

# SOA 기반 서비스 식별을 위한 상향식 접근

이현주\*, 최병주\*, 이정원\*\*  
 \*이화여자대학교 컴퓨터정보통신학과  
 \*\*아주대학교 전자공학과

e-mail : [hj\\_lee@ewhain.net](mailto:hj_lee@ewhain.net), [bjchoi@ewha.ac.kr](mailto:bjchoi@ewha.ac.kr), [jungwony@ajou.ac.kr](mailto:jungwony@ajou.ac.kr)

## A Bottom-up Approach for Service identification on SOA

Hyeon-joo Lee\*, Byoung-ju Choi\*, Jung-won Lee\*\*

\*Dept. of Computer Science and Engineering, Ewha Womans University

\*\*Dept. of Electronic Engineering, Ajou University

### 요 약

서비스 지향 아키텍처(SOA)는 기업이 적정한 비용으로 오늘날 가장 중요한 IT 요구사항인 민첩성과 유연성을 만족시킬 수 있는 IT 아키텍처 수립의 토대로, 경영환경이 빠르게 급변하는 최근에 떠오른 이슈이다. 기존의 서비스 지향 아키텍처의 서비스 개발은 주로 비즈니스 환경에서 어플리케이션방향으로 접근하는 하향식 방법이 대부분이었으나, 본 논문에서는 이미 개발된 컴포넌트 기반 시스템에서 접근하는 상향식 서비스 식별법을 제안한다. 본 논문에서 제안하는 방법은 사용자의 이벤트 정보를 담고 있는 GUI Flow-Event 를 이용하여 컴포넌트를 서비스로 식별하고, 서비스 식별시 발생하는 서비스간 연관관계에 따른 문제점을 최소화함으로써 비즈니스 도메인에 더욱 가깝고, 약결합을 지향하는 서비스로 식별할 수 있다.

### 1. 서론

서비스 지향 아키텍처는 기업의 인프라의 복잡성 및 유지비용을 최소화하고, 기업의 생산성과 유연성을 극대화할 수 있는 방안으로 경영환경이 빠르게 급변하는 최근에 각광 받고 있다.

서비스 지향 아키텍처의 서비스 식별에 대한 연구는 SOA 를 지원하기 위한 개발 방법론(Service Based Development) 등에 언급되어 있으나[1][2][3][4], 개발 절차상의 서비스 식별 방법을 하향식(Top-down method), 상향식(Bottom-up method), 절충형(Hybrid method) 방법으로 구분할 뿐 세부적인 절차나 활동을 명시하지 않고[5], 구체적이라 하더라도 대부분 하향식 방법론에 대해 제시하고 있다[6].

서비스 지향 아키텍처는 서비스 간에 약결합(Loosely-Coupled)을 지향하고 서비스간의 의존도를 지양(statelessness)한다. 그러므로 서비스간 결합도가 낮아야 한다. 물론 이러한 특성은 비단 서비스 지향 아키텍처에만 해당하는 것이 아니라 컴포넌트 개발 방법론에서도 강조되어 온 특성이다. 그러나 컴포넌트는 기능적인 단위의 최소 결합도를 가지도록 식별되는 반면, 서비스는 독립적인 비즈니스 가치를 창출할 수 있는 단위로서 기능보다는 한 차원 위에서 비즈니스 규칙과 그 프로세스를 포함하므로 컴포넌트와 구별된다[7]. 그러므로 기업이 전자 시스템을 서비스 지향 아키텍처로 전환할 때, 기존 컴포넌트를 그대로 서비스로 정의할 것이 아니라, 비즈니스 서비스 모델과 적합한 정도의 큰 크기(Granularity)를 가진 서비스로 컴포넌트를 합성(Composition) 또는 분할(Partitioning)해야

한다.

본 논문에서는 시스템의 GUI(Graphical User Interface)가 시스템의 기능보다 한 차원 위에서 사용자의 행위를 관찰할 수 있도록 한다는 점에 착안, GUI 정보를 이용하여 기존의 컴포넌트를 서비스로 식별하는 상향식 서비스 식별 방법에 대해 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서는 제안하는 ‘SOA 를 위한 컴포넌트 시스템의 서비스 식별법’을 기술한다. 3 장에서는 본 논문에서 제시한 방법을 경영정보시스템에 적용한 사례를 통한 결과를 분석하고, 끝으로 4 장에서 결론을 맺는다.

### 2. SOA 를 위한 서비스 식별

레거시 컴포넌트 시스템을 SOA 로 구축하기 위한 GUI 기반 서비스 식별 과정은 그림 1 에서 보는 바와 같이 크게 후보 서비스 식별을 위한 준비 단계인 전처리 과정과 후보 서비스를 식별하고 정제하는 서비스 식별과정으로 나누며 이를 통해 최종 서비스를 식별한다. 여기서 말하는 최종 서비스란 상향식 서비스 식별 방법론을 통해 얻어진 서비스를 말한다.

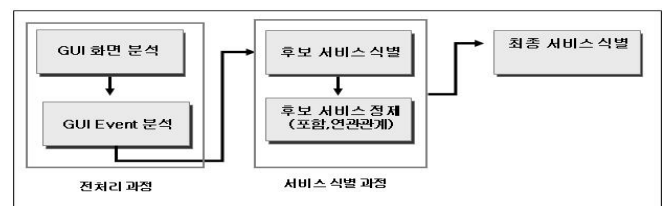


그림 1. 상향식 서비스 식별 과정

## 2.1. 전처리 과정

서비스 후보를 식별하기 위해서는 먼저, 식별할 대상인 GUI 화면에 대한 분석이 이루어져야 한다. GUI 화면은 단순히 정보를 보여주지만 하는 Display 영역과 화면 변환이 일어나게 하는 Interactive 영역으로 나누어지는데 그 각각에는 표, 그래프, 검색영역, 그 외 기타 영역으로 구분되고 세부적으로 라디오버튼, 버튼, 텍스트, 체크박스, 콘텐츠, 아이콘, 자료 입력 창, 선택 등이 있다.

다음으로 각 GUI 가 발생하는 이벤트를 GUI Event-Flow Diagram 로 나타낸다. 그림 2 에서 보는 바와 같이 GUI Event-Flow Diagram 은 출력이벤트와 입력이벤트를 화살표 방향으로 나타낸 것이다. 이는 가장 많은 이벤트를 발생시키는 GUI 화면이 무엇 인지와 GUI 화면 간의 이벤트관계가 어떤지를 시각적으로 보여준다.

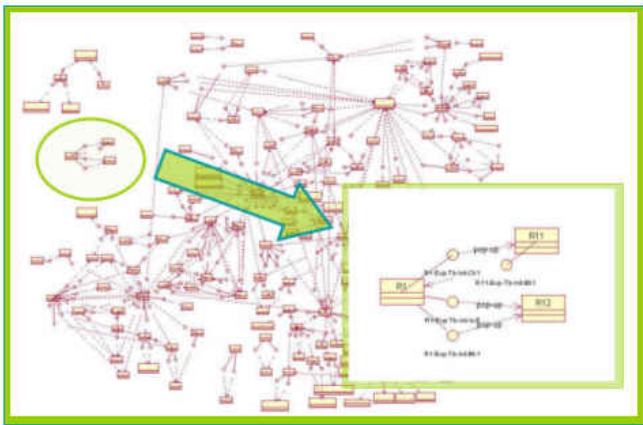


그림 2. GUI Event-Flow Diagram

## 2.2. 서비스 식별 과정

전처리 과정을 통해 작성된 GUI Event-Flow Diagram 을 이용하여 1 차 후보 서비스를 식별하고 식별된 후보 서비스를 정제하여 2 차 서비스와 3 차 서비스를 식별하여 궁극적으로 최종 서비스로 식별해내는 과정이다.

후보 서비스는 서비스간 출력 이벤트 발생 최다빈도수를 기준으로 식별한다. GUI Event-Flow Diagram 을 참고하여 출력 이벤트 발생 수에 따라 GUI 화면들을 정리한 후, 출력이벤트발생빈도가 많은 GUI 화면들을 일정한 기준에 따라 CoreRoot 로 정한다. CoreRoot 를 기준으로 CoreRoot 에서 출력되는 이벤트와 연결된 GUI 화면들을 모두 묶어 하나의 1 차 후보 서비스로 식별한다.

다음으로, 식별된 1 차 후보 서비스 사이에 발생하는 포함관계 분석하여 2 차 후보 서비스를 식별하고, 2 차 후보 서비스간의 연관 관계를 분석하여 3 차 후보 서비스를 식별한다. 포함관계와 연관관계는 1 차 후보 서비스의 조합 사이에 발생하는 결합도(정의 1)와 최종 서비스의 개념과 크기를 고려하여 임의로 정한 기준결합도로부터 구한다.

### 정의 1. 결합도

1 차 후보 서비스 간에 중복되는 GUI 화면의 비율

### 정의 2. 포함관계

결합도가 기준 결합도 보다 큰 경우로, 두 후보 서비스간 결합된 GUI 화면수가 많은 경우

### 정의 3. 연관관계

결합도가 기준 결합도 보다 작은 경우로, 두 후보 서비스간 결합된 GUI 화면수가 적은 경우

그림 3 에서 보는 바와 같이 두 1 차 후보 서비스간의 관계가 포함관계(정의 2)인 경우를 먼저 파악한 후 하나의 2 차 후보 서비스로 식별한다. 그리고, 2 차 후보 서비스들 간에 발생하는 연관관계(정의 3)를 최소화하여 3 차 후보 서비스로 식별함으로써 최종 서비스 식별이 이루어진다. 즉, 2 차 후보 서비스간에 연관 관계에 있는 GUI 화면들을 각각의 2 차 후보 서비스에 중복하도록 하여 3 차 후보 서비스로 식별하거나, 또는 중복된 부분을 변형시켜 3 차 후보 서비스로 식별한다.

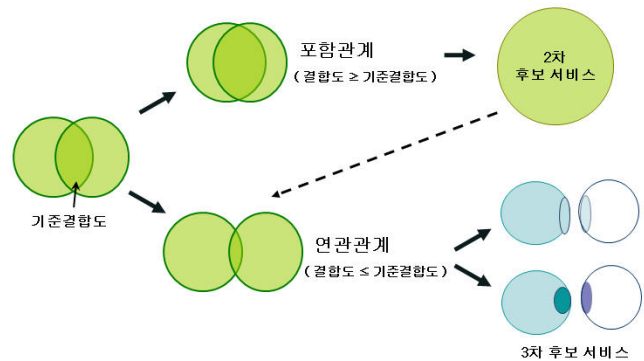


그림 3. 후보 서비스 식별

2 차 후보 서비스간에 연관관계 발생 부위는 다음과 같이 4 가지 경우로 분류할 수 있다.

1. 한 후보서비스의 Root 에서 다른 후보 서비스의 Root 를 호출한 경우
2. 한 후보서비스의 Root 에서 다른 후보서비스의 Subset 을 호출한 경우
3. 한 후보서비스의 Subset 에서 다른 후보서비스의 Root 를 호출한 경우
4. 한 후보서비스의 Subset 에서 다른 후보서비스의 Subset 을 호출한 경우.

연관관계가 발생하는 부분에는 이벤트 발생 유형에 따라 크고 작은 문제점이 발생하는데, 이벤트 발생 유형은 단순히 정보를 보여주지만 하는 참조(Retrieval)와 데이터베이스의 변형을 일으켜서 GUI 컴포넌트의 서비스 식별시 좀 더 신중해야 하는 수정, 생성, 삭제(Update)로 나눌 수 있다.

2 차 후보 서비스간에 발생하는 연관관계에 따른 문제점을 어떻게 해결할 것인가는 최종후보서비스의 질을 결정하는 데 중요한 역할을 한다. 서비스 지향 아

키텍처는 서비스간에 약결합(Loosely-Coupled)을 지향하며, 서비스간 결합도가 높으면 한 모듈의 변화에 따른 다른 모듈로의 파장이 커지게 되어 시스템의 유지보수가 어려워지기 때문이다.

### 3. 적용 사례 및 분석

국내 기업의 실제 사용되고 있는 경영정보시스템에 본 논문에서 제시한 방법론을 적용해 보았다. 이 경영정보시스템은 129 개의 GUI 로 구성되어 있고, 13 개의 컴포넌트로 이루어져 있었다.

본 논문에서 제시한 방법에 따라 1 차 후보 서비스를 식별한 결과 6 개의 1 차 후보 서비스가 식별되었다. 식별된 6 개의 1 차 후보 서비스간의 결합도는 표 1 과 같았다.

표 1. 1 차 후보 서비스간 결합도

MIS	CS1_1	CS1_2	CS1_3	CS1_4	CS1_5	CS1_6
CS1_1		50%	25%	37.5%	6.25%	2.5%
CS1_2	36%		84%	0%	4%	84%
CS1_3	19%	100%		0%	0%	0%
CS1_4	25%	8.3%	0%		83.3%	0%
CS1_5	5%	0%	0%	100%		0%
CS1_6	100%	0%	0%	25%	0%	

이를 토대로 포함관계(정의 2)를 분석, 2 차 후보 서비스로 식별한 결과는 다음 그림 4 와 같았다. 그림 4 에서 실선은 강결합으로 포함관계를, 점선은 약결합으로 후보서비스들 간에 연관관계가 있음을 의미한다.

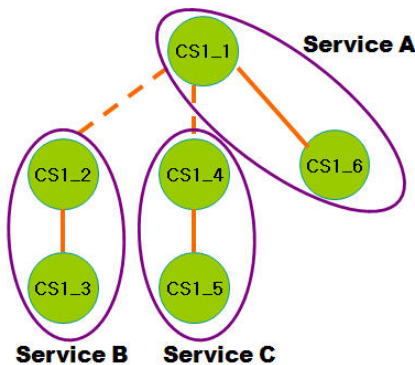


그림 5. 포함관계 분석을 통한 2 차 서비스 식별

본문에서도 말했듯이 서비스지향아키텍처는 서비스간에 약결합(Loosely-Coupled)을 지향하므로, 2 차 후보 서비스간에 발생하는 연관관계에 따른 문제점을 어떻게 해결할 것인가는 최종후보서비스의 질을 결정하는 데 중요한 역할을 한다.

표 2 는 컴포넌트를 그대로 서비스로 식별한 경우와 본 논문에서 제시한 방법으로 서비스를 식별한 경우, 2 차 후보 서비스간에 발생하는 연관관계 위치 및 발생 이벤트 발생 유형에 따른 최초 문제점 수 와 식별

후 문제점 수에 대한 측정 결과이다.

표 2. 연관관계에 따른 문제점 발생 수

연관관계에 따른 문제점 발생 유형		본 논문 제시 방법으로 서비스 식별		컴포넌트의 서비스 식별	
		최초 문제점 수	식별 후 문제점 수	최초 문제점 수	식별 후 문제점 수
한 후보서비스의 Root에서 다른 후보 서비스의 Root를 호출한 경우	Retrieval	20	7	4	1
	Update	7	1	0	0
한 후보서비스의 Root에서 다른 후보 서비스의 Subset을 호출한 경우	Retrieval	6	6	9	9
	Update	1	0	1	1
한 후보서비스의 Subset에서 다른 후보 서비스의 Root를 호출한 경우	Retrieval	0	0	6	6
	Update	0	0	1	1
한 후보서비스의 Subset에서 다른 후보 서비스의 Subset을 호출한 경우	Retrieval	0	0	6	6
	Update	0	0	5	2

그림 6 은 컴포넌트를 그대로 서비스로 식별한 경우와 본 논문에서 제시한 방법으로 서비스를 식별한 경우 각각에 대한 문제점 해결 비율을 그래프로 나타낸 것이다. 기존 컴포넌트를 바로 서비스로 식별하는 경우 단순히 정보를 보여주는 Retrieval 19%, 데이터베이스와 같은 자원에 대한 분할 문제를 가져오는 Update 43%로, 총 18.75%의 문제점만을 해결하는 데 반해, 본 논문에서 제시한 방법으로 서비스를 식별하는 경우는 Retrieval 50%, Update 87.5%로 총 60%의 문제점을 해결한다. 특히 서비스 식별 시 자원의 분할 문제를 일으켜 더 큰 문제점을 야기하는 Update 의 문제점 해결 비율은 본 논문에서 제안한 방법이 무려 40% 이상 높아 효과적이다.

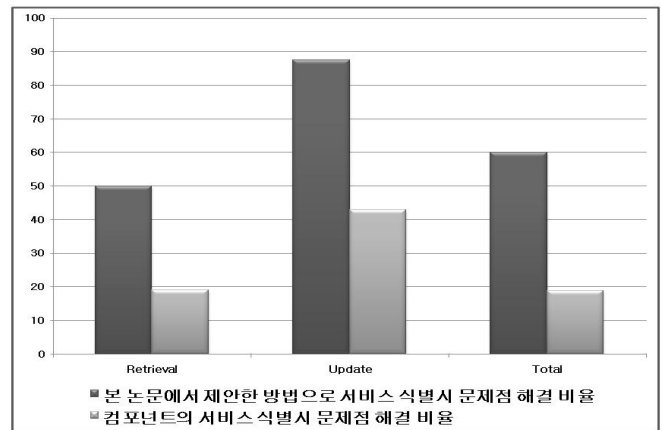


그림 6. 서비스 식별 시 문제점 해결 비율 비교

결과적으로, 본 논문에서 제안한 방법으로 서비스를 식별할 경우 컴포넌트를 바로 서비스로 식별하는 것에 비해 발생하는 문제점을 탁월하게 해결할 수 있고 이를 통해 서비스간 약결합을 실현할 수 있다.

#### 4. 결 론

본 논문에서는 SOA 를 위한 상향식 방법의 서비스 식별법을 제안하였다. GUI 기반 서비스 식별 과정은 후보 서비스 식별을 위한 준비 단계인 전처리 과정과 후보 서비스를 식별하고 정제하는 서비스 식별과정으로 나누어지고, 이를 통해 최종 서비스를 식별한다. 또, 논문에서 제안한 방법을 실제 경영정보시스템에 적용한 사례를 통해 제안한 방법이 서비스지향아키텍처의 서비스간 약결합(Loosely-Coupled)을 실현하는 데 더욱 효과적임을 보였다.

본 논문에서 제안한 방법은 컴포넌트로 구축된 레거시 시스템을 SOA 로 구축하면서 GUI 를 최대한 재활용하기 위한 방법으로 사용될 수 있다.

향후, 비즈니스 단계에서의 서비스 식별 과정 연구를 통해 하향식 서비스 식별법을 제안하고, 궁극적으로 본 논문에서 제시한 상향식 접근 방법과 통합한 절충형(hybrid) 서비스 식별 방법론을 제안하기 위한 연구를 계속 할 것이다.

#### 참고문헌

- [1] Gregg Kreizman, "How to Build a Business Case for Service-Oriented Development of Applications in Government," Gartner. Industry Research, 2005.9
- [2] Kunal Mittal, " Service Oriented Unified Process(SOUP)," IBM Journal, 2005.6
- [3] Ali Arsanjani, "Service-Oriented Modeling and Architecture : How to identify, specify, and realize services for your SOA," IBM developerWorks, 2004.11
- [4] Ash Parikh, Rajesh Pradhan and Nirav Shah, "Modeling of Web Services : A Standards-Based Approach," Software Magazine, 2004.5
- [5] 김유경, 윤홍란, 박재년, "웹서비스 개발 프로세스에서 구현전략 결정을 위한 평가 지침," 정보과학회 논문지: 소프트웨어 및 응용 제 33 권 제 5 호, 2006.5
- [6] Keith Levi, Ali Arsanjani, " A Goal-driven Approach to Enterprise Component Identification and Specification to Enterprise Component Identification and Specification", Communications of The ACM, vol.45, No.10.,2002
- [7] Drik Krafzig, Karl Banke, Dirk Slama, "Enterprise SOA", Pearson Education, Inc.,2005