

# SOC 환경에서의 개발 프로세스 모델 정립을 위한 개별 프로세스의 제시 및 역할 분석

이동훈\*, 백인섭\*

\*아주대학교 정보통신전문대학원

e-mail : [heaven37@gmail.com](mailto:heaven37@gmail.com)

## Presentation and behavior analysis of individual processes for establishing the development process model in the SOC environment

Dong-Hoon Lee\*, In-Sup Paik\*

\*Dept. of Information and Communication, Ajou University

### 요 약

소프트웨어 개발 방법론의 새로운 패러다임으로 제시되고있는 서비스 지향적 컴퓨팅방법론은 소프트웨어의 개발에 서비스라는 요소를 이용하고 있다는 점에서 다른 방법론과는 많은 차이점을 가지고 있다. 기존의 여러 방법론은 그 동안 여러 논문에서 효율적인 개발 프로세스 모델을 제시하고 있지만, 서비스 지향적 컴퓨팅방법론은 아직 개발 프로세스 모델에 대한 연구는 진행중이지 못한 실정이다. 기존의 방법론과 서비스 지향적 컴퓨팅방법론의 여러 차이점 때문에 기존 방법론에 대한 개발 프로세스 모델을 적용하기에는 여러 문제점이 따르며 새로운 개발 프로세스 모델에 대한 연구가 필요하다. 개발 프로세스 모델에 대한 연구를 하기 위해서는 서비스 지향적 컴퓨팅 방법론에서 요구사항분석 및 정의, 설계, 개발, 실행, 유지보수와 같은 각각의 프로세스에 대한 연구가 선행되어야 할 것이다. 이에 본 논문에서는 서비스 지향적 컴퓨팅방법론의 특징을 살펴봄으로써 개발 프로세스 모델에서 사용되어질 프로세스들을 제시해보고자 한다.

### 1. 서론

소프트웨어를 개발하는 방법은 초기의 구조적 프로그래밍 (structured programming) 방법론에서 객체지향 프로그래밍 (object oriented programming), 컴포넌트 기반 소프트웨어 개발 (component based software development) 방법론으로 발전해왔다. 이러한 발전 방향은 소프트웨어의 개발 속도를 높이기 위해 기존에 개발된 소프트웨어를 다시 이용할 수 있도록 재사용성 (reusability) 를 높이고, 사용자 요구사항의 변화에 따른 소프트웨어의 유지보수를 손쉽게 하기 위해 유연성 (flexibility)를 높이는 방향으로 진행되어져 왔다. 하지만 이러한 발전에도 불구하고 기대했었던 재사용성과 유연성을 얻을수 없었으며, 이에 소프트웨어를 하나의 서비스로 이용하자는 개념인 서비스지향적 컴퓨팅 (이하 SOC) 방법론이 나오게 되었다. 기존의 방법론이 객체나 컴포넌트를 이용하여 어떻게 소프트웨

어를 만들수 있는가에 기본을 두고 있다면 SOC 방법론은 소프트웨어의 기능을 어떻게 사용자에게 전달할 것인가에 기본을 두고 있다. 이러한 기본적인 차이점 때문에 기존의 방법론과 SOC 방법론은 많은 차이점을 가지고 있다. [1]

기존의 방법론들은 이미 많은 연구가 진행 되어졌으며 각 방법론에 맞는 개발 프로세스 모델들이 제시 되어져 있다. 예를 들어, 구조적 프로그래밍 방법론에서는 폭포수 (waterfall) 모델이 객체지향 방법론에서는 Rational 사의 RUP (Rational Unified Process) 모델이 대표적으로 사용되어져 왔다. [2][3] 이러한 개발 프로세스 모델들은 각 개발 방법론의 특징에 맞게 만들어졌으며 이를 이용할 경우 보다 안정적이며 쉽게 소프트웨어를 개발할 수 있다. 따라서 SOC 방법론에서도 SOC 환경에 맞는 개발 프로세스 모델이 제시되어져 할 것이다. SOC 방법론은 기존의 방법론과는 소프트

웨어를 바라보는 기본 관점이 다르기 때문에, 기존의 방법론에서 사용되었던 개발 프로세스 모델을 변형, 이용하기 보다는 새로운 개발 프로세스 모델의 제시가 필요하다.

어떠한 방법론에 맞는 개발 프로세스 모델을 만들기 위해서는 먼저 개발 프로세스 모델을 구성하는 각각의 프로세스들에 대한 정의가 선행되어야 한다. 폭소수 모델과 RUP 모델은 각 개발 환경에 맞는 프로세스들로 구성되어져 있으며, 프로세스들은 개발 환경에 맞는 역할들을 수행하도록 정의되어져 있다. 이에 본 논문에서는 SOC 방법론의 특징 및 장점을 살펴보고, SOC 환경에 맞는 개발 프로세스 모델을 작성할 때 사용되어질 수 있는 프로세스들을 정의해 보고자 한다.

## 2. SOC 의 소개 [4]

SOC 는 응용 프로그램 (application) 을 개발하는데 있어서 서비스를 기본 요소로 이용하는 새로운 컴퓨팅 패러다임이다. 여기서 서비스란 어떠한 기능을 수행하는 요소이며, 서비스가 수행하는 기능은 간단한 사용자의 요청에서부터 복잡한 비즈니스 프로세스까지 포함하고 있다. 기존의 방법론이 객체 또는 컴포넌트를 이용해서 응용 프로그램을 개발한다고 한다면, SOC 방법론은 어떠한 개발과정도 필요치 않고 단지 원하는 기능을 제공하는 서비스들을 조합하는 과정만으로 자신이 원하는 프로그램을 완성할 수 있다. 이러한 이유로 SOC 는 기존의 방법론보다 빠르고 저비용으로 응용 프로그램을 완성할 수 있다.

### 2.1 서비스의 특징

SOC 의 기본 요소인 서비스는 다음과 같은 세 가지 특징을 갖는다.

- **기술 중립적 (Technology neutral)**  
서비스는 거의 모든 IT 환경에서 사용 가능한 표준화된 기술을 이용하여 호출되기 때문에 서비스의 내부 구현에서 이용한 특정 기술에는 영향을 받지 않는다.
- **약한 연결 (Loosely coupled)**  
서비스들 사이의 상호작용, 또는 사용자와 서비스 사이의 호출은 서비스의 명세서 (description) 에 따르는 인터페이스 (interface) 통해서 이루어지기 때문에 서비스의 내부적인 구조는 다른 서비스나 사용자에게 영향을 미치지 않는다.
- **위치 투명성 (location transparency)**  
서비스는 자신이 제공하는 기능과 자신을 호출하기 위해 필요한 위치 정보를 UDDI (Universal Description, Discovery, and Integration) 와 같은 등록소에 저장해 둔다. 서비스를 호출하고자 하는 사용자는 등록소를 이용해 원하는 서비스를 찾을 수 있으므로 서비스가 어떠한 곳에 위치하고 있는지는 사용자에게 영향을 미치지 않는다.

서비스의 이러한 특징들은 SOC 환경에서 개발되어진

프로그램들이 재사용성과 유연성이라는 장점을 갖게 한다.

### 2.2 Service Oriented Architecture (이하 SOA)

SOA 는 SOC 환경에서 프로그램을 개발 하는 방법을 논리적 구조로 보여준다. 기존의 방법론과는 다르게 프로그램의 개발에 서비스를 기본 요소로 사용하기 때문에 SOC 환경에서 SOA 는 중요한 의미를 갖는다. SOA 는 그림 1 과 같이 서비스 제공자 (service provider) 서비스 등록소 (service registry) 서비스 사용자 (service client) 로 구성되어져 있으며 이들 사이에는 공표 (publish), 검색 (find), 연결 (bind) 이라는 상호작용이 일어난다.

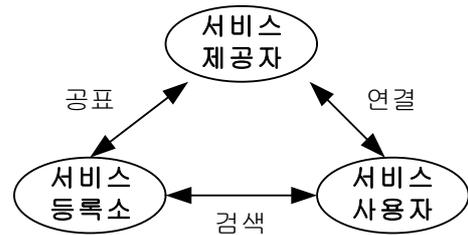


그림 1 Service Oriented Architecture

- **서비스 제공자 (Service Provider)**  
서비스 제공자는 직접 서비스를 개발하거나, 또는 다른 서비스들을 조합하여 추가적인 기능을 갖는 서비스를 만들어낼 수 있다. 서비스 제공자는 자신이 제공하는 서비스의 명세서를 서비스 등록소에 공표하여 서비스 사용자가 자신을 찾을 수 있도록 해야한다. 서비스 제공자가 공표하는 명세서는 서비스가 제공하는 기능 (function), 성능 (capability), 품질 (quality) 뿐만 아니라 자신을 호출하는 방법을 알려주는 인터페이스 (interface) 등으로 구성되어져 있다.
- **서비스 등록소 (Service Registry)**  
서비스 등록소는 서비스 제공자가 공표한 명세서를 저장하고 있음으로써 서비스 사용자가 자신이 필요로 하는 서비스를 검색할 수 있도록 해준다. 서비스 등록소를 이용함으로써 서비스 제공자는 모든 서비스 사용자에게 자신의 명세서를 공표할 필요가 없으며, 서비스 사용자는 모든 서비스 제공자에게 문의할 필요없이 자신이 원하는 서비스를 검색할 수 있다.
- **서비스 사용자 (Service Client)**  
서비스 사용자는 자신이 원하는 기능을 제공하는 서비스를 서비스 등록소에서 검색할 수 있다. 여러 서비스들 중에서 자신이 원하는 성능과 품질을 갖는 서비스를 선택 한 후에는 해당 서비스 제공자와 연결을 맺고, 자신이 원하는 시점에서 요청을 보냄으로써 서비스의 기능을 이용할 수 있다.

사실 SOA 가 SOC 환경에서 일어나는 모든 일들을 나타내고 있지는 않다. 여기에는 서비스 제공자와 서비스 사용자 사이에서 일어날 수 있는 협의 (negotiate), 여러 서비스를 조합함으로써 추가적인 기능을 제공하는 서비스로 만들어내는 조합 (composite), 서비스 제공자와 서비스 사용자의 연결을 풀기 위한 연결 해제 (unbind), 여러 서비스들 중에서 자신에게 적합한 서비스를 찾기위한 서비스 선택 (select) 와 같은 작용들까지는 표현하고 있지 않다. 하지만 이와 같은 작용이 일어나기 위한 배경에는 SOA 가 기본적인 역할을 하고 있기 때문에 SOC 환경에서 SOA 는 중요한 의미를 갖는다고 할 수 있다.

**3. SOC 환경에서의 프로세스 제시 및 역할 분석**

지금까지는 SOC 방법론의 특징 및 장점과 함께 기본적인 SOA 에 대해서 알아보았다. 본 절에서는 이를 기반으로 SOC 방법론에서 필요한 프로세스들을 제시해 보고, 각 프로세스의 역할에 대해서 분석해 보고자 한다.

프로세스의 역할은 SOC 방법론이 프로그램의 개발에 서비스를 기본 요소로 사용하고 있다는 점 때문에 서비스를 제공하는 서비스 시장 (service market) 에 많은 영향을 받을 수 밖에 없다. [5] 따라서 서비스 시장의 성숙도에 따라 프로세스의 역할은 달라질 수 있다. 서비스 시장의 성숙도란 사용자가 원하는 기능을 제공하는 서비스들이 서비스 등록소에 공표되어지는 정도로 판단할 수 있다. 사용자가 원하는 기능은 매우 다양하다. 이러한 기능은 일반적으로 사용되는 기능일 수도 있으며, 매우 특별한 경우에만 필요로 되는 기능일 수도 있다. 사용자가 원하는 기능들에 대해서 이를 제공하는 서비스가 다수 존재한다면 성숙한 서비스 시장이지만, 해당 기능을 제공하는 서비스가 하나 또는 전무 할 때 이는 미성숙한 서비스 시장이다. 현재 SOC 방법론은 연구가 진행중인 상태이기 때문에 서비스 시장은 미성숙 상태이며, 이에 프로세스의 역할을 분석할 때에는 이러한 점을 감안해야 할 것이다.

**3.1 요구사항 분석 및 정의 (Requirement Analysis and Definition) 프로세스**

기존의 개발 환경에서 요구사항 분석 및 정의 프로세스는 사용자의 요구사항을 분석하고 정의하는 역할을 한다. SOC 환경에서의 요구사항 분석 및 정의 프로세스는 이러한 역할 뿐만 아니라 요구사항을 만족시키는 후보 서비스들을 검색해서 적절한 서비스를 선택하고 이를 종합하는 역할까지 수행해야 한다. 이러한 역할은 세 가지 경우로 나누어질 수 있다. 첫 번째는 서비스 시장이 성숙한 경우이며, 두 번째와 세 번째는 서비스 시장이 미성숙한 경우이다.

1. **요구사항을 만족시키는 모든 서비스가 있는 상태**  
이러한 상태는 매우 이상적인 상태로써 비용, 품질, 보안과 같은 비기술적 요구사항 (non-technical) 부분을 토대로 알맞은 서비스를 선택할

수 있다. 이 후 선택되어진 서비스들을 조합 (composite) 해보고, 조합해본 결과가 원래 의도대로 요구사항을 만족시키는지 여부를 확인해야 한다. (그림 2-a)

2. **요구사항을 완전히 만족시키지는 않지만 유사한 기능을 제공하는 서비스가 있는 상태**  
이러한 경우 두 가지 역할이 필요하다. 첫 번째는 유사한 기능을 제공하는 서비스 공급자와의 협상 (negotiate) 를 통해 변형된 (customized) 서비스를 공급받는 것이다. 이러한 과정에서 서비스 공급자는 변형된 서비스를 다시 서비스 등록소에 공표함으로써 서비스 시장을 보다 성숙하게 만들 수 있다. (그림 2-b)  
두 번째는 요구사항 자체를 변경하는 것이다. 서비스 제공자와의 협상이 이루어지지 않았으며, 해당 부분이 프로그램의 주요 부분이 아니라면 요구사항 자체를 변경함으로써 자체적인 개발의 필요없이 보다 쉽게 프로그램을 개발할 수 있다.
3. **요구사항을 만족시키는 서비스가 없는 경우**  
이러한 경우 SOC 이전의 개발 방법론을 이용해서 설계 및 개발이 이루어져야 한다. SOC 방법론의 여러 장점을 잃어버릴 수 있지만 서비스 시장이 완전한 성숙 단계에 이르기 전까지는 이러한 경우가 많이 발생하게 된다. 서비스 사용자는 자체 개발한 서비스를 서비스 등록소에 공표함으로써 이를 필요로 하는 사용자에게 공급할 수 있으며, 이러한 경우 서비스 공급자 역할을 하게 된다. (그림 3-c)

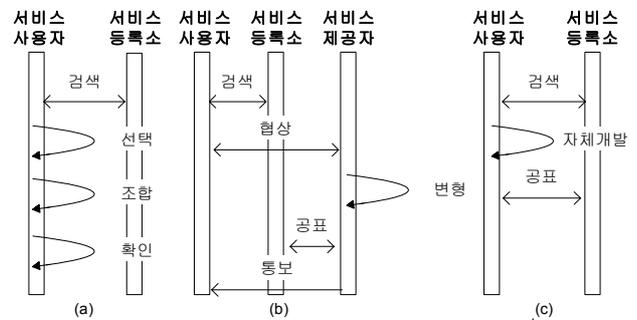


그림 2 요구사항 분석 및 정의 프로세스의 역할 (a - 성숙한 서비스 시장 b, c - 미성숙한 서비스 시장)

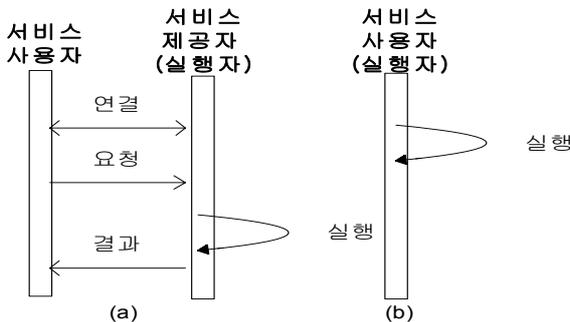
**3.2 설계 (design) 프로세스와 개발 (development) 프로세스**

SOC 환경에서의 설계 프로세스와 개발 프로세스의 역할은 기존의 개발 환경에서 같은 역할을 한다. SOC 환경에서의 설계, 개발 프로세스는 세 가지 경우에 실행되어질 수 있다. 첫 번째 경우는 서비스 제공자가 사용자들이 필요로 할 것이라고 예상하는 서비스를 개발하고자 하는 경우로써 가장 보편적으로 사용되어지는 경우이다. 두 번째 경우는 그림 2-b 에서와 같이 서비스 사용자로부터 현 서비스에 대한 변형이 요구되어지는 경우이다. 서비스 사용자와 제공자 사이에서 현 서비스의 변형에 대한 협상이 이루어진 경우,

서비스 제공자는 사용자의 요구사항에 맞도록 서비스를 변형하기 위해 설계 프로세스와 개발 프로세스를 이용하게 된다. 세 번째 경우는 그림 3 - c 에서와 같이 원하는 서비스를 제공하는 서비스 제공자가 없을 경우로써 서비스 사용자가 SOC 이전의 개발 방법론을 이용해서 직접 설계, 개발해야 경우이다.

**3.3 실행 (execution) 프로세스**

실행 프로세스에서는 사용자의 요청에 의해 서비스가 실행 (execute) 되어진다. 서비스의 실행을 담당하는 측은 서비스 실행자 (service executor) 라고 불리게 된다. 서비스 제공자 또는 서비스 사용자는 경우에 따라 서비스 실행자가 될 수 있다. 실행 프로세스의 역할은 요구사항 분석 및 정의 프로세스의 결과에 따라 달라지게된다. 즉 그림 2 - (a), (b) 와 같이 사용자가 서비스 제공자의 서비스를 이용하는 경우 사용자와 제공자 사이에는 그림 3 - (a) 와 같이 연결 (bind), 요청 (request), 실행 (execute), 결과 전송과 같은 과정이 일어나게 되며, 이때 서비스 실행자는 서비스 제공자가 된다. 또한 그림 2 - (c) 와 같이 사용자가 원하는 서비스를 찾기 못해 직접 설계, 개발 과정을 거친 경우 그림 3 - (b) 와 같이 서비스 사용자가 스스로 실행하게 되며 이 때 서비스 실행자는 서비스 사용자 본인 이 된다.



**그림 3 실행 프로세스의 역할**  
 (a - 서비스 제공자의 서비스를 이용하는 경우  
 b - 자체 개발한 서비스를 이용하는 경우)

**3.4 유지보수 (Maintenance) 프로세스**

기존의 방법론에서 유지보수 프로세스는 가장 많은 비용과 노력, 시간을 필요로 하는 프로세스 였다. 하지만 재사용성과 유연성이 높은 SOC 환경에서의 유지보수 프로세스는 간단한 역할만을 담당하게 된다. 유지보수 프로세스의 역할은 두 가지 경우로 나누어진다. [5]

● **사용자의 요구사항이 변경된 경우**

사용자는 현재 프로그램에 새로운 기능의 추가 또는 이전 기능의 삭제등을 원할 수 있다. 새로운 기능의 추가는 이를 만족시키는 서비스를 찾기위해 요구사항 분석 및 정의 프로세스의 호출을 필요로 하며, 이전 기능의 삭제는 그림 3 - (a) 에서와 같은 사용자와 제공자 사이의 연결을

해제하기 위한 연결 해제 (unbind) 과정이 필요하다.

● **하위 서비스의 결합으로 인한 통합 서비스의 오류**

SOC 환경에서는 하나의 프로그램이 여러 서비스의 통합으로 이루어져 있기 때문에 한 서비스의 예기치 못한 오류는 전체 프로그램의 오류를 일으키게 된다. 이러한 경우 어떠한 서비스가 오류를 발생시켰는지를 알아내는 과정이 필요하다. 오류를 발생하는 서비스를 찾은 후에는 앞서세와 같이 해당 서비스에 대한 연결을 해제하고, 요구사항 분석 및 정의 프로세스를 이용해서 같은 기능을 제공하는 서비스를 다시 찾아야 한다.

**4. 결론 및 향후 연구과제**

본 논문에서는 SOC 환경의 개발 프로세스 모델에서 사용되어 질 수 있는 프로세스들을 제시해 보았으며, 해당 프로세스들이 수행해야 하는 역할과 프로세스들 사이의 상호작용을 SOC 환경에서 일어날 수 있는 상황에 따라 분석해 보았다. 본 논문에서는 요구사항 분석 및 정의, 설계, 개발, 실행, 유지보수라는 다섯 개의 프로세스를 제시해 보았다. 요구사항 분석 및 정의 프로세스는 서비스 시장의 성숙도에 따라 그 역할이 달라지게 되며, 이는 다시 실행 프로세스의 역할에 영향을 미치게 된다. 설계, 개발 프로세스는 기존의 개발 방법론에서와 같은 역할을 담당하지만 SOC 라는 상황에 따라 기존의 방법론에서와는 다른 시점에서 사용되어진다. 또한 유지보수 프로세스의 역할은 SOC 의 재사용성고 유연성이라는 장점으로 그 역할을 간편화 되었다.

본 논문에서 제시한 프로세스들을 이용하여 SOC 방법론을 위한 개발 프로세스 모델을 작성하기 위해서는 SOC 환경에서 일어날 수 있는 여러 시나리오에 본 논문에서 제시한 프로세스들을 적용시켜 보는 과정이 필요하며, 이러한 과정의 결과로써 보편적으로 이용가능한 개발 프로세스 모델이 제시되어질 수 있다.

**참고문헌**

[1] Turner, "Turning Software into a Service", computer volume: 36, 10, Oct. 2003  
 [2] Rogers S. Pressman "Software Engineering - Practitioner's approach", McGRAW-HILL International Edition, 1997  
 [3] Rational Software Corporation Whitepaper, "Rational Unified Process - Best Practices for Software Development Teams"  
 [4] Mike P. Papazoglou, "Service-Oriented Computing: Concepts, Characteristics and Directions", WISE, 10-12 Dec. 2003  
 [5] Nicolas Gold and Andrew Mohan, "Understanding Service-Oriented Software", software, IEEE, 2, March-April 2004  
 [6] IEEE, "Guide to Software Engineering Body of Knowledge", 2004