

# SOA기반 SW 기능성의 품질평가 메트릭 개발

강배근\*, 신석규\*\*, 양해술\*

\*호서대학교, \*\*한국정보통신기술협회

e-mail : dandyishkang@gmail.com, skshin@tta.or.kr hsyang@hoseo.edu

## Quality estimation metrik development of SOA based SW functionality

Bae-Keun, Kang\* Seok-Kyoo Shin\*\* Hae-Sool Yang\*

\*Hoseo University,

\*\*SQEC, TTA

### 요 약

SOA는 특정 기술이나 플랫폼에 종속되지 않고 느슨한 결합(Loosely Coupled)을 가지고 상호 연동할 수 있는 서비스들의 조합으로 어플리케이션 개발을 가능하게 하는 정보시스템 아키텍처이다. 즉, 한 덩어리의 방대한 코드로 이루어진 어플리케이션들을 각각 개발하는 대신 각각의 비즈니스 기능을 수행하는 서비스를 구성하고, 이 서비스를 조합하거나 분리함으로써 비즈니스 프로세스들을 구현할 수 있게 하는 정보시스템 구축을 목표로 한다. 본 연구에서는 SOA기반 소프트웨어의 분야의 기반 기술을 조사하고 SOA기반 소프트웨어 시장과 표준화 동향을 조사하며 SOA기반 소프트웨어 기능성의 품질평가 메트릭을 개발하였다.

### 1. 서 론

### 2. 관련 연구

서비스 지향은 “아키텍처”와 결합할 때 비로소 기술적인 의미를 지니게 된다. “서비스 지향 아키텍처”는 자동화 로직이 작고 특정한 로직으로 분할할 수 있는 모델을 뜻한다. 결론적으로, 이러한 단위들이 모여 더 커다란 비즈니스 자동화 로직을 구성한다. 각 단위 로직은 분산 환경에 배치될 수 있다.

자동화 로직을 각 로직 단위로 분산 구성하는 것은 예전부터 해왔던 일이다. 쇼핑몰과 같은 비즈니스 커뮤니티 내부의 개별 상점들이 분산되어 있더라도 상점 간에 비즈니스 의존성이 지나치게 되면, 개별 상점의 잠재된 가능성이 드러나지 못할 수 있다. 개별 상점이 서비스 차원에서 상호작용하고 서로 돕는 것은 바람직하지만, 상점들의 형태 자체가 단단하게 결합되어 지나치게 내부적으로 비즈니스 간에 의존은 피해야 한다.

개별 상점이 스스로 자신의 서비스를 제어할 수 있는 권한을 개별 상점에 부여하면 독립성을 가지되 서로 유기적으로 상호작용하면서 성장하고 진화할 수 있다.

#### 2.1 SOA기반 소프트웨어의 특성

##### 2.1.1. 발견과 동적 바인딩

SOA는 “서비스의 발견과 동적 바인딩”이라는 개념을 지원한다. 어떤 서비스를 필요로 하는 사용자는 작업 수행 중(runtime)에 필요로 하는 서비스를 찾고, 그것을 사용할 수 있는 것이다.

예를 들어 금융관련 애플리케이션에서 신용카드 기능을 포함시키고자 한다고 하면, 금융 애플리케이션은 레지스트리에서 신용카드 기능을 지원하는 서비스를 조회한 후, 자신에게 맞는 서비스를 골라 동적으로 바인딩하게 된다.

##### 2.1.2 모듈 독립성

SOA에서 사용하는 서비스는 컴포넌트에서와 마찬가지로 독립적인 모듈이다. 각각의 서비스는 독립적으로 개발, 유지, 관리되며, 서로의 작동 자체에 큰 영향을 미치지 않는다.

이런 점은 특정 서비스를 수정했을 때 발생하는 파문 효과(ripple effect)를 최소화 시켜 결합도를 낮추는 중요한 특성이 된다.

### 2.1.3 상호 운용성

SOA에서의 서비스는 플랫폼에 관계없이 상호운용이 가능하다. 각 서비스는 자신을 호출할 수 있는 인터페이스를 제공하고 있다. 이때 호출이 이루어지는 프로토콜과 호출 메시지의 포맷만 이해할 수 있다면 서로 다른 플랫폼 위에서 개발, 운영되는 서비스끼리도 통신이 가능한 것이다.

### 2.1.4. 느슨한 연결

각 모듈을 연결하는 방법에는 밀접한 연결(tightly coupling)과 느슨한 연결(loose coupling)이라는 두 가지 방법이 있다. 뒤에 다시 언급하겠지만 느슨한 연결 방법은 단단한 연결 방법에 비해 모듈간의 결합도를 낮춘다.

결합도가 높을수록 한 모듈의 변화가 다른 모듈에도 영향을 주어 파문효과(ripple effect)를 일으키게 되는데, 파문 효과가 클수록 시스템을 유지보수는 어려워진다. 파문 효과는 하나에 영향이 생겼을 때 다른 것들에 많은 영향을 끼치는 것이다.

대개의 소프트웨어 구조는 결합도가 낮은 것을 지향하는데, SOA에서는 서비스 사용자와 제공자를 느슨하게 연결해 결합도를 낮출 수 있도록 하고 있다.

## 2.2 SOA기반 소프트웨어 기술동향

### 2.2.1 표준 서비스 인프라

SOA는 표준 서비스 인프라를 제공한다. 즉, 제공하는 서비스에 대한 인터페이스를 XML 기반의 표준 형태로 기술할 수 있으며, 메시지도 XML 기반의 표준 메시지 포맷을 사용하여 통신하게 된다. 제공 서비스가 인터넷과 같은 글로벌 네트워크상에 어디든 존재할 수 있다. 서비스 사용자는 표준화된 서비스 발견 방식에 따라서 어디서든지 원하는 서비스를 찾아 사용이 가능하다.

### 2.2.2 약결합(Loosely-Coupled) 형태의 서비스 연결

SOA는 약결합 형태의 서비스 연결을 지원한다. 서비스 간 결합도가 높을수록 한 모듈의 변화가 다른 모듈에도 영향을 주어 파문 효과를 일으키게 되는데, 파문 효과가 클수록 시스템을 유지보수하기가 어려워진다. 약결합 형태의 서비스 연결을 위해 SOA는 먼저 서비스 인터페이스와 서비스 내부설계 및 구현을 분리하는 Black Box 형태의 서비스를 지향한다. Black Box 형태의 서비스에서 클라이언트는 서비스가 제공하는 인터페이스가 명시하는 규약들만

을 이해하고 서비스 내부의 여러 규약들에 대해서는 이해할 필요가 없다.

두 번째로 약결합 형태의 서비스 연결을 위해서는 서비스 레지스트리와 관련해 서비스 발견, 동적 바인딩을 지원함으로써 서비스 위치 투명성을 제공한다. 세 번째로 약결합 형태의 서비스 연결을 위해 SOA는 여러 인프라에 독립성을 제공해야 하며 운영 중에 서비스 발견, 융합, 확인 등을 제공한다.

### 2.2.3 서비스 재사용성

SOA는 서비스 재사용을 위하여 비즈니스 서비스 모델과 적합한 정도로 큰 크기의 서비스를 제공한다. 서비스의 재사용성 문제는 이미 객체지향이나 컴포넌트 기반기술의 주요 이슈 중의 하나이다. 서비스 요청자와 제공자가 동의한 한 개 이상의 인터페이스를 가지고 있다는 면에서 서비스는 클래스나 컴포넌트와 동일하다. 하지만, 서비스는 클래스나 컴포넌트와 같이 어떤 객체의 타입에 대한 어떤 것이 아니고 스키마에 대한 것이며 또한, 메서드 호출에 대한 것이 아니라 메시지에 대한 것이다.

더불어 서비스 크기는 객체나 컴포넌트 보다 일반적으로 크다. 이러한 서비스의 크기에 대한 결정은 서비스에 대한 중요한 설계상의 결정문제이다. 서비스 크기가 잘못된다면 그들이 필요한 것 이상의 더 많은 서비스 호출을 해야 하는 불상사가 생길 수 있으며 또한 서비스 레벨상의 보안 관련 문제가 생길 수 있다. 그리고 잘못 사용된 경우 어떤 메서드로부터 특정 사용자를 제한하는 것이 불가능할 수도 있다. 하지만 이러한 서비스 크기를 결정할 때 중요한 문제는 서비스의 유지보수를 헤치지 말아야 한다는 것이다.

## 3. SOA기반 SW의 기능성의 품질특성

이절에서는 ISO/IEC 9126의 품질특성 체계를 기반으로 품질특성과 부특성의 체계에 따라 매트릭을 정의하고 개발하였으며 SOA기반 소프트웨어의 기능성의 품질특성을 분류하고 분석하여 품질특성 체계를 정리하였다.

### 3.1 기능성의 품질특성

기능성은 소프트웨어가 주어진 조건에서 사용될 때 명시적 혹은 묵시적으로 요구된 기능을 제공하는 능력으로써 <표 1>과 같이 부특성으로는 적합성,

정확성, 상호운용성, 보안성, 준수성 등이 있다.

[표 1] 기능성의 품질 특성

| 주특성  | 부특성                      |   |
|--|--------------------------|---|
| 기능성 (Functionality)                                | 적합성 (Suitability)        | 사용자의 목적하는 바에 따라 적절한 기능을 제공하는 능력               |
| 소프트웨어가 주어진 조건에서 사용될 때 명시적 혹은 묵시적으로 요구된 기능을 제공하는 능력 | 정확성 (Accuracy)           | 올바른 또는 합의도니 결과를 제공하는 능력                       |
|  | 상호운용성 (Interoperability) | 하나 이상의 지정된 타 시스템과 서로 작동하는 능력                  |
|  | 보안성 (security)           | 인가되지 않은 사람이나 시스템의 액세스를 방지하여 정보 및 데이터를 보호하는 능력 |
|  | 준수성 (Compliance)         | 그 응용에 관한 표준, 규정, 관례 등을 따르는 능력                 |

### 3.2 기능성의 품질평가 메트릭

#### 3.2.1 적합성

(1) 기능정보 제공 : 제품 설명서와 사용자 문서에 SOA기반 SW 시스템의 소프트웨어적인 기능과 하드웨어적인 기능에 대해 언급하고 있습니까?

|       |                       |   |
|-------|-----------------------|---|
| 측정 항목 | A                     | SOA기반 소프트웨어에 의해 제공되는 모든 기능의 수<br>- 다른 경로로 접근되는 동일한 기능의 수는 중복 계산하지 않는다.(소프트웨어 방식과 하드웨어 방식의 운영 메커니즘의 기능 포함)                 |
|       | B                     | 사용자 문서에 언급된 소프트웨어에 의해 제공되는 기능의 수<br>- SOA기반 소프트웨어의 메뉴, 단축키, 스마트 카드, IC칩 등을 통해 나타난 기능을 확인하고 사용자 문서로부터 관련 기능에 대한 정보를 대응시킨다. |
| 계산식   | 기능 정보 제공(FDI) = B/A   |   |
| 결과 영역 | 0 ≤ 기능 정보 제공(FDI) ≤ 1 | 결과값   |

(2) 기능 충분성 : 평가 대상 유형의 SOA기반 SW 시스템에 필요한 필수적인 소프트웨어 기능이 충분히 구현되어 있습니까?

|       |                     |  |
|-------|---------------------|--|
| 측정 항목 | A                   | 유사한 유형의 SOA기반 SW 시스템들이 갖는 공통적인 필수 기능의 수<br>(필수 기능의 유형)<br>- 단말기 성능한계상 낮은 보안수준을 거절하는 기능<br>- 통신 품질을 표시하는 기능<br>- 통신 품질을 개선하는 방법의 구현<br>- 정보의 유출, 왜곡, 손상 방지 기능 |
|       | B                   | 필수 기능 중 대상 시스템에 구현된 기능의 수<br>- 필수 기능으로 정의된 기능 중에서 평가 대상 SOA기반 소프트웨어에 구현된 기능의 수   |
| 계산식   | 기능 충분성(FSL) = B / A |  |
| 결과 영역 | 0 ≤ 기능 충분성(FSL) ≤ 1 | 결과값  |

#### 3.2.2 정확성

(1) 기능구현 정확성 정보 제공 : 사용자 문서에 SOA기반 SW 시스템의 각 기능에 대한 정보가 정

확하게 기술되어 있습니까?

|       |                             |   |
|-------|-----------------------------|---|
| 측정 항목 | A                           | SOA기반 SW 시스템에서 제공하는 전체 기능 수<br>- 다른 경로로 접근되는 동일한 기능의 수는 중복 계산하지 않는다.(메뉴, 단축키, 스마트 카드 등을 통한 접근이 모두 가능한 기능의 경우) |
|       | B                           | 사용자 문서에 정확히 기술되어 있는 기능 수<br>- 구현된 기능과 문서에 기술된 명세를 비교하여 판단   |
| 계산식   | 기능 구현 정확성 정보 제공(AIP) = B/A  |   |
| 결과 영역 | 0 ≤ 기능구현 정확성 정보 제공(AIP) ≤ 1 | 결과값   |

(2) SOA기반 소프트웨어 정확성 : SOA기반 소프트웨어 시스템이 환경(장소)의 변화에 영향을 받더라도 상거래 결과에 큰 차이가 발생하지 않습니까?

|       |  |   |
|-------|--|---|
| 측정 항목 | A  | 환경 변화를 시도한 테스트케이스의 수<br>- 일반적인 적용상황에서 발생할 수 있는 수준으로 SOA기반 환경을 변경하여 테스트케이트를 작성하여 시도한 수 |
|       | B  | 테스트케이스 중 SOA기반 SW에 성공한 경우의 수<br>- 테스트케이스를 이용하여 테스트를 수행하여 성공한 경우를 체크                   |
| 계산식   | - SOA기반 SW 정확성(LRA) = B/A<br>$B = \frac{\sum_{i=1}^A \text{Success\_TC}_i}{\text{Total\_TC}_i}$<br>- Success_TC : i번째 기능 확인을 위해 수행한 테스트 케이스 중 성공한 건 수<br>- Total_TC : i번째 기능 확인을 위해 수행한 테스트케이스 수 |   |
| 결과 영역 | 0 ≤ SOA기반 SW 정확성(LRA) ≤ 1  | 결과값   |

#### 3.2.3 상호운영성

(1) 연결 가능성 : SOA기반 소프트웨어가 다른 SOA기반 소프트웨어와 어느 정도 연결되어 사용될 수 있습니까?(S/W형인 경우)

|       |                     |   |
|-------|---------------------|---|
| 측정 항목 | A                   | 연결가능하리라 기대되는 SOA기반 소프트웨어의 평균 SOA기반 SW 장치의 수<br>- SOA기반 SW 서비스 유형에 따라 해당 유형의 SOA기반 소프트웨어들이 평균적으로 지원하는 SOA기반 SW 장치의 수를 체크   |
|       | B                   | 평가 대상의 SOA기반 소프트웨어가 실제 연결 가능한 SOA기반 SW 장치의 수<br>- 평균 연결 가능 장치 중, 평가 대상 SOA기반 소프트웨어에 연결되어 정상적으로 기대한 업무를 수행할 수 있는 장치의 수를 체크 |
| 계산식   | 연결 가능성(POC) = B / A |   |
| 결과 영역 | 0 ≤ 연결 가능성(POC) ≤ 1 | 결과값   |

(2) 데이터 교환성 : 다른 관련 소프트웨어와의 데이터 교환이 어느 정도 가능합니까?

|       |   |  |
|-------|---|--|
| 측정 항목 | A | 데이터 교환이 가능한 관련 소프트웨어의 수<br>(데이터 교환이 가능한 관련 소프트웨어)<br>- 다른 소프트웨어(또는 시스템)와 휴대 단말기가 연계되어 상거래를 처리하여 품질을 높일 수 있는 소프트웨어의 수 |
|       | B | 각 항목별 테스트케이스가 성공한 경우의 수  |

|       |                      |   |
|-------|----------------------|---|
|       |                      | - 데이터 교환이 가능하다고 명시된 관련 소프트웨어에 대해 테스트케이스를 작성하여 테스트를 수행하고 성공한 경우를 체크  |
| 계산식   |                      | - 데이터 교환성 (DEC) = B/A<br>$- B = \frac{\sum_{i=1}^A Success\_TC_i}{Total\_TC_i}$ - Success_TC : i번째 기능 확인을 위해 수행한 테스트 케이스 중 성공한 건 수<br>- Total_TC : i번째 기능 확인을 위해 수행한 테스트케이스 수 |
| 결과 영역 | 0 ≤ 데이터 교환성(DEC) ≤ 1 | 결과값   |

### 3.2.4 보안성

(1) 접근 통제 가능성 : SOA기반 소프트웨어에 접근 통제를 위한 기능이 구현되어 있습니까?

|       |                       |  |
|-------|-----------------------|--|
| 측정 항목 | A                     | 접근 통제를 위한 모든 기능 수<br><br>(체크할 기능의 예)<br>- SOA기반 시스템에서 비승인된 자에게 노출되지 않도록 지분 및 거래 메시지의 기밀을 유지하는 기능<br>- 거래사실을 부인하지 못하게 하는 기능   |
|       | B                     | 각 항목별 테스트케이스 성공률의 합<br>- 접근 통제 기능에 대한 테스트케이스를 시험하여 성공한 경우를 체크  |
| 계산식   |                       | - 접근통제 가능성 (ACC) = B/A<br>$- B = \frac{\sum_{i=1}^A Success\_TC_i}{Total\_TC_i}$ - Success_TC : i 번째 기능 확인을 위해 수행한 테스트케이스 중 성공한 건 수<br>- Total_TC : i번째 기능 확인을 위해 수행한 테스트케이스 수 |
| 결과 영역 | 0 ≤ 접근통제 가능성(ACC) ≤ 1 | 결과값  |

(2) 프라이버시 가능성 : SOA기반 소프트웨어에 프라이버시가 어느 정도 가능합니까?

|       |                       |  |
|-------|-----------------------|--|
| 측정 항목 | A                     | 프라이버시가 가능하기 위한 모든 기능 수<br><br>(체크할 기능의 예)<br>- 이용자의 구매에 관한 프라이버시가 상점이나 은행이 결탁해도 노출되지 않도록 하는 기능<br>- 은행과 상점이 어떠한 공모를 행하더라도 지분한 이용자의 지분정보와 인출정보는 서로 연결될 수 없어야 하는 기능              |
|       | B                     | 각 항목별 테스트케이스 성공률의 합<br>- 프라이버시 기능에 대한 테스트케이스를 시험하여 성공한 경우를 체크  |
| 계산식   |                       | - 프라이버시가능성(PCR) = B/A<br>$- B = \frac{\sum_{i=1}^A Success\_TC_i}{Total\_TC_i}$ - Success_TC : i 번째 기능 확인을 위해 수행한 테스트 케이스 중 성공한 건 수<br>- Total_TC : i번째 기능 확인을 위해 수행한 테스트케이스 수 |
| 결과 영역 | 0 ≤ 프라이버시가능성(PCR) ≤ 1 | 결과값  |

### 3.2.5 준수성

(1) 기능표준 준수율 : SOA기반 시스템이 문서에 기술된 대로 기능 표준 준수성을 구현하였습니까?

|       |                       |   |
|-------|-----------------------|---|
| 측정 항목 | A                     | 평가할 기능표준 준수 항목 수<br><br>(다음과 같은 유형의 정보 제공 여부를 파악)<br>- 기능표준 준수와 관련된 정보<br>- SOA기반 시스템이 준수하는 규정, 기준 및 사용지침   |
|       | B                     | 각 항목별 테스트케이스 성공률의 합<br>- 테스트케이스를 시험하여 성공한 경우를 체크  |
| 계산식   |                       | - 기능표준 준수율(FCR) = B/A<br>$- B = \frac{\sum_{i=1}^A Success\_TC_i}{Total\_TC_i}$ - Success_TC : i번째 기능 확인을 위해 수행한 테스트 케이스 중 성공한 건 수<br>- Total_TC : i번째 기능 확인을 위해 수행한 테스트케이스 수 |
| 결과 영역 | 0 ≤ 기능표준 준수율(FCR) ≤ 1 | 결과값   |

## 4. 결론

SOA 기반 소프트웨어에 대한 제품 인증 체계가 구축되기 위해서는 먼저 품질 시험을 위한 측정 방법과 기준에 대한 연구가 선행되어야 한다. 국내에서 패키지 소프트웨어 분야를 필두로 소프트웨어 품질시험 방법에 대한 연구에 많은 진전이 있었으며 초기단계의 품질인증 서비스가 진행되고 있지만 다양한 소프트웨어 분야를 전반적으로 커버할 수 있는 수준에 이르기 위해서는 향후 지속적인 연구 개발이 이루어져야 할 것이다.

본 연구에서는 SOA기반 소프트웨어의 분야의 기반 기술을 조사하고 SOA기반 소프트웨어 시장과 표준화 동향을 조사하며 SOA기반 소프트웨어 기능성의 품질평가 메트릭을 개발하였다. 향후 실질적인 활용을 통해 고품질 소프트웨어의 개발을 촉진함으로써 높은 부가가치를 창출하고 국제적으로 경쟁력을 갖춘 제품의 개발을 지원할 수 있을 것으로 기대한다.

### 참 고 문 헌

- [1] 더크 크래프지그, 칼 방케, 더크 슬라마, “엔터프라이즈 SOA(서비스 지향 아키텍처 베스트 프랙티스)”, 태극미디어, 2006.11
- [2] 토마스 얼, “SOA : 서비스 지향 아키텍처(XML과 웹서비스 통합을 위한 필드 가이드)”, 성안당, 2007. 1.
- [3] 이하용, 양해술, 황석형, “패키지 소프트웨어 시험을 위한 ISO/IEC 12119의 적용”, , 정보처리학회 학술발표논문집, 2000. 5.
- [4] 이하용, 양해술, 황석형, “패키지 소프트웨어 시험평가 모듈의 개발”, 한국정보처리학회 학술발표논문집, 2001. 10.