

매시업(Mashup) 방식의 서비스 어플리케이션 품질모델

오상현, 김수동
숭실대학교 대학원 컴퓨터학과

An Quality Model for Mashup-Based Service Application

Sang Hun Oh and Soo Dong Kim

Soongsil University

E-mail: shoh@otlab.ssu.ac.kr, sdkim777@gmail.com

요 약

일반적으로 알려진 SOA의 장점으로는 개발비용 절감, 기민성, 확장 용이성, 비즈니스 수준의 재사용 등이 있다. 특히 서비스의 재사용은 다양한 시각에서 비추어 지고 있으며, 서비스 재사용 기술 중 일반적이고 많이 사용되는 방법 중 하나가 매시업(Mashup)이다. 그러나, 기존의 서비스나 정보들을 잘 융합하여 사용자가 만족할 수 있는 서비스나 어플리케이션이 모두 개발되는 것은 아니다. 매시업이 되는 과정 중 여러 곳에서 문제점이 발생할 수 있기 때문이다. 따라서 사용자가 만족할 수 있는 고품질의 서비스를 제공하기 위해 본 논문에서는 매시업 방식의 서비스 어플리케이션을 평가하는 품질모델을 제안하여, 이를 기반으로 보다 높은 수준의 매시업 방식의 어플리케이션을 사용자에게 제공할 수 있다.

1. 서론

현재 서비스 지향 아키텍처(Service-Oriented Architecture, SOA)에서 서비스는 재사용의 한 방법으로 많이 이용되고 있으며, 현재도 웹에서는 다양하고 많은 서비스들이 사용자들에게 제공되고 있다[1][2]. 특히 서비스의 재사용은 다양한 시각에서 비추어 지고 있으며, 서비스 재사용 기술 중 일반적이고 많이 사용되는 방법 중 하나가 매시업(Mashup)이다. 이 매시업은 웹에서 제공하고 있

는 정보와 서비스를 융합하여 새로운 소프트웨어나 서비스, 데이터베이스 등을 만드는 것을 말한다[3]. 매시업의 장점은 새로운 서비스 제작을 위해 투입되는 비용이 저렴하고, 오픈API라는 공개된 자원을 활용하기 때문에 적은 노력으로 큰 효과를 거둘 수 있어 경제성이 높다[4]. 그러나, 서비스 매시업 기반의 개발이 끝난 후에 발생할 수 있는 인터페이스 mismatch, 데이터 에러, 성능 등 다양한 곳에서 문제점이 발생할 수 있다[5].

따라서 이러한 문제가 발생되지 않고, 사용자에게 고품질의 서비스를 제공하기 위해 본 논문에서는 매시업을 평가하기 위한 품질모델(Quality of Mashup, QoM)을 제안한다.

2. 관련연구

Braga의 연구에서는 서비스들 간의 조합 시 발생 할 수 있는 문제점 및 한계 그리고 다양한 인터페이스와 파라미터 타입등 매시업 방식의 새로운 이슈에 대한 해결책을 제시하였다[3].

Yu의 연구에서는 매시업 개발 시 사용되었던 개발 도구 및 프레임워크들을 설명하였고, 이들에 대한 문제점을 제시하였다[4]. 또한 앞으로 매시업 기반의 개발을 보다 쉽게 하기 위한 접근 방법 등을 제시하였다.

3. 서비스의 특징

본 장에서는 여러 논문들과 책을 기반으로 하여 서비스가 가지고 있는 고유한 특징들을 도출한다. 다음 그림 1은 서비스가 가지고 있는 고유한 특징들 이다.

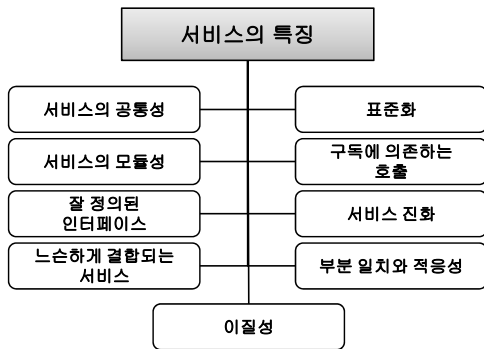


그림 1. 서비스의 특징들

서비스의 공통성은 SOA의 기본적인 개념은 다양한 어플리케이션들간에 공통적인 기능을 가지는 서비스를 개발하고 배포하여, 이 서비스를 재사용하는 것이다[1]. 서비스의 모듈성은 SOA에서 서비스는 잘 모듈화된 기능 단위로서, 복잡한 과정을 거치지 않고 다양한 비즈니스 프로세스에 조합될 수 있다[1]. 잘 정의된 인터페이스는 서비스는 서비스 내부에 대한 상세한 내용을 외부에 노출시키지 않고, 서비스 소비자가 서비스의 기능을 명

확하게 이해할 수 있도록 잘 정의된 인터페이스를 제공한다[1][2]. 인터페이스 명세서는 메소드 시그니처 외에 시멘틱 명세와 제약사항도 포함되어야 한다. 느슨하게 결합되는 서비스는 SOA에서 서비스 제공자와 서비스 소비자 간의 사전 지식 없이 서비스 제공자는 서비스를 배포하고, 서비스 구매자는 서비스를 구독할 수 있다[2]. 이질성은 서비스는 다양한 방법, 또는 여러 서비스 제공자에 의해 설계되고 구현되며, 다양한 개발 언어 또는 플랫폼을 사용하여 구현된다[1][2]. 표준화는 서비스가 다양한 플랫폼을 기반으로 여러 벤더들에 의해 제공되기 때문에, 서비스들은 특별한 제약 없이 서로 조합되기 위해 SOA의 표준을 준수하는 것이 필수적이다[1]. 구독에 의존하는 호출은 SOA 환경에서 필요할 때 마다 서비스 소비자가 적절한 서비스를 찾아서 호출하며 서비스 사용에 대한 비용을 지불한다[2]. 서비스 진화는 서비스 저장소에 이미 배포된 서비스를 서비스 소비자의 동의 없이 진화 할 수 있다[1]. 부분 일치와 적응성은 서비스는 공통적인 기능을 제공하는 단위이다. 그러나 실제 서비스를 사용할 때에는 서비스 소비자들 간에도 이미 배포된 서비스에 대해 요구하는 것이 약간 다를 수 있다[1][2].

4. QoM의 품질속성

본 장에서는 3장에서 정의한 서비스 특징을 기반으로 하여 매시업 방식으로 개발된 서비스 어플리케이션을 평가하기 위한 QoM의 품질속성을 도출한다.

사용성: 사용성은 서비스 특징들 중에서 공통성과 모듈성으로부터 도출되었다. 매시업 방식을 통해서 개발된 어플리케이션의 사용성이 높기 위해서는 공통성과 모듈성이 좋아야 한다. 특히 어플리케이션의 사용성이 높기 위해서는 다른 곳에서도 많이 재사용되어야 하기 때문에 공통적으로 사용될 수 있는 기능들이 많이 포함되어야 한다.

신뢰성: 신뢰성은 서비스 특징들 중에서 표준화와 구독에 의존하는 호출로부터 도출되었다. 매시

업 방식을 통해 개발된 어플리케이션의 신뢰성이 높기 위해서는 개발 당시 SOA의 표준을 얼마나 준수하였는지에 따라 어플리케이션의 신뢰성이 좌우된다.

조합성: 조합성은 서비스 특징들 중 모듈성과 이질성 그리고 잘 정의된 인터페이스로부터 도출되었다. 이러한 특징들이 도출된 이유는 메시업 기반으로 개발될 당시 서비스들 사이에서의 인터페이스가 잘 정의되어 있지 않다면, 부적절하게 조합이 될 것이고, 또한 서비스가 모듈성이 낮다면 많은 시간과 비용을 통해 조합이 될 것이고, 특정 개발언어나 플랫폼에 한정되어 있다면 메시업하는 과정에서 많은 어려움이 따르게 된다.

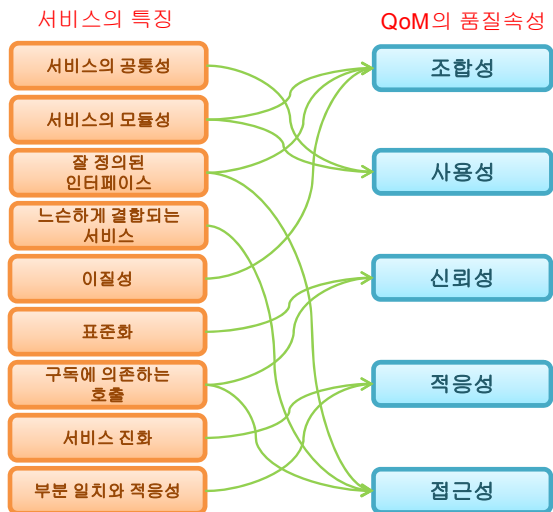


그림 2. 서비스의 특징과 QoM 속성 간의 맵핑

접근성: 접근성은 서비스 특징들 중 잘 정의된 인터페이스와 느슨하게 결합되는 서비스 그리고 구독에 의존하는 호출로부터 도출되었다. 메시업 기반으로 개발된 어플리케이션은 다른 서비스들과의 상호운영이 잘 되어야 한다. 그러기 위해서는 인터페이스가 잘 정의되어 있어야 하며, 특히 서비스 제공자와 서비스 소비자 간의 사전 지식 없이 서비스 제공자는 서비스를 배포하고 서비스 소비자는 서비스를 구독할 수 있어야 한다. 마지막으로 서비스 구매자가 필요할 때 마다 서비스 소비자가 적절한 서비스를 찾아서 호출 할 수 있어야 한다. 위와 같은 환경이 조성되어야 메시업 기

반으로 개발된 어플리케이션의 접근성이 높다.

적응성: 적응성은 서비스 특징들 중 부분 일치와 적응성 그리고 서비스 진화로부터 도출되었다. 메시업 기반으로 개발된 어플리케이션은 다양한 서비스 소비자에게 서비스를 동적으로 적응시킬 수 있어야 하며, 또한 이미 개발된 서비스 어플리케이션은 서비스 소비자의 동의 없이 진화되어야 하고, 동적으로 발견될 수 있어야 한다.

5. QoM을 위한 메트릭

본 장에서는 4장에서 정의된 QoM의 품질속성들을 기반으로 하여 QoM의 메트릭을 도출한다. 다음 그림 3은 QoM의 품질속성을 측정하는 메트릭간의 관계를 나타낸다.

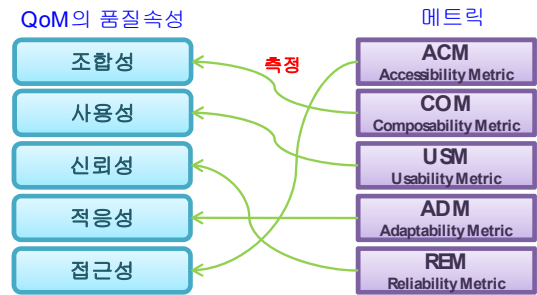


그림 3. QoM의 품질속성과 메트릭간의 관계

사용성 (Usability Metric, USM): 메시업 방식으로 개발된 어플리케이션은 다른 곳에서도 많이 재사용되어야 하기 때문에 얼마나 많은 기능들이 공통적으로 재사용될 수 있는지를 측정하는 메트릭이다. ($0 < USM < 1$, 1에 가까울수록 사용성은 높음)

$$USM = \frac{\text{공통으로 재사용될 수 있는 기능들의 수}}{\text{전체기능들의 수}}$$

USM의 메트릭에서 분모는 어플리케이션이 가지고 있는 전체기능들의 수 이고, 사용성이 높기 위해서는 분자 즉, 공통성으로 재사용될 수 있는 기능들의 수가 높아야 메시업 방식으로 개발된 어플리케이션의 사용성은 높아진다.

신뢰성 (Reliability Metric, REM): 메시업 방식으로 개발된 어플리케이션의 신뢰성은 SOA의 표준을 얼마나 준수하였는지에 따라 신뢰성이 좌우 될 것이다. ($0 < REM < 1$, 1에 가까울수록 신뢰성은 높

음)

$$REM = \frac{\text{메시업 개발 시 적용된 표준들의 횟수}}{\text{SOA와 관련된 표준들의 수}}$$

REM의 메트릭에서 분모는 SOA와 관련된 표준들의 수 이고, 이 수는 메시업 방식의 서비스 어플리케이션 품질을 평가 하는 품질평가자에 의해 결정된다. 또한 분자는 메시업 기반의 어플리케이션 개발 시 적용된 표준들의 횟수 이다. 즉, 적용된 표준들의 횟수가 높아야 신뢰성이 높아진다.

조합성(Composability Metric, COM): 메시업 방식으로 개발된 어플리케이션의 조합성은 다른 서비스들과의 조합이 얼마나 높은지를 측정하는 메트릭이다. ($0 < COM < 1$, 1에 가까울수록 조합성은 높음)

$$COM = \frac{\text{다른 서비스와 조합된 기능들의 수}}{\text{다른 서비스와 조합될 수 있는 기능들의 수}}$$

COM의 메트릭에서 분모는 메시업 기반의 어플리케이션이 다른 서비스와 조합될 수 있는 기능들의 수 이고, 분자는 조합될 수 있는 기능들 중에 실제 조합 된 기능들의 수이며, 분자의 수가 높아야 조합성이 높아진다.

접근성(Accessibility Metric, ACM): 접근성은 서비스 소비자들이 접근하기 쉽고 발견이 잘 될 수 있도록 인터페이스가 잘 정의 되어 있어야 하며 서비스 소비자가 호출했을 때 빠른 응답속도를 가져야 한다. 다음은 접근성을 측정하기 위한 메트릭이다. ($0 < ACM < 1$, 1에 가까울수록 접근성은 높음)

$$ACM = \frac{\text{서비스 소비자들이 접근 성공한 횟수}}{\text{서비스 소비자들이 접근한 전체 횟수}}$$

ACM의 메트릭에서 분모는 소비자들이 서비스를 이용하기 위해 접근했던 횟수이고, 분자는 소비자들이 접근했을 때 접근이 성공한 횟수를 나타내며 분자의 횟수가 클수록 메시업 기반 어플리케이션의 접근성은 높아진다.

적응성(Adaptability Metric, ADM): 메시업 방식으로 개발된 어플리케이션은 다양한 서비스 소비

자에게 서비스를 동적으로 적응시킬 수 있어야 한다. 다음은 적응성을 측정하기 위한 메트릭이다. ($0 < ADM < 1$, 1에 가까울수록 접근성은 높음)

$$ADM = \frac{\text{적응시킬 기능들의 수}}{\text{적응시킬 수 있는 전체 기능들의 수}}$$

ADM의 메트릭에서 분모는 메시업 기반의 어플리케이션이 가지고 있는 기능들 중 적응이 가능한 기능들의 수 이고, 분자는 이러한 기능들 중에 실제 적응이 된 기능들의 수 이다. 따라서, 분자의 수가 클수록 메시업 기반 어플리케이션의 적응성은 높아진다.

6. 결론

현재 서비스의 재사용은 다양한 시각에서 비추어지고 있으며, 서비스 재사용 기술 중 일반적이고 많이 사용되는 방법 중 하나가 메시업(Mashup)이다. 본 논문을 통해서 메시업 기반의 어플리케이션 개발을 품질평가 하는 품질모델을 제시하였다. 또한 후속연구로 품질모델을 기반으로 제시된 메트릭을 검증하고 적용하여 사례연구가 현재 진행 중에 있다.

[참고문헌]

- [1] Erl, T., *SOA Principles of Service Design*, Prentice Hall, 2008.
- [2] Erl, T., *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, 2005.
- [3] Braga, D., Ceri, S., and Martinenghi, D., "Mashing Up Search Services," *IEEE Internet Computing*, Vol. 12, No. 5, pp. 16-23, 2008.
- [4] Yu, J., Benatallah, B., Casati, F., and Daniel, F., "Understanding Mashup Development," *IEEE Internet Computing*, Vol. 12, No. 5, pp. 44-52, 2008.
- [5] Liu, X., Hui, Y., Sun, W., and Liang, H., "Towards Service Composition Based on Mashup," *Proceedings of the IEEE Congress on Services (SERVICES 2007)*, pp. 332-339, Beijing, China, July 9-13, 2007.