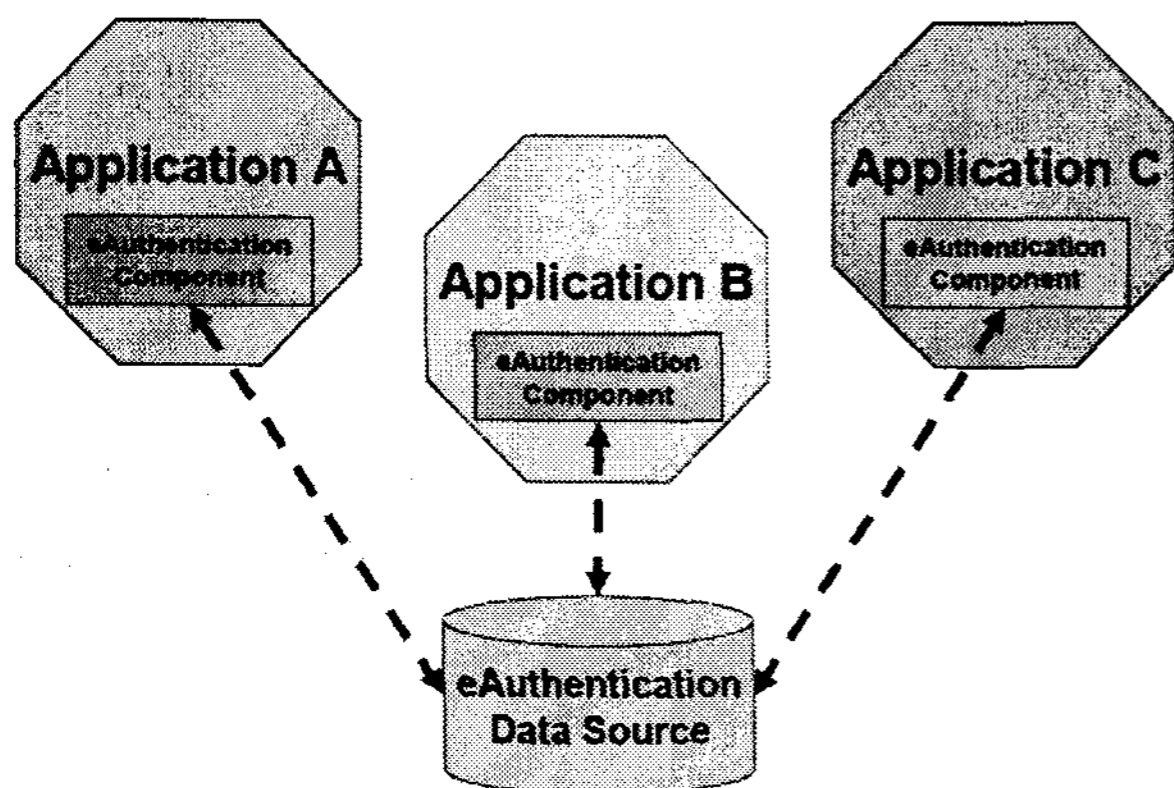


그림1 컴포넌트 기반 재사용의 예시

일반적으로 재사용이 가능한 컴포넌트는 구조가 정의되고 컴포넌트를 구별하기 위한 이름이 부여된 디자인의 단위로 정의할 수 있다. 또한, 디자인 문서의 형태로 되어있는 디자인 가이드라인은 컴포넌트의 재활용을 지원하며 어떻게 그 컴포넌트가 재활용 될 수 있는지에 대한 내용을 기술한다. (예를 들어 어느 다른 컴포넌트가 반드시 같이 사용되어야 하는지에 대한 제약 조건) [2]

2.3 서비스(Services)의 정의

서비스의 목적은 공급자/소비자 모델에 기반하여 비즈니스나 기술적 요구사항을 만족시키는 것이다. 서비스는 컴포넌트 보다 넓은 개념이다. 서비스는 어떤 결과를 제공하는 것이 아니라 요청에 응답하는 행동을 한다. 모든 컴포넌트는 서비스와 같은 기능을 제공하지만 모든 서비스는 컴포넌트로 구성되어 있지 않다. 다음 그림2는 서비스와 컴포넌트의 차이를 그린 것이다. 서비스는 비즈니스 요청으로부터 요구되지만 컴포넌트는 서비스를 제공하기 위한 방법이다. 예를 들어 서비스를 구현하는 방법 중 하나는 어떤 레거시 기능에

인터페이스를 붙이는 것이다. 레거시 시스템은 비구조적이고 (개별적인 컴포넌트로 구별되어 있지 않을 수 있음) 새로운 어플리케이션이나 다른 서비스 요구에 대한 인터페이스가 없을 수 있다.

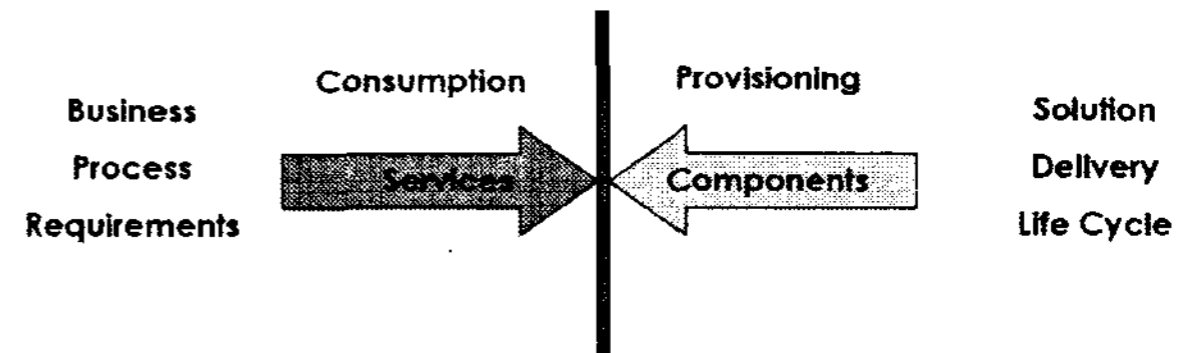


그림2 서비스와 컴포넌트의 차이

2.4 레거시(legacy) 시스템의 정의

레거시(legacy)는 과거에 전개되었던 기존의 IT자산(asset)을 의미한다. 이러한 자산들은 20년 전 또는 어제까지 설치되었던 모든 것들을 의미하고, 대부분 레거시 자산들은 중요한 비즈니스 프로세스를 실행하고 있다. 레거시 소프트웨어와 애플리케이션들은 종종 엔터프라이즈의 "cash cow(돈벌이가 되는 것)"로 간주되기 때문에 레거시 소프트웨어와 애플리케이션들은 기업의 운영 이익에도 큰 책임이 있다. 이러한 이윤은 레거시 자산을 관리하는데 필요한 비용을 훨씬 초과하고, 초과분은 기업이 다른 전략적 이니셔티브에 자금을 투자하는데 사용된다. 게다가, 레거시 소프트웨어나 애플리케이션들은 인수 합병의 결과로 엔터프라이즈로 들어온다. 많은 경우, 애플리케이션의 개발과 관리에 책임이 있는 사람들은 더 이상 라이프 사이클에는 책임이 없다.

레거시 인프라스트럭처는 메인프레임 애플리케이션과 서비스에만 국한될 필요가 없다. 레거시 애플리케이션들은 기업에 있어서 중요한 비즈니스 가치를 나타낸다. 예를 들어, 2천억 줄의 COBOL 코드가 있고, 3백억 COBOL 기반 트랜잭션들이 매일 처리되고 있다.

레거시 시스템들은 다음 패턴들 중 한 개 이상을 기반으로 구현된다.

- 그린 스크린(Green screen): 메인프레임 CICS® 시스템에 대한 3270 터미널 액세스 같은 전형적인 문자 기반 애플리케이션이다.
- 팻 클라이언트(Fat client): Windows®-기반 PC 같은 클라이언트 워크스테이션 기반으로 실행되는 그래픽 UI를 사용하여 단일 레거시 애플리케이션에 액세스 할 수 있다.
- 다중 세션: 비즈니스 프로세스가 하나의 사용자가 여러 애플리케이션들에 걸쳐 여러 개의 활성 세션들을 가질 것을 요구하고, 사용자는 사전 정의된 순서대로 액티비티를 실행해야

한다.

- 많은 P2P: 레거시 애플리케이션을 가진 인터페이스는 레거시 시스템과 종속 시스템간 고유한 p2p 인터페이스를 진화시켰다.
- 레거시 제약 조건: 기술적 구현은 비즈니스의 확장 기능을 제한하는 수단이다.
- 사장된 레거시: 늘어나는 사용자 요구를 레거시 애플리케이션이 수용할 수 없기 때문에(레거시 제약 조건 참조) 분산 애플리케이션들이 비즈니스 프로세스와 데이터가 다중 시스템들간 분산되는 방식으로 레거시 애플리케이션들을 확장 및 매장했다.

레거시 인에이블먼트(enablement)란 기존 소프트웨어와 정보 에셋들을 활용하여 새로운 비즈니스 프로세스에 사용될 수 있도록 하는 프로세스이다. [3]

2.5 서비스 재활용 공식

CIO council에서는 SOA 환경하에서 컴포넌트의 재활용을 측정하기 위한 reuse metric을 제시하였다. 이 식은 수식(1)에 정의되어 있으며 값이 0에 가까우면 재활용이 낮음을 의미하고 1 이상이면 재활용이 높음을 의미한다. 이 수식은 고의적으로 요구조건의 특이성이나, 어려운 임무, 서비스의 범위 같은 재활용 수준에 영향을 미칠 수 있는 많은 요인을 무시하고 있다. 이러한 요인들을 고려하려면 측정이 매우 복잡해지며 간단히 계산, 추적, 사용하기가 어렵다. 모든 프로젝트가 재사용성이 높아야 하는 것은 아니다. 목적과 대상을 선정할 때 특정 프로그램과 목표가 반드시 고려되어야 한다.

$$\text{reuse quotient} = \frac{\left(\begin{array}{l} \text{total number of} \\ \text{"service-components"} \\ \text{in project reused FROM} \\ \text{other systems or processes} \end{array} \right) + \left(\begin{array}{l} \text{total number of} \\ \text{"service-components"} \\ \text{in project reused IN} \\ \text{other systems or processes} \end{array} \right)}{\text{total number of "service-components" in project}} \quad (1)$$

3. 레거시 시스템 컴포넌트의 재활용 측정

3.1 레거시 시스템 컴포넌트의 재활용 공식

본 연구에서는 서비스 컴포넌트의 재활용 공식을 변형하여 레거시 컴포넌트의 재활용 공식을 도출하여 보았다. 그 식은 수식 (2)와 같다.

$$\text{레거시 시스템의 재활용} = \frac{\left(\begin{array}{l} \text{시스템이나 프로세스} \\ \text{내에서 사용된} \\ \text{"레거시 컴포넌트"의 총합} \end{array} \right) + \left(\begin{array}{l} \text{다른 시스템이나 프로세스에서} \\ \text{가져가서 사용된} \\ \text{"레거시 컴포넌트"의 총합} \end{array} \right)}{\text{프로젝트내에서 "서비스 컴포넌트"의 총합}} \quad (2)$$

값이 0에 가까우면 레거시 시스템의 재활용이 낮음을 의미하고 1 이상이면 재활용이 높음을 의미한다.

3.2 레거시 시스템 컴포넌트 재활용도 예시

레거시 시스템 컴포넌트의 재활용도는 다음과 같이 측정할 수 있다.

(1) 레거시 컴포넌트를 서비스 컴포넌트로 패키징하여 생성하고 사용하는 경우

1) 1개의 레거시 컴포넌트를 서비스 컴포넌트화 하여 사용하고 다른 프로젝트에서 만든 2개의 서비스 컴포넌트를 가져다 사용하는 경우

$$\frac{(1)+(0)}{(3)} = 0.33 \quad (3)$$

2) 이후 새로 만든 1개의 레거시 컴포넌트를 publish한 후 다른 2곳의 프로젝트에서 호출하여 사용하는 경우

$$\frac{(1)+(2)}{(3)} = 1 \quad (4)$$

(2) 레거시 프로세스를 레거시 컴포넌트로 재패키징 하는 경우

1) 기존에 사용하던 레거시 프로세스를 레거시 컴포넌트로 만들고 신규로 1개의 새로운 서비스 컴포넌트를 만들어 사용하는 경우

$$\frac{(1)+(0)}{(2)} = 0.5 \quad (5)$$

2) 이후 레거시 컴포넌트가 다른 1곳의 프로젝트에서 호출되어 사용된 경우

$$\frac{(1)+(1)}{(2)} = 1 \quad (6)$$

4. 결론

본 연구에서는 레거시 시스템 컴포넌트의 재활용도를 측정하기 위하여 기존의 서비스 컴포넌트 재활용도 측정 방법을 수정하여 레거시

시스템 컴포넌트의 재활용 측정식을 만들었다. 도출한 레거시 시스템 컴포넌트의 재활용도 측정 공식은 간단한 측정을 위하여 재활용에 영향을 미치는 여러 조건들이 제외되어 있지만 간단한 수식을 통해 기존 레거시 시스템 컴포넌트의 재활용도를 측정하는 방법을 제시하였기에 의미가 있다고 하겠다. 본 연구에서 도출된 레거시 시스템 컴포넌트의 재활용도 측정방법은 정보시스템의 효과분석을 위한 기본 자료로 활용될 수 있을 것이다.

참고문헌

- [1] The Federal CIO Council Architecture and Infrastructure Committee (2006). *Services and Components Based Architectures: Version 3.5*. Federal Chief Information Officers Council
- [2] Casati F, Castano S, Fugini MG, Mirbel-Sanchez I, Pernici B (2000). "Using patterns to design rules in workflows". IEEE Transactions on Software Engineering, Vol. 26, NO. 8
- [3] Calvin Lawrence (2007). "Adapting legacy systems for SOA" IBM
<http://www.ibm.com/developerworks/webservices/library/ws-soa-adaptleg/>