

# 상황 인지 기반 서비스 지향 적응형 시스템 개발 프레임워크

(A Framework for Developing Service-Oriented Adaptive System based on Context Awareness)

유 찬 우 <sup>†</sup> (Chanwoo Yoo)	심 재 근 <sup>†</sup> (Jaekun Shim)
한 종 대 <sup>†</sup> (Jongdae Han)	박 영 기 <sup>†</sup> (Youngki Park)
정 우 성 <sup>†</sup> (Woosung Jung)	김 희 천 <sup>**</sup> (Heechern Kim)
이 병 정 <sup>***</sup> (Byungjeong Lee)	우 치 수 <sup>****</sup> (Chisu Wu)

**요 약** 유비쿼터스 시대가 도래함에 따라 서비스 지향 적응형 시스템의 중요성이 커지고 있다. 서비스 지향 적응형 시스템의 개발 단계에 있어, 개발자들은 상황이나 서비

· 이 논문은 2009 한국컴퓨터종합학술대회에서 '맥락 인식 기반 서비스 지향 적응형 시스템 개발 프레임워크'의 제목으로 발표된 논문을 확장한 것임

<sup>†</sup> 학생회원 : 서울대학교 컴퓨터공학부  
chanwoo.yoo@gmail.com  
jkshim@selab.snu.ac.kr  
elvenwhite@selab.snu.ac.kr  
ykpark@selab.snu.ac.kr  
wsjung@selab.snu.ac.kr

<sup>\*\*</sup> 정 회원 : 한국방송통신대학교 컴퓨터학과 교수  
hckim@knou.ac.kr

<sup>\*\*\*</sup> 종신회원 : 서울시립대학교 컴퓨터과학부 교수  
bjlee@uos.ac.kr  
(Corresponding author인)

<sup>\*\*\*\*</sup> 종신회원 : 서울대학교 컴퓨터공학부 교수  
wuchisu@selab.snu.ac.kr

논문접수 : 2009년 8월 13일  
심사완료 : 2009년 9월 21일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제10호(2009.10)

스 탐색과 관련한 문제들에 부딪히게 된다. 지금까지 이러한 문제를 해결하기 위한 연구들이 개별적으로 수행되어 왔지만, SOA를 적응형 시스템에 도입하여 통합적으로 문제를 해결하는 연구는 부족한 실정이다. 본 연구는 서비스 지향 적응형 시스템 개발에 관련된 문제들을 해결하는 통합 프레임워크를 제안한다. 먼저 프레임워크의 프로세스를 정의하고 각 프로세스 요소 별로 지원 방법과 산출물을 설명한다. 그리고 시스템의 개발 단계에서 구체적인 서비스의 사례를 통해 본 프레임워크의 프로세스가 어떻게 적용될 수 있는지 살펴본다.

**키워드** : 서비스, 적응, 상황, 프레임워크

**Abstract** As Ubiquitous era comes, the importance of service-oriented adaptive software increases. However, there are many issues in development phases of service-oriented adaptive software, like contexts and service search. Though there are many researches which manage these issues separately, still we need more integrated and mature framework for solving these issues. So in this paper, we propose an integrated framework for development of service-oriented adaptive software. Our framework defines processes and work products for development. We also perform a case study which our framework is applied to.

**Key words** : Service, Adaptation, Context, Framework

## 1. 서론

유비쿼터스 시대가 도래함에 따라, 상황에 맞게 반응할 수 있는 적응형 소프트웨어의 중요성이 커지고 있다. 또한 적응형 소프트웨어의 동적인 특성을 잘 살릴 수 있는 개발 방법으로 SOA(Service-Oriented Architecture)를 적용한 서비스 지향 개발 방법론이 각광을 받고 있다. SOA는 서비스, 동적 발견(dynamic discovery), 메시지 등의 핵심 컴포넌트로 이루어진 분산 시스템 모델을 말한다[1]. 서비스 지향 적응형 시스템이 환경에 동적으로 적응하기 위해서는 환경 상황에 맞는 최적의 서비스를 탐색 후 적용하는 것이 중요하다.

서비스 지향 적응형 시스템의 개발 과정에서, 개발자들은 상황이나 서비스 탐색과 관련된 여러 가지 문제들을 다루게 되고, 따라서 이에 대한 지원을 필요로 한다. 예를 들어, 설계 단계에서는 상황 정보를 반영하기 위한 표현 수단 및 모델링 방법이 필요하다. 또한 작성된 모델링 정보와 적응 규칙을 검증하기 위해서 모델 정보를 검사하는 수단 역시 필요할 것이다. 그리고 구현 단계에서는 상황이라는 관심(concern) 요소를 반영하여 시스템을 구현해야 하기 때문에 어떻게 상황을 반영할지에 대한 방법론이 필요하다. 마지막으로 시스템의 유지보수 단계에서는 시스템이 상황에 따라 적응할 수 있도록 서비스를 효과적으로 탐색하는 메커니즘이 필요하다. 이와 같은 문

제들을 개별적으로 다루는 연구들은 많이 수행되어 왔지만 본 연구처럼 SOA를 적응형 시스템에 도입하여 이러한 문제들을 통합적으로 다루는 연구는 부족한 실정이다.

본 연구는 서비스 지향 적응형 시스템 개발을 지원하기 위한 통합 프레임워크인 COFSAS(A Context-based development Framework for Service-oriented Adaptive Software)를 제안한다. 본 프레임워크는 통합 개발 프로세스, 그리고 각 프로세스의 지원 방법 및 산출물에 대한 내용을 포함하고 있다. 또한 주식 증가 예측 서비스를 사용하는 서비스 지향 적응형 시스템의 개발 과정에서, 제안한 프레임워크가 어떻게 적용될 수 있는지 사례연구를 통해 살펴본다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 본 연구와 관련되어 수행되었던 다른 연구들을 소개한다. 3장에서 COFSAS프레임워크를 제안하고 프로세스를 설명한 뒤, 4장에서 프레임워크의 적용 사례 연구를 보인다. 마지막으로 5장에서 결론을 맺는다.

## 2. 관련 연구

서비스 지향 아키텍처의 모델링에 있어서 사실상의 표준은 아직 정해지지 않은 상태이다. IBM과 같은 곳에서 UML을 이용하여 모델링 하려는 시도[2]가 있고, 그 외에 SOMF[3]나 SOMA[4]와 같은 연구가 진행된 바 있다. 한편, 센서와 같은 환경 요소들에 대한 모델링은 아직 소프트웨어적인 고려보다는 하드웨어의 전기적 특성에 따른 모델링 위주로 진행되었다[5]. 소프트웨어적으로 상황을 모델링한 연구로는 홈 네트워크 도메인에 초점을 맞춰서 서비스 지향 적응형 시스템의 모델에 대한 연구가 진행된 바 있다[6].

적응형 소프트웨어를 위한 아키텍처 및 프레임워크를 제시하는 연구들[7-9]이 수행되었지만 이 연구들은, 추상적인 모델 수준에서의 기술에 머물거나 부분적인 구현 상의 도움을 주는 데 그쳤다는 한계를 가진다. 환경에 따라 행동이 달라지는 적응형 소프트웨어를 기술하기 위한 방법인 COP(Context Oriented Programming) 관련 연구[10,11]들도 수행되었다.

시맨틱(semantic) 정보를 이용하여 웹 서비스를 검색하는 연구는 많이 진행되었으며, 이러한 연구들을 실제로 실험해 볼 수 있는 OWL-S/UDDI Matchmaker 등과 같은 도구들도 개발되었다[12]. 그러나 대부분의 범용적인 웹 서비스 검색 엔진들은 비용 측면에서 실용적이지 않기 때문에 시맨틱 정보를 활용하지 않는다는 연구 결과가 있다[13]. 이에 적은 시맨틱 정보를 사용하지만 웹 서비스를 효과적으로 탐색하는 연구도 진행되었다[14]. 이 연구는 저렴한 비용으로 서비스 검색 환경에 바로 적용될 수 있는 장점이 있다.

## 3. COFSAS 프레임워크

### 3.1 개요 및 프로세스

그림 1은 본 연구에서 제안하는 COFSAS 프레임워크의 개요를 나타내고 있다. COFSAS는 좌측의 각 개발 프로세스 과정과 관련한 우측 원안의 지원 방법들로 구성되어 있다. 각각의 개발 프로세스 과정에서의 문제들과 프레임워크가 제시하는 지원 방법은 표 1과 같다.

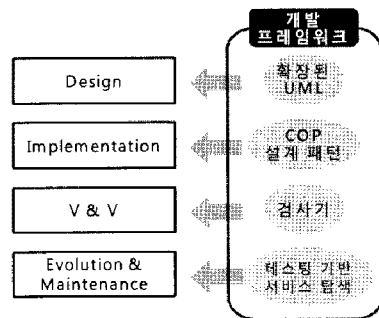


그림 1 COFSAS 개요

표 1 개발과정에서 발생하는 문제와 지원 방법

설계 과정	문제	환경 상황이나 하위 시스템들이 연결되는 상황 정보를 표현하는 것이 필요.
구현 과정	문제	상황 정보를 잘 반영한 시스템을 구현.
	지원	COP 설계 패턴을 제시. 상황 정보 모델로부터 해당 COP 설계 패턴이 적용된 템플릿 코드를 생성.
V&V 과정	문제	상황 정보 모델과 적응 규칙 검증 필요.
	지원	상황 정보 모델과 적응 규칙을 확인할 수 있게 해주는 검사기(checker)를 제공.
진화 및 유지 보수 과정	문제	상황에 따라 서비스들을 탐색하여 최적의 것을 선택하는 것이 중요.
	지원	시맨틱 웹 서비스 탐색 방법 중, 상황 정보 모델을 입력으로 하는 테스트 기반 서비스 탐색 방법을 제시.

그림 2는 프레임워크의 프로세스와 각 지원 방법들의 관계를 보여준다. 사각형은 프레임워크가 제공하는 지원 방법 또는 지원 도구를 뜻하고, 꺾인 사각형은 그와 관련된 산출물이다.

그림 2의 프로세스와 지원 방법간의 관계를 설명하면 다음과 같다. 먼저 상황 정보를 확장된 UML을 통해 모델링하고 이 과정에서 산출물로 상황 정보 모델링 문서를 얻는다. 이 산출물은 다른 3개의 지원 방법의 입력 정보로 사용된다. 먼저 상황 정보 모델링 문서는 그 자신의 모델 정보를 검사하기 위한 검사기(checker)의 입력으로 사용된다. 검사기는 상황 정보 모델링 문서에서 상황 정보 모델 정보를 추출한 XML 문서와 개발자가

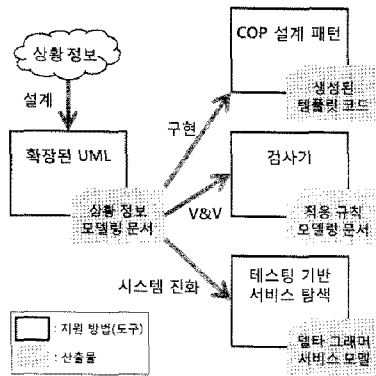


그림 2 프로세스 및 관계도

작성한 적응 규칙 모델의 XML 문서를 입력으로 받는다. 검사기는 이 XML 문서들의 내용을 개발자가 쉽게 검사하도록 보여준다. 따라서 검사기의 입력으로 사용되는 적응 규칙 모델링 문서가 이 과정의 산출물로 나온다. 다음으로 상황 정보 모델링 문서는 COP 설계 패턴이 적용된 템플릿 코드를 얻기 위한 지원 방법의 입력으로 사용된다. 상황 정보 모델링 문서로부터 상황 정보를 추출하여 COP 설계 패턴으로 뽑아내고, 이 패턴에 따라 템플릿 코드를 생성한다. 이 과정을 거치면 상황이 COP 설계 패턴으로 반영된 템플릿 소스 코드를 얻는다. 마지막으로 상황 정보 모델링 문서는 동적 블랙박스 테스트(Dynamic Black-box Testing) 서비스 탐색을 하기 위한 입력 정보로 사용된다. 상황 정보 모델로부터 서비스 탐색을 하기 위한 IOPEs(Inputs, Outputs, Preconditions, Effects) 정보를 추출하고, 이를 델타 그래머(delta-grammar)[15] 그래프로 바꾼다. 이 델타 그래머 그래프를 가지고 동적 블랙박스 테스트 서비스 탐색 방법을 거치면 원하는 서비스를 찾는다. 이 과정에서 델타 그래머 그래프로 기술된 서비스 모델이 산출물로서 나온다.

### 3.2 상황 정보 모델링

본 연구에 앞서 서비스 지향 적응형 시스템 중 홈 네트워크 도메인에 관련하여 상황 정보 모델링 연구가 선행된 바 있다[6]. 해당 선행 연구에서 확장한 UML의 메타모델과 그에 따른 표현법을 사용하여 상황을 모델링 한다. 다만 선행 연구가 홈 네트워크 도메인에 한정했던 것에 비해 그 적용 범위를 일반적인 서비스 지향 적응형 서비스로 확대 하였다. 이 과정을 통해 개발자는 상황 정보 모델의 UML 문서를 산출물로 가진다. 이 상황 정보 모델의 UML 문서는 다른 지원 방법의 입력으로 사용되기 때문에 상황 정보 모델을 올바르게 설계 하는 것은 중요하다.

### 3.3 상황과 적응 규칙 모델 검사기

앞서 작성된 상황 정보 모델링 산출물은 프레임워크의 다른 지원 방법의 입력으로 사용되기 때문에 정확하게 기술될 필요가 있다. 하지만 이를 사용자가 확인하고

그림 3 상황 및 적응 규칙 모델 검사기

수정하는 절차는 시간이 오래 걸리고 귀찮은 작업이다. 본 연구는 손쉽게 상황 정보 모델링 정보와 적응 규칙 모델을 사용자가 빠르게 확인할 수 있도록 검사기를 구현했다. 검사기의 모습은 그림 3과 같다.

모델 검사기는 홈 네트워크 도메인에 초점을 맞춘 선행 연구[6]에서 제안한 검사기를 개선한 것으로, 좀더 빠르게 모델 정보와 적응 규칙을 확인할 수 있게 하였다. 사용자가 'STOP' 버튼을 누른 뒤, 검사기의 동작을 멈추고 환경 변수 값을 변화시켜서 즉각적으로 상황 적응을 확인할 수 있다. 입력 정보의 형태는 선행 연구[6]와 같이 XML 문서의 형태이기 때문에 UML로 기술된 상황 정보 모델링 문서로부터 XML 문서를 작성하는 것이 필요하고 적응 규칙도 XML 문서로 작성한다. 검사기의 입력으로 사용될 XML 문서의 스키마는 다른 선행 연구에서 UML을 확장하여 기술한 XML 스키마를 따른다[16].

### 3.4 COP 설계 패턴

본 연구는 확장된 UML로 기술된 상황 정보 모델링 산출물을 바탕으로 목표 시스템의 코드 일부를 생성해 내는 방법을 제시한다. 기존 연구에서는 COP를 달성하기 위해 MOP(Meta Object Protocol)[17]을 사용하였는데, MOP를 지원하지 않는 언어에서는 COP를 사용하는 것이 어렵다. 이에 MOP를 이용하여 객체 지향 시스템 자체를 수정하는 방식이 아니라, 기존 객체 지향 시스템을 이용하여 COP를 표현하는 설계 패턴을 제시한다. 설계 패턴을 UML로 표현하기 위해 우선 UML 메타모델을 확장하였고, 확장된 UML 메타모델과 설계 패턴의 모습은 그림 4와 같다.

우선 상황과 클래스, 메소드 사이의 관계는 그림 4와 같이 모델링 된다. 상황은 클래스를 포함하는 도메인으로, 클래스를 포함하는 클래스로 구현된다. 즉, 상황 역시 클래스이다. 그렇기 때문에 UML 메타모델에서 스테레오타입 'Context'는 메타클래스 'Class'를 상속받고, 메타클래스 'Class'를 속성(attribute)으로 갖는 관계를 가진다.

앞에서 산출물로 나온 상황 정보 모델링 문서로부터 상황과 서비스, 서비스의 기능 정보를 추출하여 COP 설

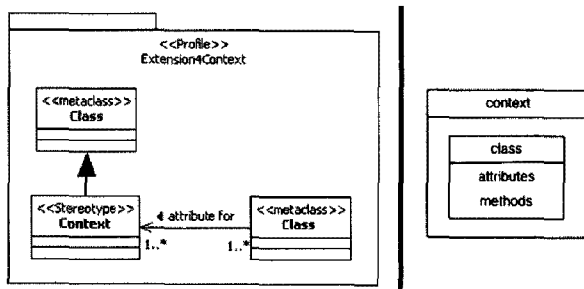


그림 4 COP 설계 패턴과 패턴 표현을 위해 확장된 UML 메타모델

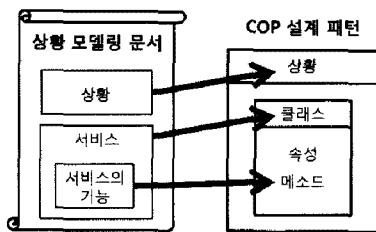


그림 5 상황 정보 모델링 문서로부터 패턴 정보 추출

계 패턴이 반영된 소스 코드를 일정 부분 생성하는 것이 가능하다. 코드 생성은 두 단계로 이루어지는데 첫 번째 단계에서는 상황 정보 모델링 문서로부터 상황과 클래스, 메소드 이름들을 그림 5와 같이 사상(mapping) 후 추출하여 XML 파일로 생성한다. 클래스는 상황에 따라 선택되는 서비스들을 나타내고, 메소드는 서비스의 기능을 나타낸다. 두 번째 단계에서는 추출한 XML 파일을 바탕으로 가능한 정보 안에서 템플릿 코드를 생성해낸다.

### 3.5 테스트 기반 서비스 탐색 방법

본 연구에 앞서 시맨틱 정보를 기반으로 동적 블랙박스 테스트 기반 웹 서비스 검색 프레임워크에 대해 선행 연구를 진행한 바 있다[14]. 이 서비스 탐색 방법이 웹 서비스 요청자에게 요구하는 시맨틱 정보는 IOPEs이다. 입력(Inputs)과 출력(Outputs)은 웹 서비스가 수행되기 위해 필요한 값과 웹 서비스의 결과물을 나타내는 것이며, 전제 조건(Preconditions)과 효과(Effects)는 웹 서비스가 수행된 이후의 내부 상태 변화를 나타낸다. IOPEs를 표현하는 방법은 다양한데, 그 중에서 델타 그래머를 사용한다. 따라서 상황 정보 모델링 문서로부터 이 델타 그래머 그래프를 추출한 뒤 서비스 탐색을 수행한다.

상황 정보 모델링 문서에는 서비스의 IOPEs 정보중에 P와 E에 대한 정보는 가지고 있지 않다. 따라서 상황 정보 모델링 문서에 나타나 있는 입력과 출력 정보를 델타 그래머의 입력과 출력으로 사상하고 입력의 제약 조건으로 작용할 수 있는 상황 정보는 델타 그래머의 위쪽으로 넣어 주어서 그래프를 작성한다.

## 4. 사례 연구

주식 증가 예측 서비스를 사용하는 서비스 지향 적응형 시스템을 상정하고, 해당 서비스와 관련된 상황 정보를 제안한 프레임워크에 적용시켜 본다. 주식 증가 예측 서비스는 주식 회사의 종목 번호를 입력으로 받아서 당시의 KOSPI 지수의 등락률을 상황으로 삼아 당일의 주식 증가를 추산하여 반환하는 서비스이다. 서비스의 이름은 'Stock Price Prediction'이고 등락률에 따른 상황은 상승장인 'Bull Market'과 하락장인 'Bear Market', 중간 정도를 'Normal Market'이 있다. 각각의 상황은 증가를 추산하는데 필요한 'predictionFactor'의 값이 다르다.

먼저 서비스와 상황에 대해 확장한 UML로 상황 정보를 모델링 하였다. 상황 정보 모델링 문서의 모습은 그림 6과 같다. 서비스와 상황간의 관계에 따라 상황 서비스들이 기술되어 있다. 상황이 상승장의 경우와 하락장인 경우, 보통 장의 경우에 각각의 상황 서비스들은 'BullPrediction', 'BearPrediction', 그리고 'NormalPrediction'이다. 상황을 판단하는 근거가 상황 서비스에 노트로 달려서 표현된다. 예를 들어 'Stock Price Prediction' 서비스는 KOSPI 지수가 5%이상 올랐을 때 'BullPrediction' 상황 서비스가 된다. 또한 'predictionFactor'의 값이 상황에 따라 값이 각각 다르게 설정되어 있다.

이렇게 작성된 상황 정보 모델링 문서는 검사를 통해 그 내용을 검사한다. 상황을 결정하는 'KOSPI rate' 값을 변화 시키면서 서비스가 올바르게 상황에 따라 적용하는지 살펴볼 수 있다.

상황 정보 모델링 문서로부터 상황, 서비스, 서비스 기능을 추출하여 XML 파일을 생성한 뒤, 제안한 COP 설계 패턴을 적용하여 템플릿 코드를 생성한다. 생성된 템플릿 코드는 서비스와 메소드들의 바디는 구현되어 있지 않고, 상황과 서비스, 서비스의 기능만을 반영한다.

상황 정보 모델링 문서로부터 입출력 정보를 추출하고, 입출력에 관련한 제약조건들을 추가하여 델타 그래

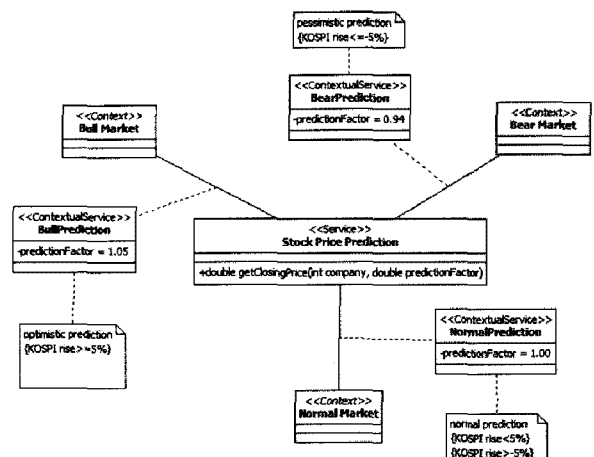
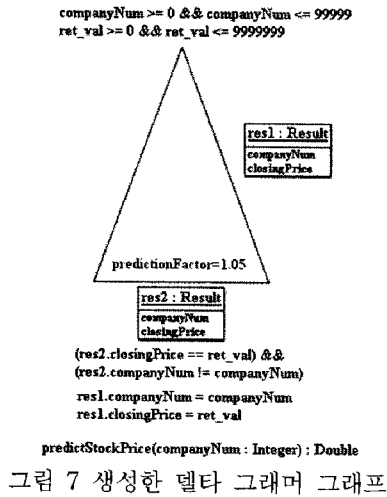


그림 6 주식 증가 예측 서비스의 상황 정보 모델



며 그래프를 도시한다. 그림 7은 앞서 기술한 상황 정보 모델링 문서에서 입출력 정보를 추출하고 제약 조건을 추가하여 그린 델타 그래머 그래프이다. 위쪽에 입력 값인 'companyNum'과 서비스의 테스트 결과로 반환되는 값인 'ret\_val'의 제약 조건이 명시된다. 가장 아래쪽에 서비스의 명세가 적혀있다.

'predictStockPrice'라는 서비스는 입력 값으로 정수형인 'companyNum'을 갖고, 출력 값은 더블형(double type)이라는 것을 나타낸다. 가운데 텍스트는 입력 값과 출력 값에 대해 가능하지 않은 조합을 보여준다. 즉, 종목 번호가 다른데 예측 증가가 같게 나오는 경우는 사실상 거의 없다. 이런 서비스는 항상 같은 값을 반환하는 것과 같이 비정상적으로 동작한다고 가정할 수 있다. 따라서 이런 서비스를 탐색했을 때 올바르지 않은 서비스로 판정하고 무시하는 내용이 담겨 있다.

### 5. 결론

지금까지 서비스 지향 적응형 시스템의 개발에 있어서 상황과 적응에 관련한 문제들을 통합하여 다루는 COFSAS 프레임워크를 제시하였다. 본 연구에서 제안한 상황 정보 모델링 방법과 검사를 통해 상황과 서비스를 비교적 쉽게 기술하고 시험해 볼 수 있을 것이다. 또한, 상황 정보 모델링 문서로부터 코드 생성과 서비스 탐색을 지원하는 방법이 해당 시스템의 개발과 시스템 유지 보수성의 향상에 도움이 될 것으로 보인다. 마지막으로, 각 산출물 간의 사상 관계를 도구로 지원하여 산출물을 자동 생성하는 연구를 추가로 진행할 경우, 프레임워크의 실용성을 더욱 높일 수 있을 것으로 기대된다.

### 참고 문헌

[1] J. McGovern, S. Tyagi, M. Stevens, S. Mathew, Java Web Services Architecture, Morgan Kaufmann, chapter 2, pp.35-41, 2003.

[2] <http://www.ibm.com/developerworks/library/ar-logsoa/>

[3] M. Bell, "Service-Oriented Modeling: Service Analysis, Design, and Architecture," Wiley & Sons, 2008.

[4] N. Bieberstein, R. Laird, K. Jones, T. Mitra, "Executing SOA: A Practical Guide for the Service-Oriented Architect," IBM Press books, 2008.

[5] C. Ortiz, J. Puiq, C. Palau, M. Esteve, "3D Wireless Sensor Network Modeling and Simulation," International Conference on Sensor Technologies and Applications, pp.307-312, Oct. 2007.

[6] J. Han, J. Shim, B. Lee, H. Lee, C. Wu, "An Integrating Modeling Technique for Service-oriented Adaptive System," Proc. of the KCSE, vol.11, no.1, pp.166-174, 2009. (in Korean)

[7] J. Georgas, A. Hoek, R. Taylor, "Architectural Runtime Configuration Management in Support of Dependable Self-Adaptive Software," ACM SIGSOFT Software Engineering Notes, vol.30, no.4, Jul. 2005.

[8] J. Cobleigh, L. Osterweil, A. Wise, B. Lerner, "Containment Units: A Hierarchically Composable Architecture for Adaptive Systems," Proc. of the 10th ACM SIGSOFT symposium on Foundations of software engineering, 2002.

[9] M. Kumar, B. Shirazi, S. Das, B. Sung, D. Levine, M. Singhal, "PICO: A Middleware Framework for Pervasive Computing," IEEE Pervasive Computing, vol.2, no.3, 2003.

[10] P. Costanza, R. Hirschfeld, "Language Constructs for Context-oriented Programming - An Overview of ContextL," Dynamic Languages Symposium, co-located with OOPSLA'05, San Diego, California, USA, Oct. 18, 2005.

[11] R. Hirschfeld, P. Costanza, O. Nierstrasz, "Context-oriented Programming," Journal of Object Technology, vol.7, no.3, pp.125-151, Mar.-Apr., 2008.

[12] J. Cardoso, "Semantic Web Services, theory, tools, and applications," Information Science Reference, pp.240-280, 2007.

[13] H. Lausen, N. Steinmetz, "Survey of Current Means to Discover Web Services," STI Technical Report, 2008.

[14] Y. Park, W. Jung, B. Lee, C. Wu, "Automatic Discovery of Web Services Based on Dynamic Black-Box Testing," Proc. of the 33rd Annual IEEE International Computer Software and Applications Conference, 2009.

[15] S. Kaplan, S. Goering, R. Campbell, "Specifying Concurrent Systems with  $\Delta$ -grammars," Proc. of the 5th International Workshop on Software Specification and Design, pp.20-27, 1989.

[16] J. Shim, B. Lee, C. Wu, "Modeling of Simulation Environments for Service-oriented Adaptive Systems with ECA Rules and Extended UML," Proc. of the KIISE Fall Conference, vol.35, no.2(B), pp.33-37, 2008. (in Korean)

[17] G. Kiczales, J. D. Rivieres, D. G. Bobrow, The Art of the Metaobject Protocol, The MIT Press, 1991.