

# MVC 아키텍처 기반 애플리케이션의 GUI 테스트<sup>†</sup>

## (GUI Testing for MVC Architecture based Applications)

주희주<sup>‡</sup>  
(Heeju Joo)

이찬근<sup>¶</sup>  
(Changun Lee)

**요약** MVC는 유지보수가 쉬워 효율적인 개발이 가능한 아키텍처이다. 웹과 PC환경의 애플리케이션뿐만 아니라 모바일 애플리케이션 개발에 많이 적용됨에 따라 그에 대한 테스트 또한 중요하다. 기존에 MVC 아키텍처를 적용한 개발에 초점을 둔 연구는 많았으나 MVC 아키텍처의 특성을 고려한 효율적인 테스트를 위한 연구는 많지 않았다. 따라서 본 논문에서는 MVC 아키텍처 기반 애플리케이션의 효율적이고 정확한 테스트를 위한 연구를 진행한다. 모델-뷰-컨트롤러의 융합된 상태를 담았던 기존의 상태 다이어그램을 뷰 상태와 모델-컨트롤러 상태 다이어그램으로 분리하고 테스트 케이스를 작성한 후, 테스트를 진행한다.

**키워드** MVC 아키텍처, 상태 다이어그램, 상태 전이, 명세 기반 테스트, GUI 테스트, 테스트 케이스

**Abstract** MVC(Model-View-Controller) architecture is well-known for high maintainability and it makes efficient development possible. Recently, it has been applied in many fields such as web, desktop and mobile applications. Therefore, effective testing for this architecture is strongly needed. Although much research has been done, there was not much efforts for exploiting MVC architecture in GUI testing. Therefore, in this paper, we propose a scheme for efficient and accurate GUI testing for MVC based applications. In this study we separate original state diagrams into view state diagram and model-controller state diagram. Then, we present a case study showing the effectiveness of our proposed scheme.

**Key words** MVC architecture, state diagram, state transition, specification-based testing, GUI testing, test cases

## 1. 서론

MVC(Model-View-Controller) 아키텍처는 UI와 프로그램 로직을 분리하여 유지보수가 쉬운 애플리케이션을 효율적으로 개발할 수 있기 때문에 로컬 애플리케이션과 웹 애플리케이션의 개발에 매우 널리 사용되고 있다. 최근 스마트 폰 시장이

발달됨에 따라 모바일 애플리케이션의 개발에도 그 적용 영역이 넓어지고 있다.

MVC를 이루고 있는 모델-뷰-컨트롤러 중 특히 GUI를 제공하는 뷰는 사용자와 밀접한 관계를 형성하는데, 이 때문에 더욱 뷰를 통한 테스트가 중요하게 여겨진다. 단순히 GUI만을 살펴보는 테스트가 아닌 프로그램 내부의 모델 및 컨트롤러의 상호작용의 테스트이기 때문에 테스트 케이스 생성에 신중해야 한다. 그러나 현재 MVC 아키텍처를 적용하는 개발 연구는 활발히 진행되는 반면 MVC 아키텍처 기반 애플리케이션에 특화된 효율적인 테스트 방법에 대한 연구는 미미하다. 따라서 본 논문에서는 MVC 아키텍처로

<sup>†</sup> 본 연구는 한국연구재단 기초연구사업(과제번호 20110013924)의 지원을 받았습니다.

<sup>‡</sup> 학생회원 : 중앙대학교 컴퓨터공학과  
soulmateof88@gmail.com

<sup>¶</sup> 종신회원 : 중앙대학교 컴퓨터공학부 교수, 교신저자  
cglee@cau.ac.kr

논문접수 : 2011년 01월 07일

심사완료 : 2011년 02월 28일

개발된 애플리케이션의 상태 다이어그램을 뷰와 모델-컨트롤러의 두 가지로 분리하여 작성하고 테스트 케이스를 생성함으로써 좀 더 정확하게 명세 기반의 상태 전이 테스트를 수행하는 방법을 제안한다.

2장에서 관련 연구를 통해 MVC 아키텍처에 대한 개념과 이를 적용한 개발 및 테스트 연구 현황에 대해 살펴 볼 것이다. 3장에서는 상태 다이어그램의 분리 및 테스트 케이스 작성에 대해 제안한다. 4장에서는 MVC 아키텍처 기반의 모바일 애플리케이션으로 사례연구를 제시하고 이를 뷰와 상태 다이어그램으로 표현 한 후, 테스트 케이스를 생성하여 테스트를 수행한다. 마지막으로, 제안하는 테스트 방법의 차별성 및 향후 연구 방향을 검토하며 논문을 결론지을 것이다.

## 2. 관련 연구

### 2.1 MVC 아키텍처

MVC아키텍처는 1970년대 객체지향 프로그래밍의 개념 위에 설계된 언어인 Smalltalk를 이용한 애플리케이션 개발에 적용하기 위하여 도입되었다[3]. MVC 아키텍처는 모델, 뷰, 컨트롤러의 세가지 구성요소가 애플리케이션 내부에서 독립적인 역할을 하며 작업 요청을 통해 상호 작용을 한다. 다음 그림 1은 MVC 아키텍처의 구성 및 관계를 보여준다.

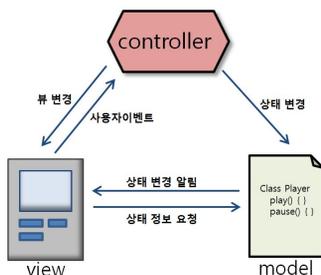


그림 1 MVC 아키텍처[3]

모델은 모든 데이터와 상태 및 로직을 처리한다. 뷰와 컨트롤러에서 데이터 상태를 조작하거나 가져오기 위한 인터페이스를 제공하며, 뷰와 컨트롤러에게 직접 관여하지 않고 독립적으로 존재한다. 뷰는 일반적으로 화면을 표현하는데 필요한 모든 처리를 담당하는데, 필요한 상태 및 데이터는 모델에서 가져온다. 컨트롤러는 사용자로부터 입력을 받아서 그것이 모델에게 어떤 의미가 있는지 파악한다. 컨트롤러는 모델에게 상태 변경을 요청하거나 뷰의 상태 변경을 요청할 수 있다. 사용자는 뷰를 통해서만 애플리케이션과 접촉이 가능하며 뷰는 모델에게 표시해야 할 상태를 요청한다. 모델의 상태가 변경되면, 모델은 뷰에게 상태가 변경되었음을 알린다.

### 2.2 상태 다이어그램

상태 다이어그램은 이벤트에 따른 객체의 상태 변화를 나타내는 다이어그램이다. 상태 다이어그램을 구성하는 요소로는 시작상태(Initial Pseudo State), 종료상태(Final State), 상태(State) 및 전이(Transition)가 있으며 상태를 변화시킬 수 있는 것은 전이이다. 이런 요소들을 조합하여 하나의 시스템 전체, 시스템의 일부 또는 개별 객체에 대한 동작을 나타내어 개발자와 유저 모두에게 유용한 정보를 제공할 수 있다. 본 논문에서는 상태 다이어그램을 이용하여 애플리케이션을 표현하고, 이를 테스트에 활용한다.

### 2.3 MVC 아키텍처 개발 및 테스트

MVC아키텍처를 적용한 애플리케이션의 개발 방법론에 대해서 많은 연구가 있었다. 웹 플랫폼 서비스 개발에 최적화 되어있는 아키텍처인 만큼 웹 분야에서는 이미 MVC 아키텍처를 이용한 연구가 활발하게 진행되고 있다[2][4]. 또한 EC system, stopwatch, command processing framework 등 다양한 서비스 개발 및 연구에 적용되었으며[1][8], 상태

다이아그램을 뷰와 모델-컨트롤러의 두 가지로 나누어 효율적인 개발에 접근한 연구[5]도 있다. 최근 스마트 폰 애플리케이션의 효율적인 개발을 위해 적극 적용하려는 연구[9]도 진행 중이다. 그러나 활발한 개발 연구와는 달리 MVC 아키텍처 기반의 애플리케이션을 위한 별도의 테스트 기법에 관한 연구는 비교적 활발하지 못했다. MVC 아키텍처 기반의 웹 애플리케이션을 위한 테스트에 관한 연구에는 추상 workflow 그래프 모델을 제안하여 state-based/code-based 테스트를 수행한 연구[7]와, 모델-뷰-컨트롤러 각각을 다른 방법으로 테스트하고 각각 다른 접근 제어 방법으로 보안성을 높인 연구[10]가 있다.

### 3. MVC 애플리케이션의 테스트 방법 제안

#### 3.1 상태 다이어그램 분리

MVC 애플리케이션의 실행 중에 상태를 파악하는 것은 쉽지 않으나 코드를 분석하여 상태를 도출하는 것은 비교적 쉽게 수행할 수 있다[5]. 따라서 본 논문에서는 애플리케이션의 코드를 잘 알고 있는 개발자가 상태 다이어그램을 작성하고 분리한다고 가정한다. 상태 다이어그램 작성 시 상태 및 전이를 정확히 작성하여 테스트 과정에서 놓치는 오류가 없도록 한다.

#### 3.2 분리된 테스트 케이스 작성

분리하여 작성된 상태 다이어그램을 기반으로 테스트 케이스를 작성하는 과정은 비교적 간단하다. 각각의 '상태'가 Pre\_Condition, Post\_Condition이 되며, '전이'는 Transition으로 매칭 된다. 모든 경우를 조합하여 테스트 케이스를 작성한다. 다음 그림 2는 상태 전이와 테스트 케이스 간 매핑의 예를 보여준다.

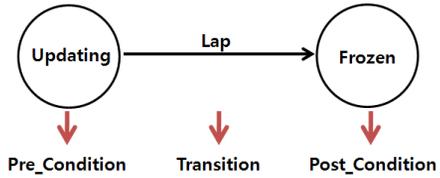


그림 2 상태 전이와 테스트 케이스 항목의 매핑

## 4. MVC 애플리케이션의 테스트 방법 제안

### 4.1 Case Study

본 논문에서는 MVC 아키텍처로 개발된 애플리케이션의 테스트 과정을 보이기 위해 예제 프로그램을 이용한다. 예제 프로그램은 안드로이드 OS에서 음악을 재생하는 'SongPlayer' 애플리케이션이며 다음 그림 3과 같은 UI를 제공한다.



그림 3 'SongPlayer'의 UI

프로그램 내부의 모델-뷰-컨트롤러 역할 및 상호연결은 다음과 같다.

- 뷰: 'SongPlayer' 사용자에게 보여지는 모든 부분을 담당한다. 시간을 체크하는 Timer와 TimeLine, 노래 제목이 나타나는 부분, 그리고 정지/재생/일시 정지 버튼이 뷰에 해당한다.
- 컨트롤러: 사용자 이벤트를 입력 받아 모델에게 상태 변경을 요청한다. 예를 들어, 사용자가 '정지'버튼을 눌렀을 경우 이벤트를 받아 모델에게 전달하여 재생 중인 노래를 멈추도록 요청한다.

- 모델: 실제 'SongPlayer'의 모든 로직을 담당한다. 컨트롤러로부터 상태 변경을 요청 받고, 변경된 상태를 뷰에게 전달한다. 재생 중인 노래를 일시 정지 했을 때, 그 상태를 뷰에게 전달하여 화면에 '일시 정지' 버튼 대신 '재생'버튼이 보이도록 한다.

### 4.2 상태 다이어그램 생성

예제 프로그램의 상태 다이어그램은 그림 4와 같다.

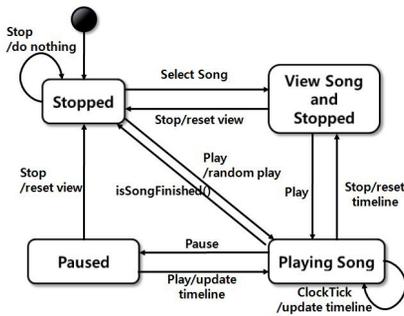


그림 4 'SongPlayer'의 상태 다이어그램

상태 다이어그램으로 예제 프로그램의 상태와 전이에 대해 한눈에 볼 수 있다. 애플리케이션의 상태는 크게 'Stopped(정지)', 'View Song and Stopped(노래 제목은 보이지만 플레이어 정지 상태)', 'Playing Song(재생)', 'Paused(일시 정지)' 4가지로 나타내어지며 각 상태로의 전이는 그림 4에 제시되어 있다.

본 논문에서 제시하는 MVC의 효율적인 테스트를 위해 모델-컨트롤러 간의 상태와 뷰의 상태 두 가지로 분리하여 그린 상태 다이어그램은 다음 그림 5, 그림 6과 같다.

그림 5에서와 같이, 'SongPlayer'의 모델-컨트롤러 간의 상태 다이어그램은 음악을 재생하는 Player 부분과 타이머 뷰의 상태를 변화시키는 Timer 부분의 동시 상태로 표현된다. 사용자 눈에 직접 보이지 않지만, 내부적으로 프로그램의 상태를 변화 시키는 로직의 상태를 볼 수 있다.

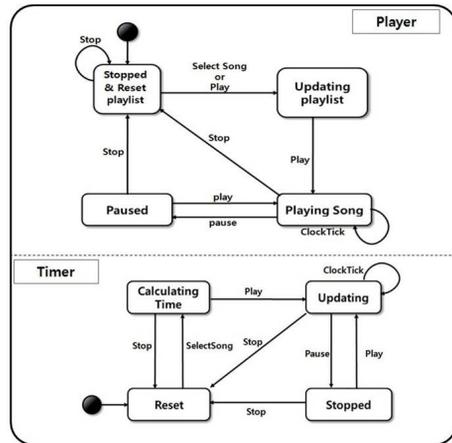


그림 5 'SongPlayer'의 모델-컨트롤러 상태 다이어그램

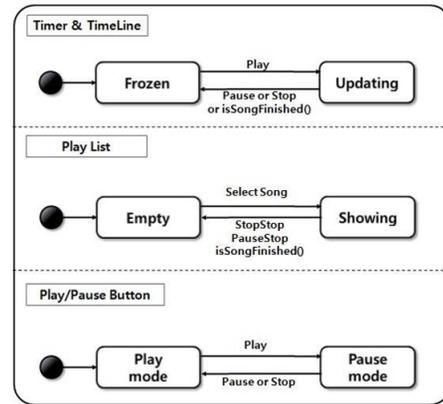


그림 6 'SongPlayer'의 뷰 상태 다이어그램

그림 6은 'SongPlayer'의 뷰 상태 변화를 보여 준다. 뷰는 노래 시간을 나타내는 Timer & TimeLine 부분과 노래 제목을 나타내는 Play List 그리고 Play/Pause Button의 동시 상태로 나누어진다. 내부 로직을 알 필요가 없이, 사용자 눈에 보이는 뷰의 상태 변화만을 분리해 나타내었다. 이제 상태 다이어그램을 두 가지로 분리하였다. 뷰 상태 다이어그램은 유저 GUI 테스트를 위한 테스트 케이스 생성에 이용되며, 내부 로직 테스트를 위한 테스트 케이스는 모델-컨트롤러 상태 다이어그램으로부터 생성 된다.

### 4.3 분리된 테스트 케이스

상태 다이어그램을 분리함으로써 뷰만의 상태를 좀 더 자세히 볼 수 있게 되었다. 이는 MVC 아키텍처 기반 애플리케이션의 GUI 테스트를 위한 중요한 시나리오가 된다. 뷰 상태 다이어그램으로부터 도출한 테스트 케이스의 일부는 다음 표 1과 같다.

표 1 뷰 상태 테스트 케이스

No.	Pre_Condition	Transition	Post_Condition	P/F
1	Frozen	Play	T_Upd	
2	Frozen	Play	TL_Upd	
3	Updating	Pause	T_Stp	
4	Updating	Stop	T_Stp	
5	Updating	isSongFinished	T_Stp	
...	...	...	...	...
15	Pause_md	Stop	Play_md	

모델-컨트롤러 상태 다이어그램은 내부 로직 개발자 및 테스터를 위한 시나리오를 제공하며, 테스터는 뷰를 통해 사용자 입력 값을 컨트롤러와 모델에게 전달한다. 모델-컨트롤러간의 상태 다이어그램으로부터 도출한 테스트 케이스는 다음 표 2와 같다.

표 2 모델-컨트롤러 상태 테스트 케이스

No.	Pre_Condition	Transition	Post_Condition	P/F
1	Stopped	Stop	DoNth	
2	Stopped	Select Song	Upd_PL	
3	Stopped	Play	Upd_PL	
4	Upd_PL	Play	Playing	
5	Playing	Pause	Paused	
...	...	...	...	...
15	t_Cal	Play	t_Upd_ing	
16	t_Cal	Stop	t_reset	
17	t_Upd_ing	ClockTick	t_Upd_ing	

### 4.4 테스트 결과

테스트를 위해 개발자와 베타 테스터에게 테스트 케이스를 제공하였다. 뷰 테스트의 결과는 테스터의 판단을 통해 성공/실패 여부를 결정하였으며, 모델-컨트롤러 테스트의 결과는 테스트를 위해 코드를 약간 수정하여 로그가 출력되게 하였고 로그 출력 결과를 통해 테스트의 성공/실패 여부를 결정하였다.

결과의 비교를 위해 분리되지 않은 상태로 수행된 테스트(테스트 1)와 분리된 상태로 수행된 테스트(테스트 2)를 진행하였으며 두 테스트는 동일한 환경에서 진행되었다. 테스트 1은 케이스가 많지 않아 테스트 시간이 비교적 적었으나 미처 발견하지 못한 오류가 비교적 많았다. 다음 그림 7은 두 테스트에 대한 수행 결과 비교 그래프이며, 이를 바탕으로 표 3에서 두 가지 테스트의 장단점을 비교해 보았다.

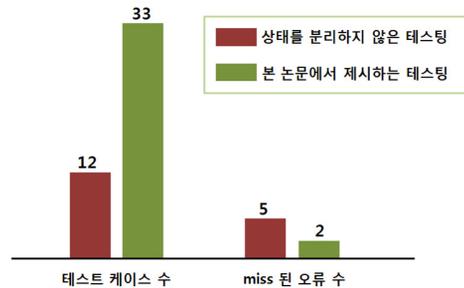


그림 7 테스트 수행 결과 비교

표 3 제안하는 테스트와의 비교

	상태를 분리하지 않은 테스트	본 논문에서 제안하는 테스트
장점	<ul style="list-style-type: none"> <li>· 테스트에 드는 비용(시간)이 짧다.</li> <li>· 모델/뷰/컨트롤러의 상태를 동시에 테스트 할 수 있다.</li> </ul>	<ul style="list-style-type: none"> <li>· 테스트 Fail시 수정 할 부분을 빠르게 인지할 수 있다.</li> <li>· 뷰 테스트의 진행이 쉽다.</li> </ul>
단점	<ul style="list-style-type: none"> <li>· 세밀한 테스트 케이스 생성이 어렵다.</li> <li>· 테스트 Fail시 수정 할 부분을 인지하는 데에 비용(시간, 노력)이 많이 필요하다.</li> </ul>	<ul style="list-style-type: none"> <li>· 테스트에 드는 비용(시간)이 길다.</li> <li>· 모델/컨트롤러의 테스트 준비 작업을 위해 기존 소스 코드를 수정해야 한다.</li> </ul>

## 5. 결론

본 논문에서는 MVC 아키텍처 기반 애플리케이션을 효율적으로 테스트 하기 위해, 상태 다이어그램을 분리하여 분리된 테스트 케이스를 도출하는 테스트 방법을 제안하였다. MVC 아키텍처는 모델-뷰-컨트롤러의 세 가지 구성요소가 애플리케이션 내부에서 독립적인 역할로 상호작용하는 구조를 가지며, UI와 프로그램 로직을 분리하여 유지보수가 쉬운 애플리케이션을 효율적으로 개발할 수 있기 때문에 로컬 애플리케이션과 웹 애플리케이션의 개발에 매우 널리 사용되고 있다.

본 논문에서 제안하는 테스트 방법은 상태 다이어그램을 통해 테스트 케이스를 도출한 명세 기반의 GUI 테스트로, 상태 다이어그램을 분리하기 전 보다 더 세밀한 테스트를 진행할 수 있다. 특히 뷰 상태만을 테스트 하거나 모델-컨트롤러만을 테스트 할 때 유용하여 MVC 아키텍처 기반의 애플리케이션 테스트를 효율적으로 수행할 수 있다. 또한, 상태 다이어그램을 기반으로 하기 때문에 특정 개발 환경에 영향을 받지 않는 테스트가 가능하다. 그러나 상태 다이어그램을 분리하고 그로부터 테스트 케이스를 작성하는 과정에서 시간이 많이 소요된다는 등의 단점을 가진다.

향후 연구 방향으로는 본 논문의 MVC 아키텍처 GUI 테스트를 Model-Based GUI Testing에 관한 연구[6]로 확장하여 조금 더 체계적인 뷰/모델-컨트롤러 테스트 모델을 설계한다. 또한, 테스트 케이스 생성을 자동화하는 방법에 대해 제안하고 구현하여 테스트 비용을 감소시키는 연구를 진행할 것이다.

## 참고 문헌

- [1] 서순모, 양해술, “MVC Architecture 기반의 EC System용 상품전시 컴포넌트의 설계 및 구현,” 한국정보과학회 2001년도 가을 학술 발표논문집 제28권 제2호(1), 2001.
- [2] T. Bodhuin et al., “Migrating COBOL Systems to the WEB by using the MVC Design Pattern,” In Proc. of Working Conference on Reverse Engineering, 2002.
- [3] E. Freeman and E. Freeman, Head First Design Patterns, O Reilly, 2004.
- [4] X. Qiu, “Building Desktop Applications with Web Services in a Message-based MVC Paradigm,” In Proc. of IEEE International Conference on Web Services, 2004.
- [5] S. Hansen and T. V. Fossum, “Refactoring Model-View-Controller,” Journal of Computing Sciences in Colleges, Vol. 21, Issue. 1, 2005.
- [6] A. Kervinen et al., “Model-Based Testing Throu-gh a GUI,” In Proc. of International Workshop on Formal Approaches to Testing of Software, 2006.
- [7] M. Karam et al., “An Abstract Model for Testing MVC and Workflow Based Web Applications,” In Proc. of Advanced Int’l Conference on Telecommunications and Internet and Int’l Conference on Web Applications, 2006.
- [8] A. Naderlinger and J. Templ, “A framework for command processing in Java/Swing programs based on the MVC Pattern,” In Proc. of 6th international symposium on Principles and practice of programming in Java, 2008.

- [9] 이호중 외 2명 “서비스 기반 모바일 애플리케이션의 MVC 아키텍처 및 적용 사례연구,” 정보과학회논문지 : 컴퓨팅의 실제 및 레터 제16권 제11호, pp.1111-1115, 2010.
- [10] Q. Wu et al., “Unit Testing and Action-Level Security Solution of Struts Web Applications Based on MVC,” In Proc. of International Conference on Biomedical Engineering and Computer Science, 2010.
- 2007~현재 중앙대학교 컴퓨터공학부 조교수.  
 <관심분야> 실시간 소프트웨어, 수행시간 모니터링, 소프트웨어 테스트

## 저 자 소 개



**주 희 주**

2011년 중앙대학교 컴퓨터공학부 학사.  
 2011년~현재 중앙대학교 컴퓨터공학과 석사과정.  
 <관심분야> 실시간 소프트웨어, 소프트웨어 테스트.



**이 찬 근**

1996년 중앙대학교 전자계산학과 학사  
 1998년 KAIST 전산학과 석사.  
 2005년 Univ. of Texas at Austin 전산학과 박사.  
 2005년~2007년 미국 인텔 소프트웨어 엔지니어.



# 소프트웨어 재사용성 증대를 위한 재사용 관련 지식의 표현, 관리 및 공유 방법<sup>†</sup>

## (Representation, Management and Sharing of Reuse-related Knowledge for Improving Software Reusability)

구형민<sup>\*</sup>      고인영<sup>†</sup>  
(Hyung-Min Koo)      (In-Young Ko)

**요약** 소프트웨어 재사용이란 새로운 소프트웨어를 개발할 때 처음부터 모든 것을 새로 개발하는 것이 아니라 기존에 개발된 재사용 가능한 자산을 활용하여 개발하는 개념이다. 기존 자산들은 다른 개발자에 의해 다른 목적으로 개발되었으므로 개발자들은 이러한 자산을 올바르게 재사용함에 있어서 어려움에 직면할 수 있다. 이러한 어려움에 직면했을 때 개발자들은 도움을 얻기 위하여 그 자산들과 관련된 적절한 지식이나 과거 유사한 어려움을 겪었던 개발자로부터의 정보를 얻고자 하는 경향이 있다. 따라서 개발 도메인 전문가나 다른 개발자의 지난 경험은 소프트웨어 자산 재사용 시의 참고자료로서 중요한 역할을 할 수 있다. 재사용 관련 지식은 같거나 유사한 문제나 어려움을 해결함에 있어서 시간과 노력을 줄일 수 있게 도와주고, 이미 여러 번의 재사용으로 검증된 자산을 이용할 수 있게 해 줌으로써 재사용의 위험요소 또한 줄일 수 있다. 본 논문에서는 재사용 관련 지식의 표현 모델을 제시하고, 지식의 협력적인 공유 및 증식을 용이하게 하기 위한 소프트웨어 재사용 위키의 아키텍처 및 프로토타입을 설명한다. 또한 재사용 관련 지식 모델과 그에 따른 관리 방법의 효과를 검증하기 위해 재사용 문제 해결에 관한 검증 계획을 설명한다. 본 연구를 통해 개발자들의 참여를 유도하여 재사용 관련 지식의 효과적인 공유 및 증식이 가능하고, 정형화되고 체계적인 저장 및 관리를 통해 재사용 문제 해결에 도움을 줄 수 있으리라 기대한다.

**키워드** 소프트웨어 재사용, 소프트웨어 재사용 관련 지식, Web 2.0 기반 소프트웨어공학

**Abstract** Software reuse the concept of developing software by using existing software assets, rather than developing it from scratch. Developers may face difficulties of reusing existing software assets because existing assets are normally developed by other developers for different purposes. Developers tend to seek appropriate knowledge about effectively reusing software assets from the developers who have faced and solved similar problems in reusing software assets previously. In other words, the reuse-related knowledge of domain experts or other developers usually provides important clues to solve reuse-related problems. Such reuse-related knowledge can help developers to reduce the time and effort to identify and solve the difficulties and problems that may arise in reusing software assets and in minimizing the risks of reusing them by allowing them to reuse reliable software assets in an appropriate way and by recognizing similar requirements or constraints of reusing the assets. In this paper, we describe a model to represent reuse-related knowledge in a formal way, and explain the architecture and a prototype implementation of Software Reuse Wiki (SRW) that enables collaborative organization and sharing of software reuse-related knowledge. We have conducted an experiment pertaining to problem solving in reusing assets based on reuse-related knowledge. We also discuss about our evaluation plan for showing the benefits and contributions of reuse knowledge representation model and management methods in SRW. We expect that SRW can contribute to facilitate users' participations and make efficient sharing and growing of reuse-related knowledge. In addition, the representation model of reuse-related knowledge and management methods can make developers acquire more reliable and useful reuse-related knowledge in a straightforward manner without spending additional efforts to find solutions to solve reuse-related problems.

**Key words** Software reuse, Software Reuse-related Knowledge, Web 2.0-based Software Engineering

## 1. 서론

소프트웨어 재사용은 새로운 소프트웨어를 개발할 시, 처음부터 새로 개발하는 것이 아니라 기존에 개발되어 있는 재사용 자산을 활용하여 개발하고자 하는 개념이다[1]. 이미 개발된 자산을 재활용함으로써 소프트웨어의 생산성과 품질을 향상시키고, 개발 시간과 비용을 감소할 수 있다[2]. 기존 자산들은 다른 개발자에 의해 다른 목적으로 개발되었으므로 개발자들은 이러한 자산을 올바르게 재사용함에 있어서 어려움에 직면할 수 있다. 이러한 어려움에 직면했을 때, 개발자들은 도움을 얻기 위하여 그 자산들과 관련된 적절한 지식이나 과거 유사한 어려움을 겪었던 개발자로부터의 정보를 얻고자 하는 경향이 있다[3]. 따라서 개발 도메인 전문가나 다른 개발자의 지난 경험은 소프트웨어 자산 재사용에 참조자료로써 중요한 역할을 할 수 있다.

소프트웨어 재사용 관련 지식은 “재사용과 관련된 개인이나 기관의 경험, 혹은 다른 개발자에게 도움을 줄 수 있는 유용한 히스토리 정보”라고 정의할 수 있다. 그러나 재사용 관련 지식을 제공하기 위해서 풀어야 할 문제점들이 아래와 같이 존재한다[3, 4, 5, 6].

- 비정형화된 재사용 관련 지식: 현재의 (재사용) 지식은 주로 자연어를 통해 개발자의 의도와 성향에 따라 기술되는 것이 보통이다. 이러한 구조 없이 비정형화된 지식들은 개발자들에게 유용하고 적절한 지식을 검색 및 추천되어 지기 힘들다.
- 모호성: 재사용 관련 지식은 주로 개발자들의 의도와

그들의 생각을 중심으로 기술되기 마련이다. 따라서 다른 개발자들이 이러한 지식을 보았을 때 잘못 이해하거나 오해할 여지가 있다. 또한, 다른 개발자가 이해할 만큼 충분히 기술되지 못한 지식들은 이해하기 모호한 경우가 많다.

- 부분적 지식: 재사용 관련 지식을 제공함에 있어서 가장 중요한 점은 지식이 활용될 수 있을 만큼 충분히 완전하게 기술되어야 한다는 것이다. 물론 처음부터 완벽한 지식을 제공하기는 힘들지만, 여러 사람들이 협력적으로 지식을 공유하면서 정제해 나가면 재사용 관련 지식들은 점차 충분한 것이 되어 갈 수 있을 것이다. 그러나 개발자들은 이러한 지식의 정제에 참여하는 활동 자체를 귀찮고 추가적인 일로 여길 수 있는 문제점이 있다.

재사용 관련 지식의 제공은 지식들이 많은 사람들에게 의해 공유되고 협력적으로 증식 및 정제 되었을 때 그 의미가 있다. 지식의 증식 및 정제는 지식의 유용함을 점차 발전시켜 나가는 반복적인 행위로 볼 수 있다. 이러한 재사용 관련 지식 증식 및 정제를 위해 본 연구에서는 아래와 같이 꼭 제공해야 할 요구사항들을 정리하였다. 이 요구사항들은 국방도메인의 네 군데 기관을 방문하여 7차례 설문조사 및 인터뷰를 수행한 결과와 관련연구를 바탕으로 한다[3, 4, 5, 6, 7].

- 지식의 정형화 및 구조화: 재사용할 개발자들의 추가적인 노력을 야기 시키는 모호함과 오해의 문제점을 해결하기 위해 정형화된 지식의 구조를 제공해야 한다. 정형화된 지식의 표현은 개발자들이 지식을 획득할 시, 보다 이해하기 쉬운 방법을 제공한다. 또한 정형화된 지식을 활용하여 재사용 지원 도구는 적절한 지식 추론 및 개발자에게의 추천을 해 줄 수도 있다. 정형화된 지식 제공을 위해서는 기술되어야 할 중요한 요소들로 구성된 재사용 표현 모델이 필요하다.
- 협력적인 지식의 조직: 재사용 관련 지식은 앞서 설명한 바와 같이 많은 사람들이 협력적으로 공유 및 정제 했을 때, 그 유용성이 더 커진다. 협력적 공유 및 정제를 할 수 있도록 지원하기 위해서는 지식 기술을 위해 개발자들이 작성해야 할 명확하고 사용하기 쉬운 인터페이스가 필요하다. 명확한 지식 기술 인터페이스는 모호함과 오해의 문제 해결에 도움을 줄 수 있다.

† 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다. (2010-SW-12-DM-01)

‡ 학생회원 : 한국과학기술원 전산학과  
hmkoo@Kaist.ac.kr

§ 종신회원 : 한국과학기술원 전산학과 부교수  
iko@Kaist.ac.kr

논문접수 : 2011년 01월 17일

심사완료 : 2011년 02월 22일

- 반복적인 지식의 정제: 지식은 처음부터 잘 정제되어 있는 것이 아니라 여러 사람들에 의해 반복적으로 정제되어 나갈 때 그 유용성이 커진다. 따라서 지식을 협력적이고 반복적으로 공유, 증식 및 정제를 할 수 있도록 지원해 주는 방법이 필요하다. 이러한 방법 및 도구는 개발자들이 부가적이고 부담이 되는 활동으로 여기지 않도록 사용하기 쉬운 방법 또한 제공해야 한다.

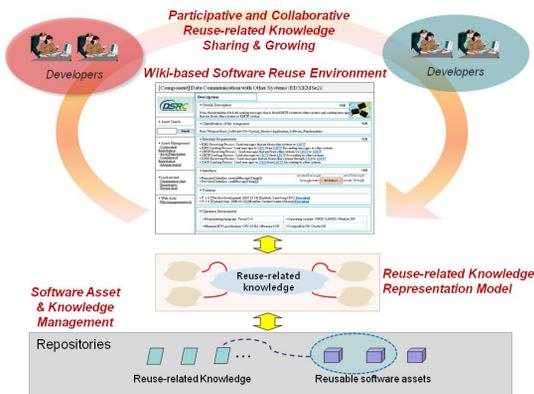


그림 1. 소프트웨어 재사용 관련 지식의 공유와 활용 개념도

본 논문에서는 소프트웨어 재사용 증대를 위한 재사용 관련 지식의 표현, 관리 및 공유 방법에 대해 설명한다. 그림 1은 개략적인 연구 방향 및 목표를 보여준다. 본 연구에서는 온톨로지 기반의 재사용 관련 지식 표현 모델을 개발하였고, 지식들은 이 표현 모델에 따라 기술되어 저장소에 저장된다. 개발자들은 위키 기반의 재사용 환경을 이용하여 지식을 협력적으로 공유, 증식 및 정제할 수 있다. 위키 기반의 지식 관리는 협력적 방법을 제공하기 때문에 좀 더 신뢰성 있고 검증된 지식을 제공할 수 있다[8,9]. 또한, 위키는 사용하기 쉬운 방법을 제공함으로써 개발자들의 참여와 협력을 유발하는데 도움을 준다[10].

위 연구 목표에 따라 위키 기반 재사용 환경의 프로토타입을 구현하였고, 재사용 관련 지식에서 가장 중요하게 고려되어야 하는 재사용 문제 해결에

초점을 두고 평가 방법을 마련하고 있다. 재사용 관련 지식 표현 모델과 위키 기반의 지식 관리 방법을 통해 개발자들의 참여를 유도하여 재사용 관련 지식의 효율적인 공유 및 증식이 가능하고, 정형화되고 체계적인 저장 및 관리를 통해 재사용 문제 해결에 도움을 줄 수 있으리라 예상된다.

다음 장에서는 관련 연구에 대해 간략히 기술하고, 3장에서는 재사용 관련 지식 표현 모델에 대해 설명한다. 4장에서 소프트웨어 재사용 위키의 아키텍처와 프로토타입에 대해 설명하고, 5장에서 연구의 평가 계획에 대해 논한다. 마지막으로 6장에서 결론 및 향후 연구에 대해 설명하고 논문을 마무리 한다.

## 2. 관련연구

본 장에서는 소프트웨어 공학을 위해 진행 중인 위키 기반의 협력적 지식 관리 환경들에 대해 논한다.

SmartAPI는 새로운 소프트웨어를 개발할 때, 개발자들에게 관련되고 연관 있는 API들을 추천하여 제공한다[11]. 시맨틱 태깅 방법을 통해 API 콜들을 분석하여 자동적으로 검색하고 추천하는 방법을 제공한다. Ontology-based OSS(Open Source Software)는 오픈 소스 개발 도메인을 위한 온톨로지 기반의 재사용 지원 방법을 제공한다[12]. 이는 개발자들과 저장소간의 비 동기 적인 상호 작용을 가능케 하고, 온톨로지 기반의 재사용 포털을 제공한다. 나사의 재사용 포털(Reuse Portal)은 지구 과학 커뮤니티를 위한 재사용 지원 포털이다 [13]. 이는 분산된 환경의 재사용 자산들을 통합적으로 관리할 수 있도록 지원해 주고, 이 자산들을 다른 개발자들과 공유 할 수 있도록 지원해 준다.

SRS(Software Reuse System)는 시맨틱 웹을 기반으로 한 재사용 지원 환경으로 개발자의 현재

개발 단계에 따른 지식과 자산을 추천하여 주는 기능을 제공한다[14]. Wikitology는 시맨틱 위키의 개념을 기반으로 자가 구성 재사용을 지원하기 위함이다[15]. 이를 위해 온톨로지 기반의 지식 통신, 취합, 추론 기능을 제공한다. Ontobrowse는 재사용 관련 지식을 검색 및 구조화하기 위해 비정형적, 정형적 문서 관리 방법을 제공하고, 문서들의 기술을 위한 웹 인터페이스를 제공한다[16].

현재 관련 연구들은 주로 웹이나 위키 기술을 소프트웨어 재사용을 위해 접목시키는 수준이고, 정형화된 지식의 표현 및 저장, 관리 방법에 대해서는 아직 부족한 실정이다.

### 3. 재사용 관련 지식 표현 모델

본 장에서는 재사용 관련 지식 표현 모델에 대해 자세히 설명하고, 모델을 구성하기 위해 파악된 필수적인 요소들과 그 관계에 대해 설명한다.

재사용 관련 지식 표현 모델을 개발하기 위해 꼭 필요한 필수요소들을 관련연구들을 통해 아래와 같이 파악하였다[17, 18, 19, 20].

- 프로젝트(Project): 자산이 개발된 프로젝트의 정보를 표현한다. 개발자들은 자신이 현재 개발하고자 하는 소프트웨어와 유사한 기존 프로젝트 정보에서 재사용 자산들과 지식을 얻을 수 있다. 기능적 요구사항 관점에서 중요한 정보를 제공할 수 있다.
- 도메인(Domain): 자산이 개발되어 활용되는 응용 도메인을 나타낸다. 도메인 정보는 공통 정의, 공통 개념, 공통 제약사항 등의 도메인 정보를 제공할 수 있다.
- 의존성(Dependency): 의존성은 재사용할 자산이 다른 특정 자산들과 통합되어 사용되어야 한다는 정보를 나타낸다. 이를 통해 개발자들은 필요한 타 자산을 쉽게 파악할 수 있다.
- 연관성(Relevancy): 자산과 관련된 산출물들의 정보를 나타낸다. 요구사항 분석서, 설계서 등의 부가 산출물

들의 정보를 제공함으로써 개발자들이 재사용하기 용이하게 도와준다.

- 재사용 문제 해결(Reuse Problem Solving): 소프트웨어 재사용은 크게 문제 해결 활동으로 볼 수 있다. 개발자들은 타 개발자가 개발한 자산을 재사용 시 보통 어려움과 문제점에 직면하게 된다. 다른 개발자들이 재사용 활동에 있어서 겪었던 문제점과 해결 경험을 지식 화하여 공유한다면 이는 다른 개발자들의 재사용 증대에 기여를 할 수 있다.
- 피드백(Feedback): 재사용함에 있어서 자산의 평가 정보를 나타낸다. 자산의 재사용 용이성, 어려운 점, 개선 방향 등을 지식 화하여 타 개발자들과 공유할 수 있게 지원한다.
- 지도(Guidance): 명백한 재사용 설명이나 튜토리얼 등의 문서가 있는지 여부를 나타낸다. 재사용할 때 수행해야 하는 작업들, 재사용의 적절성 여부를 파악하기 쉽게 도와준다.
- 재사용 히스토리(Reuse History): 자산의 재사용 역사를 보여준다. 재사용 되었던 도메인, 재사용 횟수 등을 지식 화하여 제공함으로써 개발자들이 얼마나 신뢰성 있는 자산인지 어떠한 분야에 재사용 되었었는지를 쉽게 알 수 있도록 지원한다.

그림 2는 이러한 요소들로 구성된 재사용 관련 지식 모델을 나타낸다. 프로젝트 정보는 ID, 이름, 명세, 기간의 세부요소로써 표현되고, 도메인 정보는 ID, 응용도메인 리스트, 도메인 제약사항의 세부요소로 이루어진다. 의존성은 의존성이 있는 타 자산들의 리스트, 의존성 타입, 상호 작용 다이어그램으로 구성되고, 연관성은 산출물 리스트, 자산과의 관계의 세부 요소를 가진다. 지도는 명백한 문서의 존재 여부, 문서의 위치, 링크로 기술되며, 재사용 문제 해결은 문제의 타입, 가능한 문제 해결 방안, 문제의 기술로 표현된다. 마지막으로 피드백은 추천할 만한 자산인지 여부, 약점, 기술로 표현된다.

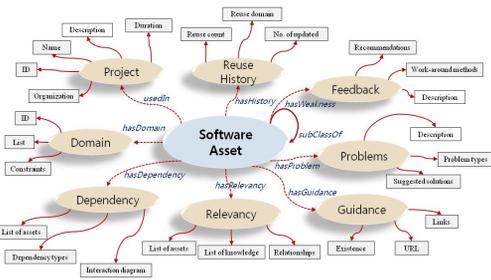


그림 2. 재사용 관련 지식 표현 모델

요소들 중 가장 중요하게 고려되어야 할 재사용 문제 해결 지식을 위해 재사용 시 주로 직면하게 되는 문제점들과 그 원인을 분석하였다. 표 1은 이러한 문제점들과 원인을 보여준다. 표 1에서 신뢰성 문제는 재사용 자산이 잘 테스트 되지 않아 검증되지 않았을 때 발생하고, 컨트롤이나 코드 상의 문제가 그 원인이 된다. 호환성 문제는 도메인이나 환경 등의 외부적인 제약사항으로 인해 발생하는 문제점이다. 이 문제에는 버전이나 표준적인 문제도 포함하고 있다. 상호운용성 문제는 버전이 맞지 않거나 꼭 통합되어야 하는 타 자산들이 있을 시 발생하게 되며, 비완전성은 재사용 자산 패키지에 필요한 요소들이 빠져 있거나 코드가 불완전해서 바로 재사용이 불가능한 경우 발생할 수 있다. 가시화 문제는 사용자 인터페이스가 알아보기 힘들거나 사용이 어려워 바로 재사용하기 힘들 때 발생할 수 있다.

표 1. 재사용 문제점과 그 원인

Problem types	Causes of problems
Reliability	Control problem / Code failure / Function failure
Compatibility	Domain constraints / Environmental constraints / Version problem / Standardization
Interoperability	Dependable assets / Version problem
Incompleteness	Incomplete package / Incomplete functions
Visualization	Readability / Usability

현재 이러한 문제점들과 원인을 바탕으로 하여 반자동적으로 가능한 해결방안을 추천해 주기

위해 지식의 정형화 방법, 추론 기반의 자동적 문제 해결방안 추천 기법을 연구 중에 있다.

#### 4. 소프트웨어 재사용 위키

본 장에서는 현재 구현 중인 소프트웨어 재사용 위키의 간략한 아키텍처와 구현된 프로토타입에 대해 설명한다.

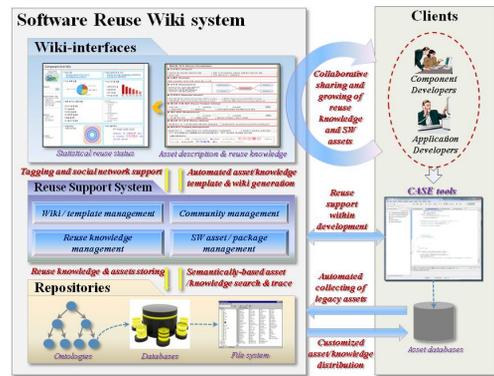


그림 3. 소프트웨어 재사용 위키 아키텍처

그림 3은 소프트웨어 재사용 위키 시스템의 개략적인 아키텍처를 보여준다. 아키텍처는 크게 네 부분으로 구성 된다: 위키 인터페이스, 재사용 지원 시스템, 저장소, 클라이언트. 위키 인터페이스는 개발자들로 하여금 쉽게 지식 관리를 할 수 있도록 웹 기반의 인터페이스를 제공하고 자산 및 재사용 관련 지식 명세를 위한 틀을 제공한다. 위키의 메인 페이지는 등록된 전체 자산의 현재 재사용 현황을 통합적으로 보여주고, 자산 위키 페이지는 자산 명세와 재사용 관련 지식을 보여준다. 재사용 지원 시스템은 위키 페이지 관리를 위한 중요 메커니즘과 지식의 템플릿을 제공하여 주고, 개발자 커뮤니티를 생성 및 관리할 수 있는 방법을 제공한다. 재사용 관련 지식과 자산 관련 정보들을 저장소에 저장 및 관리 할 수 있는 방법 또한 제공한다. 저장소는 실제 명세 된 자산 및

재사용 관련 지식의 물리적 저장 공간을 제공하고, 지정된 도메인 분류에 따라 체계적으로 관리할 수 있도록 지원하여 준다. 클라이언트 시스템은 개발자들이 로컬 컴퓨터를 이용하여 재사용 활동을 할 때 개발 도구와 연계하여 적절한 지식을 추천해 줄 수 있도록 지원해 주는 기능을 수행한다.

소프트웨어 재사용 위키의 자산 페이지는 크게 두 자산 명세 페이지, 재사용 관련 지식 페이지의 두 부분으로 구성된다. 그림 4는 자산 명세를 위한 재사용 위키 페이지의 한 예제를 보여준다. 자산 명세를 위한 요소들로는 간략한 기술, 자산의 분류, 자산이 개발되게 된 중요 요구사항들, 인터페이스, 버전 정보, 운용 환경 등이 기술되어 있다. 본 예제에서는 국방 도메인에서 개발된 “Data Communication with Other System (EDXKMSe2i)” 라는 컴포넌트의 명세를 보여주고 있다.

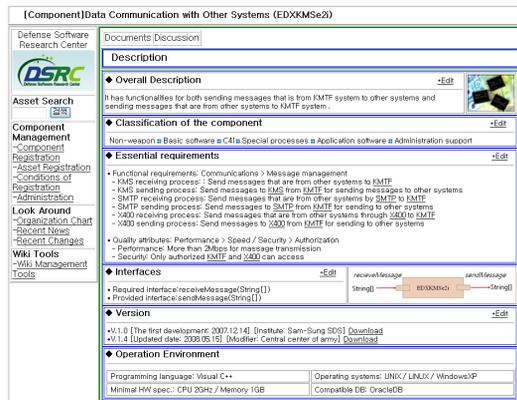


그림 4. 자산 명세를 위한 소프트웨어 재사용 위키 페이지 예제

그림 5는 재사용 관련 지식 명세를 위한 소프트웨어 재사용 위키 페이지의 한 예제를 보여준다. 재사용 관련 지식은 3장에서 설명한 재사용 지식 표현 모델에 따라 구성되어져 있다. 개발된 프로젝트, 재사용 가능한 응용 도메인, 의존성 있는 타 컴포넌트, 재사용 문제 해결, 피드백, 재사용 히스토리, 태깅 관리 부분으로 구성되어 있다. 본 재사용 관련 지식 예제는 “Data Communication

with Other System (EDXKMSe2i)” 컴포넌트를 위한 재사용 관련 지식으로 국방과학연구소에서 공통 통신관련 컴포넌트를 제공하기 위해 개발되었다는 것을 알 수 있다. 응용 도메인은 타 체계와 통합되어 통신을 하기 위한 분야라는 것을 알 수 있고, “EDXProactor”, “KMS MS”, “SMTP MS”, “X400”의 컴포넌트들과 서로 연계 및 통합되어 재사용되어야 한다는 것을 의존성 부분에서 가시화하여 보여주고 있다. 연관 산출물들의 존재 여부 및 획득은 연관성 필드에서 나타내 주고 있다. 재사용 문제 해결 필드는 앞서 정의한 재사용 관련 문제들의 타입을 선택할 수 있도록 지원하고, 가능한 해결 방안을 입력할 수 있는 인터페이스를 제공한다. 피드백은 자산의 재사용에 있어서의 평가와 개선 방향을 타입으로 선택할 수 있도록 설계했으며, 재사용 히스토리에서는 본 자산이 8번 재사용에 활용되었으며 어떤 도메인에서 재사용되었는지 여부와 랭킹 정보를 나타내고 있다.

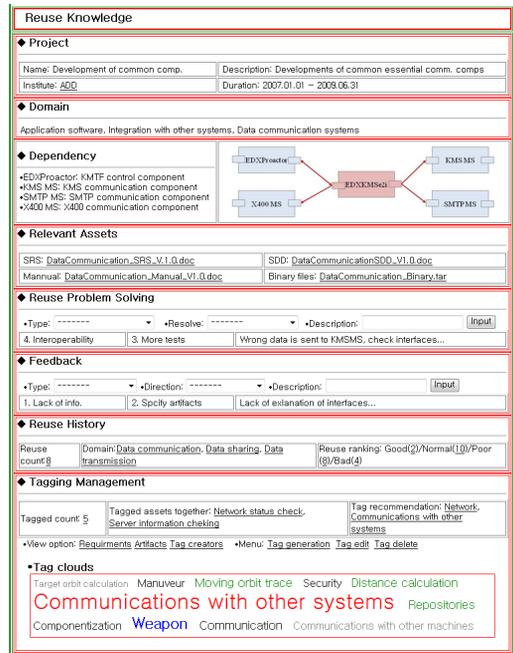


그림 5. 재사용 관련 지식 명세를 위한 소프트웨어 재사용 위키 페이지 예제

이렇게 자산에 따라 구조화되어 관리되는 재사용 관련 지식은 개발자들로 하여금 어떻게 재사용할 지를 한눈에 파악할 수 있도록 도와준다. 또한, 지식 명세를 위한 타입들을 최대한 분석 및 정의하여 텍스트의 입력이 아닌 선택을 할 수 있도록 지원함으로써 지식 명세를 위한 부담을 감소시킬 수 있다. 위키 기반의 편집 방식을 활용하였으므로 여러 사용자들이 자신의 지식을 쉽게 축적할 수 있도록 지원하여 협력적 공유, 증식 및 정제를 쉽게 할 수 있도록 지원한다.

## 5. 평가 계획

본 연구는 국방 도메인에서의 소프트웨어 재사용성 증대를 위해 진행되고 있다. 그러나 국방 도메인의 특성상 데이터 수집이 어렵고 실제적인 평가가 어렵기 때문에 현재 소프트웨어 재사용 위키를 서비스 매쉬-업 (Mash-ups) 도메인에 활용하여 평가할 계획을 진행 중에 있다. 평가는 재사용 관련 지식 모델에서 재사용 문제 해결에 초점을 두고 사용자 스터디 (User studies)를 수행하여 진행할 예정이다. 재사용 문제 해결 방안 추천을 위해 오픈 소프트웨어 개발 사이트인 소스포지 (Sourceforge.net)에서 오랜 기간 동안 해결되지 않는 문제 보고들을 수집해 보았다. 현재 가장 활발히 재사용이 이루어지고 있는 사용자 인터페이스 (User interfaces) 카테고리를 선택하여 재사용 문제를 가지고 있는 200개의 자산 데이터를 수집하였다. 200개의 자산에 대한 재사용 문제를 분석하여 보니 70개의 자산에 대해 258개의 풀리지 않았던 문제를 추출할 수 있었다. 이 문제들 중 179개의 문제들이 본 연구에서 정의한 문제 타입에 따라 분류할 수 있었고, 그 중 128개의 문제는 본 연구에서 제공하는 재사용 관련 지식으로 풀릴 수 있다는 것을 알아내었다.

이는 수동적인 데이터 분석을 통해 알아낸 결과이고, 보다 정량적이고 정성적인 평가를 위해 현재 재사용 관련 지식의 정형화된 표현 방법과 문제 해결을 위한 룰 기반의 해결방안 추천 기법을 연구 중에 있다. 지식이 정형화된 표현 기법을 통해 저장되고, 룰에 따라 재사용 문제를 해결할 수 있는 해결방안을 추천하여 주어 개발자의 재사용에 도움을 준다는 방향으로 평가를 계획 중에 있다. 앞서 설명한 바와 같이 엔드 유저(End-users)들도 애플리케이션을 개발할 수 있도록 지원하는 서비스 매쉬-업에서의 문제해결을 위한 사용자 스터디를 통하여 이러한 평가를 진행할 계획이다.

## 6. 결론 및 향후연구

소프트웨어 재사용은 기존에 개발된 소프트웨어를 활용하여 소프트웨어의 신뢰성 및 생산성을 향상시키고, 개발 비용과 시간을 단축시키고자 하는 패러다임이다. 이러한 재사용의 장점은 많은 연구자들에 의해 인정을 받았지만 현실적으로는 재사용이 잘 이루어지지 않고 있는 실정이다. 이에 본 연구에서는 소프트웨어 재사용에 있어서의 실질적인 문제들을 파악하여 이를 해결하기 위한 방안을 연구 중에 있다.

본 논문에서는 소프트웨어 재사용 시 중요한 정보를 제공하여 주는 재사용 관련 지식을 표현, 관리, 공유함에 있어서의 문제점들을 파악하였고, 지식을 정형화되고 체계적으로 관리할 수 있도록 지원하기 위한 재사용 관련 지식 표현 모델을 제시하였다. 재사용 관련 지식을 표현하기 위해 꼭 필요한 요소들을 정의하였고 중요 요소들에 대한 타입을 정의하였다. 협력적 지식의 공유, 증식 및 정제를 위한 위키 기반의 소프트웨어 재사용 환경의 아키텍처를 정의하였고, 재사용 관련 지식 표현 모델 및 아키텍처에 따른 프로토타입을 구현

하였다. 본 연구를 통해 소프트웨어 재사용에 있어서의 가장 큰 문제점인 개발자들의 부담을 감소 시킴으로써 개발자들의 참여를 유도하여 재사용 관련 지식의 효율적인 공유 및 증식이 가능케 하고, 정형화되고 체계적인 저장 및 관리를 통해 재사용 문제 해결에 도움을 줄 수 있으리라 예상한다.

현재 재사용 관련 지식의 정형화된 표현법을 개선 중에 있고, 재사용 문제 해결 방안 추천을 위한 정형화된 룰을 데이터 분석을 통해 진행 중에 있다. 또한, 이러한 기술들을 바탕으로 추론 기반의 자동적 지식 생성 및 추천 방안을 연구 중에 있다. 또한 본 연구의 효과적인 평가를 위하여 세부 기능들을 구현 중에 있고, 서비스 매쉬-업을 지원하는 사용자 실험 환경과 방법을 마련 중에 있다.

## 참 고 문 헌

- [1] William Frakes and Carol Terry, "Software Reuse: Metrics and Models", ACM Computing Surveys, 1996.
- [2] Ali Mili and et al., "Toward an Engineering Discipline of Software Reuse", IEEE Software, 1999.
- [3] O. k. Harsh, "Data, Information and Knowledge & Reuse Management Techniques", Proceedings of the World Congress Engineering (WCE), 2007.
- [4] Hafedh Mili and et al., "Reusing Software: Issues and research directions", IEEE Software Engineering, 21 (6), 1995.
- [5] Jumming Hou and et al., "A Framework for Knowledge Reuse Based on Ontology in Collaborative Design", Proceedings of the International Conference on Intelligent Networks and Intelligent Systems, IEEE, 2008.
- [6] Timo Kucza and et al., "Improving Knowledge Management in Software Reuse Process", PROFES, LNCS 2188, pp. 141-152, 2001.
- [7] Iona Rus and Mikael Lindvall, "Knowledge Management in Software Engineering", IEEE Software, 2002.
- [8] Stephan J. Andriole, "Business Impact of Web 2.0 Technologies", Communications of the ACM, 2010.
- [9] Ammy Jiranida Phuwanartnurak, "Interdisciplinary Collaboration through Wikis in Software Development", Wikis4SE' 09, ICSE, 2009.
- [10] Sajjan G. Shiva and Lubna A. Shala, "Using Semantic Wikis to Support Software Reuse", Journal of Software, Vol. 3, No. 4, 2008.
- [11] Andreas Eberhart and Sudhir Agarwal, "Smart API - Associating Ontologies and APIs for Rapid Application Development", Ontologien in der und f"ur die SoftwaretechnikWorkshop anlasslich der Modellierung, 2004.
- [12] Tharam S. Dillon and et al., "Semantic Web Support for Open-source Software Development", IEEE International Conference on Signal Image Technology and Internet Based Systems, 2008.
- [13] Ryan Gerard and et al., "The Software Reuse Working Group: A Case Studying in Fostering Reuse", NASA, IEEE, 2007.
- [14] Bruno Antunes and et al., "SRS: Software Reuse System based on Semantic Web", Proceeding of the 3rd International Workshop on Semantic Web Enabled Software Engineering, 2007.
- [15] Bjorn Decker and et al., "Self-organized Reuse of Software Engineering Knowledge

Supported by Semantic Wikis” , Proceedings of Workshop on Semantic Web Enabled Software Engineering (SWESE), IEEE, November, 2005.

- [16] H. Happel and S. Seedorf, “Ontobrowse: A Semantic Wiki for Sharing Knowledge about Software Architectures” , In Proceedings of the 19th International Conference on Software Engineering and Knowledge Engineering (SEKE), 2007.
- [17] Hing Yan Lee and et al., “Software Engineering Knowledge for Software Reuse” , IEEE, 1993.
- [18] V. R. Basili and et al., “Towards A Comprehensive Framework for Reuse: A Reuse-Enabling Software Evolution Framework” , Technical Paper, CS-TR-2158, 1988.
- [19] Eric Ras and Sebastian Weber, “Software Organization Platform: Integrating Organization and Individual Learning” , Proceedings of Wiki4SE’09, ICSE, 2009.
- [20] Frank McCarey and et al., “Knowledge Reuse for Software Reuse” , Web Intelligent and Agent Systems: An International journal 6, 2008.

2004년 금오공과대학교 컴퓨터공학과 학사  
 2007년 한국과학기술원 전산학과 석사  
 2007년 한국전자통신연구원 위촉연구원  
 2007~현재 한국과학기술원 전산학과 박사과정



**고 인 영**

1990년 서강대학교 전산학과 학사  
 1992년 서강대학교 전산학과 석사  
 2003년 미국 Univ. of Southern California 박사  
 1993년~1996년 공군사관학교 교관 (전임강사)  
 1997년 미국 USC Information Science Institute (ISI) 연구 조교  
 2003년 미국 USC ISI Postdoctoral Research Associate  
 2004년~현재 한국과학기술원 전산학과 부교수

**저 자 소 개**



**구 형 민**



# 스마트 폰 기반 계층적 모바일 컨텍스트 모델 및 사용자 상황 추론 기법<sup>†</sup>

(A Hierarchical Mobile Context Model and User Context Inference Methods based on Smart Phones)

이미연<sup>‡</sup>      이정원<sup>§</sup>      박승수<sup>‡</sup>  
(Meeyeon Lee) (Jung-Won Lee) (Seung Soo Park)

**요약** 스마트 폰은 내장된 다양한 센서와 사용자 휴대성/밀착성으로 인해 정보 수집과 지능형 서비스의 적합한 대상으로 주목 받고 있다. 즉, 센서로부터 사용자의 주변 환경 정보뿐만 아니라 스마트 폰의 사용 정보 등을 수집하여 사용자의 현재 상태를 추론할 수 있고, 추론된 상태 정보는 사용자에게 상황 인지 서비스를 제공하기 위한 중요한 근거로 활용될 수 있다. 하지만 제공하고자 하는 서비스에 따라 필요한 상황 정보가 다르기 때문에, 정확한 상황 추론을 위해서는 컨텍스트 모델링 기법이 전제되어야 한다. 따라서 본 논문에서는 스마트 폰 사용자의 일상생활 상에서의 상황을 추론하기 위한 모바일 컨텍스트 계층 모델을 제안한다. 센서 데이터로부터 추론할 수 있는 상위 컨텍스트를 컨텍스트-행위-상황의 3-계층으로 분류하여 정의하고, 각 단계를 위한 추론 기법을 제시한다. 이 모델을 통해 사용자의 단순 행위 또는 상태가 아니라 일상생활에서의 의미 있는 상황 추론이 가능해질 것이다.

**키워드** 스마트 폰, 상황 추론, 모바일 컨텍스트, 계층 모델

**Abstract** Since smart phones have various embedded sensors and high portability/usability, they have emerged as suitable targets to collect information and to provide intelligent services. That is, with a smart phone, we can collect information about user's circumstances and phone usage from sensors and infer his/her current state which is the significant basis for context-aware services. However, a service system should be founded on a context model to ensure reasonable context-awareness, because context information the system needs depends on its target services. Therefore, in this paper, we propose a hierarchical mobile context model for context inference of smart phone users in their daily life. We classify high-level context which can be draw from sensing data into three levels, Context-Behavior-Situation, and define inference methods for each level. With our mobile context model, we can user's meaningful context in his/her daily life besides simple actions or states.

**Key words** Smart Phone, Context Inference, Mobile Context, Hierarchical Model

## 1. 서론

상황 인지(context awareness)는 일반적으로 유비쿼터스 또는 퍼베이시브 환경에서 개체의 상황을 정의하고 특징지을 수 있는 정보인 컨텍스트를 감지하여 적절한 상황 판단을 하는 기술을 의미한다[1]. 사용자, 환경, 기기와 같은 개체의 변화하는 상태를 인식하고 컴퓨팅 개체가 내장

<sup>†</sup> 본 연구는 지식경제 프론티어 기술개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스 컴퓨팅 및 네트워크 원천 기술개발사업의 11C3-T3-10M 과제로 지원된 것임.

<sup>‡</sup> 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2011-0005305).

<sup>§</sup> 비 회 원 : 이화여자대학교 컴퓨터공학과  
ailmy@ewhain.net, sspark@ewha.ac.kr

<sup>§</sup> 정 회 원 : 아주대학교 전자공학부  
jungwony@ajou.ac.kr

논문접수 : 2011년 01월 13일

심사완료 : 2011년 02월 25일

되어 있는 환경에서의 지능형 서비스를 결정하는 상황 인지 기술은 유비쿼터스 컴퓨팅의 가장 중요한 기반 기술이다. 최근에는 스마트 폰의 발달과 대중화로 모바일 환경에서의 상황 인지에 대한 관심과 연구가 활발하게 이루어지고 있다. 스마트 폰은 별도의 센서를 부착할 필요가 없을 정도로 다양한 센서가 자체 내장되어 있고 사용자들이 항상 휴대하고 능동적으로 사용하기 때문에 데이터 수집에 유용하다. 이러한 장점으로 인해 스마트 폰 상에서의 다양하고 지능화된 서비스에 대한 요구도 증대되고 있다.

그로 인해 스마트 폰을 활용한 상황 인지 및 자동 서비스에 대한 연구와 기술 개발이 활발히 진행되고 있다. 사용자의 위치, 주변 환경의 상태 등을 스마트 폰 센서로 감지하여 사용자의 현재 상태와 행위를 추론하고, 이를 근거로 스마트 폰의 설정을 변경하거나 적합한 콘텐츠를 제공하고자 하는 연구들이다. 하지만 대부분이 목표 서비스의 부재 또는 단순함으로 인해, 추론하고자 하는 컨텍스트가 사용자의 단일 행위로 국한되어 있다는 한계가 있다.

따라서 본 논문에서는 스마트 폰 상의 어플리케이션을 활용한 지능형 서비스 제공을 위한 선행 연구로서, 사용자의 일상생활(daily life) 상에서의 컨텍스트를 모델링하고 추론하기 위한 개념 모델을 정의한다. 일반적으로 컨텍스트를 상위-하위로 구분하는 기존의 컨텍스트 모델과 달리 다중 레벨로 세분화하여 사용자의 상황을 의미 있는 수준으로 정의하고 추론한다.

## 2. 관련 연구

스마트 폰의 등장 초기부터 모바일 컨텍스트 표현 기법과 사용자의 상태 추론에 대한 연구들이 이루어지고 있다. VTT 연구소에서는 다양한 자동

서비스를 위한 모바일 상황 인지 연구를 진행해 오고 있다. 내장 센서 데이터로부터 자동차 운전, 걷기, 축구 게임과 같은 사용자의 상태 및 주요 행동을 파악하기 위한 컨텍스트 표현 방법을 제안하였다. 그림 1과 같이 기본적으로 모바일 컨텍스트에서 필요한 위치, 시간, 환경적인 정보들을 컨텍스트 타입으로 분류하고 각각의 컨텍스트 값, 신뢰도, 속성 등의 정보를 기술하기 위한 온톨로지 구조를 정의하였다. 이를 기반으로, 베이지안 네트워크와 SVM(Support Vector Machine) 등의 데이터 마이닝 기법을 적용하여 상황을 추론하고, 그에 따라 화면 밝기 조절, 글씨 크기 조절과 같은 폰의 기본적인 기능을 자동 조작하는 서비스를 구현하였다[2,3,4].

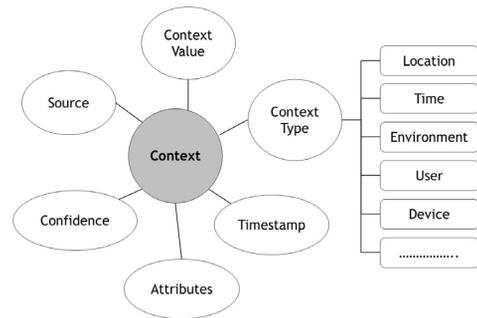


그림 1. VTT의 컨텍스트 모델

스마트 폰 기반의 모바일 컨텍스트를 표현하는 기법을 제안한 대표적인 연구들 중, 사용자의 상태와 프로필을 기반으로 적합한 콘텐츠를 추천하기 위해 사용자의 일정, 선호 항목, 관심 항목 등을 기술하는 모델도 제안되었다[5]. 또한 시간과 사용자의 위치에 따라 사용자의 목표와 역할을 구분하고 그에 맞는 정보를 스마트 폰에서 제공하고자 하는 연구도 진행되었다[6]. 대부분의 기존 모바일 컨텍스트 모델은 공통적으로 위치, 시간, 환경적 정보, 사용자 정보를 포함하고 있지만, 제공하고자 하는 서비스에 따라 범위와 표현 방식에 차이점이 있음을 알 수 있다.

사용자의 현재 상황을 추론하고자 하는 연구들도 최근까지 이루어지고 있는데, UCLA에서는 부동, 걷기, 뛰기, 자전거, 자동차의 5가지 이동 모드를 정확하게 인지하기 위한 다양한 실험 결과를 제시하고 있다[7]. 또한 최근에는 위치 및 장소가 사용자의 상태 또는 행위에 대한 결정적 단서가 된다는 가정을 바탕으로 사용자의 상황을 추론하는 방법도 제안된 바 있다[8]. 하지만 기존의 대부분의 스마트 폰 관련 상황 인지 연구들은 컨텍스트를 상위-하위로 분류하고, 센싱 데이터를 하위 컨텍스트로 간주한다. 상위 컨텍스트는 목표로 하는 서비스를 결정하는 근거가 되는 컨텍스트로, 추론하고자 하는 상황들의 집합이다. 따라서 기존의 컨텍스트 모델이나 추론 방법들은 제공하고자 하는 서비스에 따라 사용자의 행위나 상태를 단일 레벨로 정의하고 인지하기 때문에 본 연구가 목표로 하는 사용자의 일상생활을 표현하는데 한계가 있다.

### 3. 계층적 모바일 컨텍스트 모델

기존의 모바일 컨텍스트 모델링 및 인지 연구들은 추론하고자 하는 상위 컨텍스트를 사용자의 의미적 위치, 주변 환경 상태, 사용자의 단일 행위로 규정하였다. 본 연구는 사용자의 일상 생활을 추적함으로써 단순 행위 레벨이 아니라 의미 있는 상태를 인지하여 해당 상태에 적합한 서비스를 스마트 폰을 통해 제공하고자 한다. 이를 위해 필요한 지식 베이스로서, 스마트 폰의 센서로부터 수집된 센싱 데이터의 상위 개념인 컨텍스트를 그림 2와 같이 3가지 레벨로 분류한다.

즉, 센싱 데이터와 컨텍스트 개념을 분리하고, 추론하고자 하는 대상의 크기와 추론을 위해 필요한 정보에 따라 컨텍스트를 “컨텍스트(Context)” - “행위(Behavior)” - “상황(Situation)”의 다중 레벨로

세분화한다. 3-계층 컨텍스트 모델에서는 센서를 통해 얻을 수 있는 가공하지 않은 데이터는 컨텍스트에 포함하지 않는다. 각 레벨의 컨텍스트는 하위 단계의 정보의 조합으로 표현 및 추론될 수 있는데 다음 세부 절에서 설명한다.

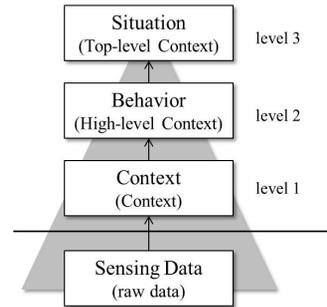


그림 2. 모바일 컨텍스트의 3-계층 모델

본 논문에서는 센싱 데이터의 수집 단계의 자세한 기법은 배제하고 상위 컨텍스트에 초점을 맞춘다.

#### 3.1 레벨 1: 컨텍스트 (Context)

컨텍스트는 “센서를 통해 수집한 1차적인 원시 데이터로부터 간단한 가공을 거쳐 얻을 수 있는 정보”로 정의한다. 고차원적인 기법을 적용하지 않고 대부분 1개의 센서로부터 판단할 수 있는 단계로서, 대부분 사용자의 상태보다는 환경적인 상태가 해당된다. 스마트 폰 센서를 활용하여 사용자의 로그를 수집하고 저장하는데 각 로그 속성 정보를 가공하여 의미적으로 분류하거나 표현한다.

표 1은 사용자 로그 데이터의 구조로, 11개 속성 정보로 구성된다. 이중에서, Location, Site, Speed 등은 센싱된 숫자 데이터가 아니라 의미 있는 정보 형태로 변환하여 저장한다.

컨텍스트를 포함한 로그 데이터는 다음 단계인 사용자의 행위를 추론하는 기반 정보로 활용된다.

표 1. 사용자 로그 데이터 구조

로그 속성	설명	값
Timestamp	발생 시간	yyyy-mm-dd HH:mm:ss
GPS-latitude	위도	(Real)
GPS-longitude	경도	(Real)
Location	의미적 위치	{Home, Office, Home-around, Office-around, Outside}
Site	실내/실외	{inDoor, outDoor}
Speed	실외 이동 속도	{Fast, Normal, Slow}
P_movement	운동량	{High, Medium, Low}
Illuminance	조도 레벨	{High, Medium, Low}
Noise	소음 레벨	{High, Medium, Low}
Mode	폰의 상태	{Vibrate, Silent, Normal}
App	실행 중인 어플리케이션	(String)

3.2 레벨 2: 행위 (Behavior)

행위는 “스마트 폰의 센싱 데이터로부터 추론 가능한 범위의 사용자 행동 또는 상태”이다. 사용자의 단일 상태 또는 행동으로서, 1단계 컨텍스트로부터 인지 가능하고, 3단계의 상황 정의에 필요한 기본적인 행위를 추출하여 정의하였다. 예를 들어, 걷기, 운동하기, 앉기, 수면하기 등과 같은 사용자의 행동이 이 단계에 해당하고, 1단계까지 가공된 로그 데이터로부터 추론된다.

기존의 모바일 상황 인지 연구들처럼 베이지안 네트워크, HMM(Hidden Markov Model) 등의 데이터 마이닝 기법을 적용하여 추론할 수 있다. 그림 3은 베이지안 네트워크를 통해 학습된 모델의 예를 도식화하고 있다.

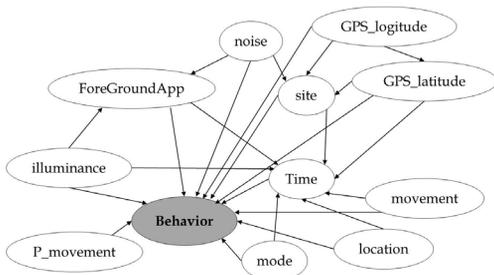


그림 3. 행위 컨텍스트 (2단계) 추론을 위한 베이지안 네트워크의 예

3.3 레벨 3: 상황 (Situation)

본 논문에서 제안한 계층 모델의 최상위 컨텍스트인 상황은 “사용자의 일상생활에서의 주요 일정 또는 상황”으로 정의한다. 즉, 사용자의 단일 행동이 아니라 일련의 다중 행동들과 환경적 요인의 복합적인 조합을 통해 알 수 있는 일정상의 복합 상황이다. 일반적으로 규칙적인 일정으로 표현할 수 있는 회사원의 일상생활을 사례 연구로 선정하여 기상, 출근, 업무 등의 상황을 정의하였다. 또한, 그림 4와 같이, 사용자의 일상생활은 배타적으로 분리되지 않고 연속적이기 때문에 상황들의 시간적인 연속성을 반영하기 위해 각 상황(S<sub>i</sub>)을 해당 상황의 완료 정도에 따라 과거(p), 현재(c), 미래(f)의 3가지 시점으로 모델링하였다.

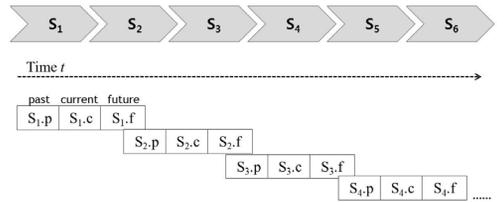


그림 4. 상황 모델링

각 상황은 2단계의 행위 컨텍스트인 사용자의 특정 행동 또는 스마트 폰 상에서의 서비스들의 일련의 집합으로부터 판단할 수 있다. 즉, 각 상황은 사용자의 행동 패턴으로 표현할 수 있다. 사용자의 행동 패턴을 추출하기 위해 로그 데이터에 대한 행위를 추론한 후에 각 로그 데이터를 사용자 행동 아이템(User Behavior Item)으로 표현한다. 사용자 행동 아이템(I)은 다음과 같이 로그 데이터의 11개 속성과 행위를 상황 결정에 중요한 요인이 되는 4개의 속성으로 축소한 형태로 표현한다.

$$I = \{Time, Space, Behavior, App\}$$

- Time: 주중/주말여부와시간정보
- Space: 로그데이터의 location 속성정보
- Behavior: 추론된2단계 컨텍스트정보
- App: 로그데이터의 App속성정보

다음 단계로, 변형된 형태인 아이템들을 시간적 순서로 그룹화하여 사용자 행동 시퀀스(User Behavior Sequence; Se)를 생성한다.

$$Se = \left\{ I_i, I_j, \dots, I_k \mid I_i.Time < I_j.Time < I_k.Time \right\}$$

연속적으로 발생하는 아이템들을 시퀀스 그룹으로 포함 또는 분할하기 위해 각 상황을 대표하는 행위들을 정의하였다. 즉, 사용자는 특정 상황에서 특정 행위를 한다는 가정을 기반으로, 모바일 컨텍스트 모델의 2 단계인 행위 레벨에 대해 추출한 행위들 중, 각 상황을 구별할 수 있는 행위들을 정의하여 상황-행위 매칭 규칙을 생성하였다. 이 규칙에 따라 발생한 아이템들의 행위 속성이 특정 상황을 대표하는 행위 그룹에 속하는지 여부에 따라 동일 시퀀스에 포함시키거나 분할한다.

이렇게 생성된 시퀀스 중, 순차 패턴 마이닝 기법을 적용하여 발생 빈도가 임의의 지지도(support) 이상인 시퀀스들을 각 상황에 대한 사용자 행동 패턴(User Behavior Pattern)으로 추출한다. 이 패턴들은 사용자 행동 패턴 저장소(User Behavior Pattern Repository)에 저장되고, 이를 기반으로 실시간에 발생하는 사용자의 로그 데이터로부터 최종적으로 상황을 유추할 수 있다. 표 2는 패턴 추출까지의 세부 단계에서 생성되는 데이터들의 예이다.

표 2. 상황 추론 시 생성되는 데이터의 예

데이터	생성 예
사용자 컨텍스트 로그 데이터베이스 레코드	"2011-06-13 08:40:56", 37.5623058, 126.946718733333, Outside, outDoor, Slow, High, High, Low, Vibrate, null, walk
사용자 행동 아이템	(weekday 07:30, Home, -, alarm) (weekday 09:19, Office-around, walk, -)
사용자 행동 시퀀스	{(weekday 07:30, Home, -, alarm), (weekday 07:33, Home, wake-up, -), (weekday 07:37, Home, still, Mail), (weekday 07:56, Home, breakfast, -)}
사용자 행동 패턴	{(weekday 07:30, Home, -, alarm), (weekday 07:33, Home, wake-up, -), (weekday 07:56, Home, breakfast, -)} → (S1.f)

#### 4. 상황 추론 시스템

앞서 모델링한 3-계층 컨텍스트 모델의 각 단계의 컨텍스트를 추론하기 위한 시스템을 그림 5와 같이 설계하였다. 사용자의 일정을 파악하여 맞춤형을 향상시키기 위해 사전에 훈련 단계를 거쳐 각 사용자에게 대한 행위 추론용 베이지안 네트워크 모델과 상황에 대한 사용자 행동 패턴을 생성한다. 이들을 기반으로 실시간에 발생하는 로그에 대해 상황을 추론할 수 있다.

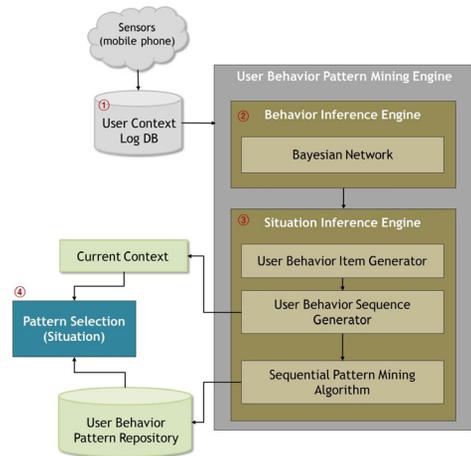


그림 5. 3-계층 컨텍스트 모델 기반 추론 시스템 구조

- ① 사용자 컨텍스트 로그 데이터베이스 생성: 센서 데이터를 가공하여 1단계 컨텍스트로 변형하고 데이터베이스에 저장한다.
- ② 행위 추론: 로그 데이터베이스의 각 레코드에 대한 2단계 컨텍스트를 추론한다.
- ③ 상황 추론(패턴 추출): 12개 속성(11개 로그 데이터 + 행위)으로 구성된 각 레코드를 사용자 행동 아이템으로 변형하고 사용자 행동 시퀀스를 생성한다. 높은 지지도를 갖는 시퀀스들을 선별하여 사용자 행동 패턴으로 추출한다.
- ④ 실시간 상황 추론: 실시간에 발생하는 로그들에 대한 사용자 행동 아이템들을 생성하고, 사용자 행동 패턴과 비교하여 현재의 상황을 결정한다.

```

procedure Pattern_Selection (Current_Sequence)
return Situation
begin
  most_similar = null;
  for (P: each pattern in user behavior pattern repository) do
    begin
      Cal_Similarity(Current_Sequence, P); // ①
    end

  most_similar = P with the max similarity score;
  Situation = Situation of most_similar;

  return Situation;

procedure Cal_Similarity (s, p) // ②
return sim
begin
  sim = 0;
  for (s.I: each item in s, p.I: each item in p) do
    begin
      sim = Sim(s.I.T, p.I.T) + Sim(s.I.S, p.I.S) // ②
        + Sim(s.I.B, p.I.B) + Sim(s.I.A, p.I.A);
    end
  end

  return sim;
end

```

그림 6. 실시간 상황 추론 알고리즘

실시간 상황 추론을 위해서는 그림 6과 같은 알고리즘을 사용한다. 실시간의 사용자 행동 시퀀스로부터 현재의 상황을 결정하기 위해서 사용자 행동 패턴 저장소의 모든 패턴들과의 유사성을 분석한다. 이 알고리즘의 핵심 부분은 현재의 시퀀스와 각 패턴의 유사도를 계산하는 부분이다(①). 3.3절에서 설명한 사용자 행동 시퀀스 형태로 표현된 현재 시퀀스와 패턴은 사용자 행동 아이템들로 구성되어 있다. 따라서 포함하고 있는 아이템들 간의 유사도가 두 시퀀스의 유사도를 결정하기 때문에, 아이템을 구성하는 4가지의 속성 값을 비교한다(②). 시간 인접성과 공간 인접성 계산 기법을 사용하여 시간과 공간 속성의 유사도를 계산하고, 행위와 App 속성은 일치 여부를 타진한다. 현재 시퀀스의 상황은 가장 높은 유사도를 갖는 패턴이 대표하는 상황이다.

$$T_j - T_i < \alpha \wedge S_j \approx S_i \Rightarrow \{I_i, I_j\} \in Sc_k$$

$$I_i = (T_i, S_i, B_i, A_i), I_j = (T_j, S_j, B_j, A_j)$$

$$T_i < T_j$$

$$\approx : \text{within a } \beta\text{-radius}$$

이와 같이, 3-계층 컨텍스트 모델은 스마트 폰 센서를 통해 수집된 원시 데이터로부터 사용자의 일정상의 중요 상황을 추론하기 위한 기반이 된다.

## 5. 결론

스마트 폰 기반의 상황 인지와 지능형 서비스에 대한 요구가 증대되면서 다양한 기법들이 개발되고 있지만, 대부분 상위 컨텍스트로서 사용자의 단일 행동을 유추하는데 그치고 있다. 따라서 본 논문에서는 스마트 폰의 어플리케이션 기반 서비스 제공에 적합한 계층적 컨텍스트 모델을 제안한다. 사용자의 일상생활을 상위 레벨로 모델링하기 위해 센싱 데이터와 컨텍스트 개념을 분리하고 컨텍스트를 3가지 단계로 세분화하였다. 특히 최상위 컨텍스트로 “상황” 단계를 두어 사용자의 일정을 추적함으로써 일상 생활을 지원할 수 있도록 하였다. 제안한 다양한 레벨의 상황 모델을 기반으로 사용자의 상황이 판단되면 이를 기반으로 스마트 폰 상에서의 지능형 서비스가 가능할 것이다. 향후에는 제안한 3-계층 컨텍스트 모델을 바탕으로 스마트 폰 서비스 시스템을 개발할 예정이다.

## 참고 문헌

- [1] M. Perttunen, J. Riekkilä, O. Lassila, “Context Representation and Reasoning in Pervasive Computing: a Review”, International Journal of Multimedia and Ubiquitous Engineering, Vol.4, No.4, pp.1-28, 2009
- [2] P. Korpipää, J. Mäntyjärvi, “An Ontology for Mobile Device Sensor-Based Context Awareness”, 4th International and Interdisciplinary Conference

on Modeling and Using Context, pp.451-458, 2003

- [3] P. Korpipää, J. Mäntyjärvi, J. Kela, H. Keränen, E.J. Malm, “Managing Context Information in Mobile Devices” , Pervasive Computing, Vol.2, No.3, pp.42-51, 2003
- [4] V. Könönen, J. Mäntyjärvi, H. Similä, J. Pärkkä, M. Ermes, “Automatic feature selection for context recognition in mobile devices” , Pervasive and Mobile Computing, Vol.6, No.2, pp.181-197, 2010
- [5] N. Weisberg, R. Gartmann, A. Voisard, “An Ontology-Based Approach to Personalized Situation-Aware Mobile Service Supply” , Geoinformatica, Vol.10, No.1, pp.55-90, 2006
- [6] Y.L. Chang, “A Dynamic User-Centric Mobile Context Model” , A Thesis of University of Waterloo, 2010
- [7] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, M. Srivastava, “Using Mobile Phones to Determine Transportation Modes” , ACM Transactions on Sensor Networks, Vol.6, No.2, pp.13:1-13:27, 2010
- [8] H.Y. Noh, J.H. Lee, S.W. Oh, K.S. Hwang, S.B. Cho, “Exploiting indoor location and mobile information for context-awareness service” , Information Processing and Management, 2011

## 저 자 소 개



이 미 연

2003년 이화여자대학교 컴퓨터학과 졸업(학사).  
2005년 이화여자대학교 컴퓨터학과 졸업(석사).  
2007년~현재 이화여자대학교 컴퓨터공학과 박사과정.

<관심분야> 유비쿼터스 컴퓨팅, 온톨로지, SOA, 인공지능 등



이 정 원

1993년 이화여자대학교 전자계산학과 졸업(학사).  
1995년 이화여자대학교 전자계산학과 졸업(석사).  
1995년~1997년 LG종합기술원 주임연구원.  
2003년 이화여자대학교 컴퓨터학과 졸업(박사).  
2003년~2006년 이화여자대학교 컴퓨터학과 BK교수, 전임강사(대우).  
2006년~현재 아주대학교 전자공학부 조교수.

<관심분야> 유비쿼터스 컴퓨팅, SOA, 임베디드 소프트웨어 등



**박 승 수**

1974년 서울대학교 수학과 졸업(학사).

1976년 한국과학기술원 전산학 졸업(석사).

1988년 미국 텍사스대학 전산학 졸업(박사).

1988년~1991년 미국 캔자스대학 컴퓨터학과 조교수.

1991년~현재 이화여자대학교 컴퓨터공학과 교수.

<관심분야> 인공지능, 시맨틱웹, 온톨로지 등

# 관점지향 프로그래밍을 적용한 실행시간 누설전력 관리 모델 설계<sup>†</sup>

(Design of a management model for runtime leakage power using  
Aspect-Oriented Programming)

김영민<sup>\*</sup>      이찬근<sup>§</sup>  
(Youngmin Kim) (Changun Lee)

**요약** 최근 임베디드 시스템이 발전함에 따라서 저 전력에 대한 요구가 중요한 관심사가 되었다. 하지만 전력 관리 코드가 핵심 관심사 코드와 횡단 결합되어 있기 때문에 가독성과 유지보수성을 저하시키는 원인이 된다. 본 연구에서는 실행시간 동안 발생하는 누설전력을 감소시키기 위한 디바이스 사용에 집중한 전력관리 모델을 제시하고, 이러한 전력관리 모델에 관점지향 프로그래밍을 적용하여 전력관리 코드를 핵심관심사 코드와 분리한다. 마지막으로 제시한 전력관리 모델을 이론적인 방법과 실험적인 방법으로 평가한다.

**키워드** 누설 전력, 태스크 분할, 관점 지향 프로그래밍

**Abstract** The importance of the low-power management has increased due to the recent advances of the embedded systems. However, it is noted that low-power concerns are detrimental to the readability and the maintainability of the codes for the core concerns. In this study, in order to reduce occurring leakage power during run-time, we present a power management while considering the run-time leakage power of devices. the power management codes is separated from the core concern codes by applying aspect-oriented programming. Finally, we analyze the theoretical model of our proposed scheme and present experimental results.

**Key words** leakage power, task division, aspect-oriented programming

## 1. 서론

임베디드 시스템의 성능이 향상되어 감에 따라 전력소비의 폭도 그만큼 늘어나고 있다. 설계자라면 반드시 성능의 향상을 꾀하면서도 배터리의 수명을 고려해야 한다. 이렇듯 저 전력에 대한

요구는 이미 시스템 설계의 중요한 가치 기준이 되었다. 한 예로써, 모바일 장치는 항상 전원이 켜져 있는 상태로 유지되어 진다. 이러한 경우에 많은 곳에서 누설전류가 발생하게 된다. 이 누설 전력은 소모되는 전력에 많은 영향을 미친다. 따라서 배터리 수명 연장에 있어 누설 전력을 낮추는 것은 주요한 관심사라 할 수 있다. 하지만 이러한 관심사를 핵심으로 하여 시스템을 구현하게 되면 실제 임베디드 시스템의 핵심 관심사 코드와 전력관리 코드가 횡단 결합(cross-cutting)되어 코드의 가독성과 유지보수성이 떨어지게 된다. 임베디드 환경과는 달리 사용자 컴퓨팅 환경에서는 오래 전부터

<sup>†</sup> 본 연구는 한국연구재단 기초연구사업(과제번호 20110013924)의 지원을 받았습니다.

<sup>\*</sup> 학생회원 : 중앙대학교 컴퓨터공학과  
remnant1120@gmail.com

<sup>§</sup> 종신회원 : 중앙대학교 컴퓨터공학과 교수, 교신저자  
cglee@cau.ac.kr

논문접수 : 2011년 02월 08일

심사완료 : 2011년 03월 11일

프로그램을 유지보수를 위한 여러 가지 연구 [Jan96] [Na00] [Mey87] [Sel05]가 제안되어 왔다.

기존의 누설전력 관리에 대한 연구에서는 태스크의 실행시간을 제외한 나머지 시간에 디바이스에서 소모하는 전력을 감소시키는 것을 목표로 한다[Sim01][Zha09]. 그리고 전력 관리 코드를 핵심관심사와 분리하는 것이 미흡한 실정이다. 본 연구에서는 임베디드 시스템의 실행시간 누설 전력 관리 모델을 제시함과 동시에 관점 지향 프로그래밍을 적용하여 핵심 관심사 코드와 전력 관리 코드를 분리하였다.

앞으로 2장에서는 횡단 결합된 전력관리 코드를 분리시키는데 사용되는 관점 지향 프로그래밍에 대하여 설명하고, 추가적으로 임베디드 시스템의 주요 가치기준으로 등장한 저 전력 설계를 위한 기법을 언급한다. 3장에서는 본 연구에서 제안하는 관점 지향 프로그래밍을 적용한 실행시간 누설 전력 관리 모델을 제시한다. 4장과 5장에서는 제안하는 전력 관리 모델을 이론적, 실험적으로 평가한다. 마지막으로 본 연구의 결론을 맺고, 향후 연구 방향에 대해 논의하고자 한다.

## 2. 관련 연구

### 2.1 관점 지향 프로그래밍

사용자의 요구가 다양해지고 기술의 빠른 변화가 있는 요즘은 프로그램의 개발에서 유지보수에 대한 비용이 많은 부분을 차지하고 있다. 이러한 이유로 프로그램의 가독성과 유지보수성을 위한 연구가 활발하게 진행되고 있다. 가독성과 유지보수성을 위해 제한된 것이 관점 지향 프로그래밍 [Lad03]이다. 관점 지향 프로그래밍은 프로그래머들이 횡단문제를 모듈화 할 때, 사용하는 프로그래밍 기술이다. 객체 지향 프로그래밍(Object-

Oriented Programming)과 같이 모듈화가 뛰어난 기술도 있지만 보안(security), 메모리 관리(memory management), 로깅(logging)과 같은 부가 기능들은 다른 클래스에 긴밀하게 결합되어 있거나 여러 클래스나 오퍼레이션에 횡단하여 결합되어 있어 모듈화가 어렵다는 데에 그 한계가 있기 때문이다[ Dou01]. 이러한 관점 지향 프로그래밍의 Aspect 코드와 핵심 관심사 코드의 생성부터 실행 코드의 생성까지의 모든 일련의 과정을 그림 1에서 보여 준다.

관점 지향 프로그래밍에 관련된 연구로는 관점 지향 프로그래밍 지원 도구(Aspect J, Aspect C)를 가지고 프로그램에서 모니터링에 관련된 부분은 따로 기술하는 연구 [Gre05] [Sun09] [Yoo10]가 있다. 추가적으로 최적 전력 흐름 시스템 개발을 위해서 관점지향 프로그래밍을 적용한 연구[ Hon07]가 있다. 이는 시스템에 객체 지향 프로그래밍을 사용하기 어려운 경우에 관점 지향 프로그래밍을 사용하여 해결 한다.

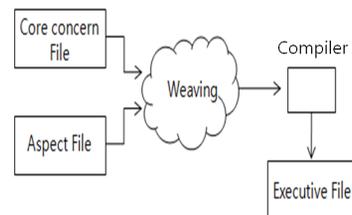


그림 1. 실행 코드의 생성 과정

### 2.2 임베디드 시스템에서의 전력 관리

휴대용 임베디드 시스템에서 배터리가 한정되어 있기 때문에 전력 소비를 줄이기 위한 연구가 이루어지고 있다. 기반이 되는 연구로는 동적 전력 관리 기법과 정적 관리 기법이 존재한다. 동적 전력 관리 기법은 동적 전력 관리(Dynamic Power Management)기법 관한 연구 [Dev08] [Swa05], 동적 전압 조절(Dynamic Voltage Scaling)기법 관한 연구 [Ayd06] [Jej04], 그리고 동적 전력 관리와

동적 전압 조절을 혼합하여 사용하는 기법에 관한 연구[ Sim01 ][ Zha09 ]가 있다. 동적 전력 관리는 결정 시간 이상으로 디바이스가 가동되지 않는 상태에 머무를 경우 디바이스의 전압 상태를 변화시키는 방법이며, 동적 전압 조절은 전압을 동적으로 조절하여 에너지 소모량을 줄이는 방법이다. 이에 반해 수동 전력 관리 기법은 사용자의 요청에 따라 저 전력 모드로 동작하게 하는 것이다. 정적 누설 관리(Static Leakage Management) [ Dro02 ]를 사용하여 대기 또는 전원 차단(Device-off) 모드로 전환한다.

### 3. 관점지향 프로그래밍을 적용한 실행시간 누설전력 관리 모델

#### 3.1 디바이스 모델

디바이스는 두 가지 상태를 지원한다. 실행을 위한 전력이 공급되고 있는 active 상태와 시스템 전력만 공급되는 sleep 상태이다. 각 상태마다 소비되는 전력은 차이를 보이게 된다. 본 연구에서는 각 디바이스를 다음과 같이 정의한다.

정의 1: 디바이스  $D_i = (P_a^i, P_s^i, E_{sd}^i, E_{wu}^i, T_{sd}^i, T_{wu}^i)$  의 각 매개변수는 다음과 같다.

- $P_a^i$ : active 상태에서의 소비전력.
- $P_s^i$ : sleep 상태에서의 소비전력.
- $E_{sd}^i$ : 디바이스의 sleep 상태로 sleep down할 때 발생하는 에너지 소모량.
- $E_{wu}^i$ : 디바이스의 active 상태로 wake up할 때 발생하는 에너지 소모량.
- $T_{sd}^i$  와  $T_{wu}^i$ : 디바이스의 상태 전환 시 소모되는 시간.

break-even time은 디바이스의 상태 변경 시 발생하는 에너지 오버헤드를 보상받기 위해서 디바이스가 sleep 상태로 유지해야하는 최소 시간이다.

정리 1: 디바이스 i의 break-even time(BET<sub>i</sub>)은 다음과 같다[ Zha09 ].

$$BET_i = \frac{E_{sd}^i + E_{wu}^i - T_{sw}^i \cdot P_s^i}{P_a^i - P_s^i}$$

디바이스의 sleep down과 wake up시 소모되는 시간의 합은  $T_{sw}^i = T_{sd}^i + T_{wu}^i$  이다.

#### 3.2 디바이스 사용에 집중한 전력 관리 모델

본 연구에서는 하나의 프로세서를 가진 임베디드 시스템으로 가정한다. 시스템의 요구 사항에 따라서 핵심 관심 사항을 나눈 후 태스크로 할당한다. 그리고 핵심 관심사를 디바이스의 사용에 따라 여러 개의 Job으로 분할한다. Job을 분할 시 Job의 최소 실행시간이 태스크에서 사용하는 디바이스의 가장 긴 BET보다 길어야 한다. 또한 임베디드 시스템의 특성을 감안하여 전력 소모를 줄이기 위해서 실행 중에 있는 Job의 실행이 종료 될 때까지는 다른 Job은 실행되지 않는다. 이것은 디바이스의 실행을 보장함으로써 지연으로 인한 누설 전력을 감소시킨다. 제안하는 시스템 모델은 그림 2에 나타나 있다. Task1은 디바이스 D1과 D2,D3의 사용에 의해서 2개의 Job으로 분할한다. 그리고 D2,D3는 분할하지 않는다. 이것은 최소 실행 조건을 만족하지 않기 때문이다. 마지막으로 task3은 분할되지 않은 것을 볼 수 있다. 이것은 분할 시 Job의 실행시간이 최소 실행 시간을 만족하지 않기 때문이다.

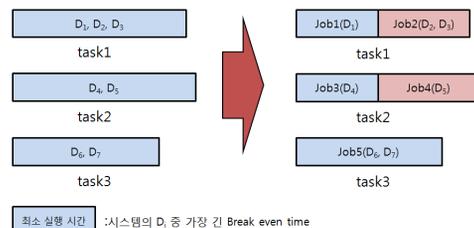


그림 2. 디바이스 관점의 태스크 분할

### 3.3 디바이스 사용에 집중한 전력 관리 모델

전력 관리 모델을 지원하기 위해서 전력 관리 코드들은 도출된 핵심 관심사 내부에 횡단 결합되어있다. 이러한 코드의 가독성과 유지 보수성 향상을 위해서 본 연구에서는 관점 지향 프로그래밍을 적용하여 전력관리 모델을 지원하도록 한다. 관점으로 고려되어야 할 부분으로는 디바이스의 상태전환을 포함한다. 그림 3은 분할 된 태스크에 삽입되어야 할 전력 관리 코드들의 위치를 보여주고 있다.



그림 3. 전력 관리 코드의 삽입 위치

이러한 전력 관리를 위한 관점 지향 프로그래밍 코드는 Job의 시작점과 Job의 종료점을 가리키는 교차점 선언하고, 각 교차점에서 수행될 코드를 포함한다. Job의 시작점에서는 디바이스를 active 상태로 전환하고, Job의 종료점에서는 디바이스를 sleep 상태로 전환한다. 그림 4는 이러한 aspect코드 형태를 보여준다.

```

aspect power management
1  pointcut declare(Job의 시작점)
2  advice 생성
3  {
4      device를 active state로 전환
5  }
6  pointcut declare(Job의 종료점)
7  advice 생성
8  {
9      device를 sleep state로 전환
10 }
    
```

그림 4. aspect 코드 형태

### 4. 이론적 평가

본 절에서는 제시한 전력관리 모델에 따라 예제 태스크를 수행시켰을 경우 소모하는 전력을 계산한다. 그리고 그 결과 값을 분할하지 않고 태스크를 수행하였을 경우에 얻어지는 전력 소모량과 비교한다.

수학적 평가를 위해서는 각 Job에서 소모하는 전력을 계산하기 위한 수식이 필요하다. 현재 수행되고 있는 Job에서 소비하는 전력은 3가지 요소의 합으로 나타내어 질 수 있다. 첫 번째 요소는 프로세서와 Job에서 사용하는 디바이스의 전력 소모이다. 두 번째는 Job이 수행을 시작하고 마칠 때 상태 전환을 위한 오버헤드 에너지이다. 마지막으로 Job이 수행되는 동안 sleep state를 유지하는 디바이스에서 소모하는 전력이다. 이를 정리 2를 통해서 확인 할 수 있다.

정리 2: 각 Job에서 소모한 전력은 다음과 같다.

$$E^m(c_{cur}) = (af^3 + \sum_{i|D_i \in D^n} P_a^i) \frac{c_{cur}}{f} + \sum_{i|D_i \in D^n} (E_{sd}^i + E_{wu}^i) + \sum_{i|D_i \in D-D^n} (\frac{c_{cur}}{f} - BET_i) P_s^i$$

Job이 수행하는 동안 시스템 내의 전력 소모는 ccur은 현재 수행 중인 Job의 실행시간이다. a는 프로세서의 switching capacitance이다.

평가를 진행 할 때 많은 태스크를 생성해서 평가하는 것은 무의미하다. 왜냐하면 태스크를 분할하여 실행 시간 중에 디바이스의 누설전력을 줄이는 것을 목표로 하기 때문이다. 그러므로 본 연구에서는 하나의 태스크만을 가지고 평가를 진행한다. 수행 될 태스크 조건은 다음과 같다.

- 디바이스 D1(0.5, 0.1, 5, 5, 10, 10), D2(0.8, 0.2, 5, 5, 10, 10)를 사용한다.
- 최악 경우 실행시간, WCET는 43ms이다.
- switching capacitance, a는 1이다.
- 프로세서의 구동 주파수, f는 fmax=1이다.

수식과 조건을 가지고 계산한 전력 소모량과 분할하지 않고 태스크를 수행 하였을 경우 계산된 전력 소모량의 차이를 분할된 2개의 Job으로 나누어서 그림 5의 그래프에서 보여준다. 여기서는 분할하지 않은 태스크 수행 시 소모된 전력을 각 Job에 따라 나누어 나타내기 위해서 Job의 수행 시간에 비례하게 나누도록 한다. 그래프를 통해서 각 Job의 계산 결과는 분할하지 않은 태스크 수행 시 보다 제안된 알고리즘에 의해 태스크를 분할하였을 경우 전력 소모가 적다는 것을 확인할 수 있다.

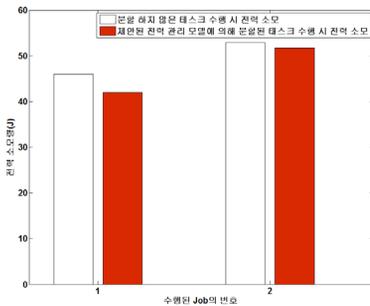


그림 5. 계산 결과 및 전력 비교

### 5. 실험적 평가

실시간성이 배제된 임베디드 시스템을 고려한 간단한 예제를 통해서 제시한 알고리즘을 평가한다. 성능 평가를 위해서 사용한 임베디드 시스템은 ARM7 32bit 마이크로프로세서를 탑재한 레고 마인드스톱이다. 그리고 인텔 코어 2 Duo 3.0GHz를 탑재한 PC에서 모든 프로그램이 작동하고 블루투스2.0을 사용하여 레고 마인드스톱을 제어한다. 마지막으로 PC에서 작동하는 프로그램에 관점지향 프로그래밍을 적용하기 위해서 AspectJ를 사용하였다. 생성된 코드는 임베디드 시스템에 포팅 가능하지 않다.

본 연구에서 사용하는 예제는 모터, 빛 센서, 초음파 센서를 사용하는 1개의 태스크이다. 태스크는 2개의 Job으로 분할된다. Job1은 빛 센서와 초음파센서를 사용하고, Job2는 모터를 사용한다.

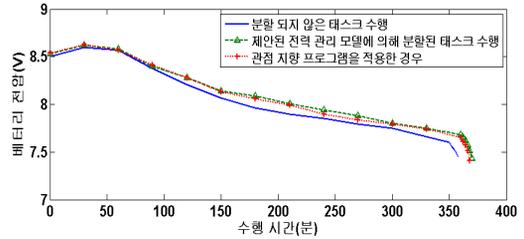


그림 6. 실험 결과

그림 6은 예제 태스크를 분할하지 않고 수행시켰을 경우와 제시한 전력 관리 모델에서 관점지향 프로그래밍을 적용하지 않고 태스크를 수행시켰을 경우 시간에 따른 배터리 전압 강하를 보여준다. 또한 제안된 전력 관리 모델에 관점지향 프로그래밍을 적용하였을 경우 시간에 따른 전압 강하도 보여주고 있다. 그래프를 통해서 제안하는 전력 관리 모델을 사용하였을 경우에 전압이 서서히 감소하는 것과 12분의 수행시간 차이가 있었다는 것을 확인할 수 있다. 이것은 제안한 전력 관리 모델을 사용한 경우가 전력 소모가 적다는 것을 의미한다. 또한 관점 지향 프로그래밍을 적용한 경우에도 많은 오버헤드를 발생시키지 않음을 확인할 수 있다. 이러한 수행 결과들을 표 1에서 보여주고 있다.

표 1. 수행결과 비교

	기존의 전력 관리 방법을 적용한 경우	제안한 전력 관리 방법을 적용한 경우
수행 시간(분)	362	374
평균 전압 강하 (mV/분)	3.525	3.395

## 7. 결론 및 향후연구

임베디드 시스템의 성능 향상과 대규모 임베디드 시스템의 등장으로 인하여 저 전력과 코드 모듈화의 필요성이 증가하게 되었다. 본 연구에서는 임베디드 시스템에서 사용하는 디바이스의 누설전력을 감소시키기 위한 전력 관리 모델을 제안하였다. 그리고 관점 지향 프로그래밍을 적용하여 핵심 관심사 코드로부터 전력관리 코드를 분리하였다. 이로써 임베디드 시스템의 디바이스에서 발생하는 누설전력을 감소시키고, 전력관리 코드를 모듈화 하였다. 하지만 본 연구에서 제안하는 전력관리 모델은 실시간 임베디드 시스템에 적용하기에는 부족하다.

앞으로의 연구에서는 실시간 임베디드 시스템에 적용할 것이다. 또한, 태스크 분할 자동화를 지원하기 위한 기술과 관점 지향 프로그래밍을 적용하였을 때 발생하는 오버헤드에 대한 연구를 추가적으로 진행할 계획이다.

## 참 고 문 헌

- [ Ayd06 ] H. Aydin, V. Devadas, D. Zhu, "System-level energy management for periodic real-time tasks," In Proc. IEEE Real-Time Systems Symposium (RTSS), 2006.
- [ Dou01 ] R. Douence, O. Motelet, M. Südholt, "A Formal Definition of Crosscuts," Proc of the Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns, 2001.
- [ Dro02 ] S. Dropsho, V. Kursun, D.H. Albonesi, S. Dwarkadas, E.G. Friedman, "Managing Static Leakage Energy in Microprocessor Functional Units," Proc. of the IEEE International Symposium on Microarchitecture, MICRO, pp.321, 2002.
- [ Dev08 ] V. Devadas, H. Aydin, "Real-time dynamic power management through device forbidden regions," In Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 2008.
- [ Gre05 ] M. Grechanik, D. E. Perry, D. Batory, "Using aop to monitor and administer software for grid computing environments," In Proc. of COMPSAC'05, Vol. 1, pages 241-248. IEEE, 2005.
- [ Hon07 ] L. M. Honorio, D. A. Barbosa, A.C.Z. Souza, C.V. Lopes, "Intelligent optimal power flow system development using aspect-oriented modeling," IEEE Trans. on Power Systems 22(4), 1826-1834, 2007.
- [ Jan96 ] G. S Jang, Y. S. Moon, C. J. Yoo, O. B Chang, "A Study on Documentation of Object-Oriented Program for Software Maintenance and Reuse," Proc. of the 23th KIISE Spring Conference, vol.35, no.1(A), pp.669-672, 1996.(in Korean)
- [ Jej04 ] R. Jejurikar, R. Gupta, "Dynamic voltage scaling for systemwide energy minimization in real-time embedded systems," In Proc. International Symposium on Low Power Electronics and Design (ISLPED), 2004.
- [ Lad03 ] R. Laddad, "AspectJ in Action," 2nd Ed., p.3, Manning, 2003.
- [ Mey87 ] B. Meyer, "Reusability: The Case for Object-Oriented Design," IEEE Software, vol. 4, no. 2, pp. 50-64, Mar. 1987.

[ Na00 ] H. Y Na, Y. L Choi, S. R Rhew, “A Case Study of Software Maintenance by Database Reverse,” Proc. of the 27th KIISE Spring Conference. vol.27, no.1(A), pp.588-590, 2000.(in Korean)

[ Sel05 ] R. W. Selby. “Enabling reuse-based software development of large-scale systems,” IEEE Trans. Softw. Eng., 31(6):495-510, 2005.

[ Sim01 ] T. Simuinic, L. Benini, A. Acquaviva, P. Glynn, G. D. Micheli, “Dynamic voltage scaling and power management for portable systems,” In Proc. Design Automation Conference (DAC), 2001.

[ Swa05 ] V. Swaminathan, K. Chakrabarty, “Pruning-based, energy-optimal, deterministic i/o device scheduling for hard real-time systems,” ACM Trans. on Embedded Computation Systems, 4(1), 2005.

[ Yoo10 ] G. J. Yoo, E. S. Lee, “Monitoring Methodology using Aspect Oriented Programming in Functional based System,” Proc. of the IEEE International Conference on Advanced Communication Technology, ICACT, Fed 2010.

[ Zha09 ] B. Zhao, H. Aydin, “Minimizing expected energy consumption through optimal integration of dvs and dpm,” in ICCAD, 2009.

**저 자 소 개**



**김 영 민**

2010년 중앙대학교 전자전기 공학부 학사  
2010년~현재 중앙대학교 컴퓨터 공학과 석박통합과정.

<관심분야> 실시간 소프트웨어, 전력 관리 알고리즘



**이 찬 근**

1996년 중앙대학교 전자계산학과 학사  
1998년 KAIST 전산학과 석사.  
2005년 Univ. of Texas at Austin 전산학과 박사.  
2005년~2007년 미국 인텔 소프트웨어 엔지니어.  
2007~현재 중앙대학교 컴퓨터공학부 조교수.

<관심분야> 실시간 소프트웨어, 수행시간 모니터링, 소프트웨어 테스트





## 회원 가입 안내 및 회비 납부 요령



한국정보과학회 소프트웨어공학소사이어티는 회원 여러분에게 유익한 정보를 제공해 드리기 위하여 보다 충실한 내용의 논문지 발간 배포, 그리고 국제·국내 학술발표회 및 초청강연회와 단기강좌 등의 여러 가지 사업들을 추진하고 있습니다.

소프트웨어공학 소사이어티의 가입을 통해 정보 및 기술 교류, 그리고 인적 네트워크의 구성에 참여하시기를 기대합니다. 회원 가입을 위하여 아래의 회비 안내를 참고하시어 회비를 납부하시고, 다음 쪽의 입회원서를 작성하시어 아래 소프트웨어공학소사이어티 주소로 보내주시거나 팩스 또는 이메일을 통해 보내어 주시기 바랍니다.

한국정보과학회 소프트웨어공학소사이어티 연락처는 아래와 같습니다.

### ◆ 소프트웨어공학소사이어티

주 소 : (우) 121-742 서울시 마포구 신수동 1번지, 서강대학교 신과학관 202호  
한국정보과학회 소프트웨어공학소사이어티 박수용

전 화 : (02) 705-8928

팩 스 : (02) 704-8273

전자우편 : sypark@sogang.ac.kr (박수용교수), bjlee@uos.ac.kr (이병정교수)

홈페이지 : <http://www.sigse-kiss.or.kr/>

### ◆ 회비 안내

회원구분	• 학생회원 : IT 분야 학과 또는 관심 있는 학생 • 정회원, 종신회원 : IT 분야 종사자
가입비	학생회원, 정회원 : 20,000원, 종신회원 : 200,000원
년회비	학생회원, 정회원 : 20,000원

### - 회비납부 방법

- (1) 무통장입금 또는 계좌이체 후 입회원서 발송  
: 계좌 번호 : 제일은행 150-20-358028 (이병정)
- (2) 소사이어티 주관 학술행사 개최시, 행사장 당일 가입 및 납부 가능



# 개인회원용 입회원서



회원구분	학생회원 ( ) 정회원( ) 종신회원( )							
성명	한글			생년월일				
	영문							
연락처	직장전화			휴대전화				
	e-mail							
주소	직장명/ 부서			직급				
	직장주소	(우)						
학력	학사	년	월	-	년	월	대학교	과
	석사	년	월	-	년	월	대학원	과
	박사	년	월	-	년	월	대학원	과
관심분야								

본인은 한국정보과학회 소프트웨어공학소사이어티의 취지에 찬성하여 회원으로 가입하고자 이에 입회원서를 제출합니다.

년 월 일

신청인: (인)

한국정보과학회 소프트웨어공학 소사이어티 회장 귀하



## 논문지 논문 모집 (Call for Papers)



한국정보과학회 소프트웨어공학 소사이어티에서는 매년 4회에 걸쳐 ‘소프트웨어공학 소사이어티 논문지’를 발간하고 있습니다. 이 논문지에는 소프트웨어공학 전반에 걸친 연구논문과 산업계 논문을 게재해 오고 있습니다. 다음과 같은 소프트웨어공학 주제에 관련된 논문을 모집하고 있으니 학계와 산업계의 여러분들의 적극적인 논문투고를 바랍니다.

### ◆ 논문 주제

- 소프트웨어 설계 및 아키텍처
- 소프트웨어 재사용 및 프로덕트라인
- 요구공학
- 소프트웨어 품질 및 테스트
- 관리 및 프로세스
- 소프트웨어 정형 기법
- 서비스기반 소프트웨어 개발
- 임베디드, 모바일, 웹기반 소프트웨어 개발
- 기타 소프트웨어 응용 (국방, 자동차, 조선 등의 분야)

### ◆ 논문심사

- 투고된 논문은 편집위원회에서 심사 선정하며, 필요 시 외부 심사위원을 위촉하여 심사를 합니다. 제출된 논문은 반환하지 않습니다.
- 심사료 및 게재료: 없음

### ◆ 논문 제출

- 소프트웨어공학 소사이어티의 논문지 투고 양식(<http://www.sigse-kiss.or.kr/>)을 사용하며, 논문의 분량은 10장으로 제한합니다.
- 논문지 투고규정에 따라 작성된 심사용 논문파일은 온라인투고시스템을 통하여 투고하시기 바랍니다.

### ◆ 문의처 (편집위원회)

- 편집이사 : 고인영 교수 (KAIST, 042-350-3547, iko@kaist.ac.kr)
- 편집이사 : 윤희진 교수 (협성대학교, 031-299-0841, hjyoon@uhs.ac.kr)
- 편집위원 : 강성원 교수 (KAIST, 042-350-3512, sungwon.kang@kaist.ac.kr)
- 편집위원 : 김문주 교수 (KAIST, 042-350-3543, moonzoo@cs.kaist.ac.kr)
- 편집위원 : 김정아 교수 (관동대학교, 033-649-7801, clara@kd.ac.kr)
- 편집위원 : 이우진 교수 (경북대학교, 053-950-6378, woojin@knu.ac.kr)
- 편집위원 : 이찬근 교수 (중앙대학교, 02-820-5829, cglee@cau.ac.kr)
- 편집위원 : 이 근 박사 (삼성전자, 031-277-7323, gskeun@gmail.com)
- 편집위원 : 이세영 박사 (정보통신산업진흥원, 02-2132-1323, sarahlee230@gmail.com)



## 투 고 요 령



1. 소프트웨어공학소사이어티 논문지에 실리는 원고는 주제 논문, 일반 논문, 산업체 기고 등으로 구분하며 다음과 같은 분야에 대하여 모집한다.
  - 가. 소프트웨어공학 및 그 응용분야에 대한 연구결과
  - 나. 강좌 및 관련 교육사항 소개 (목적, 과정, 일정, 대상, 특징)
  - 다. 소프트웨어 도구 및 방법론 소개 (가격, 특징, 종류, 적용사례)
  - 라. 소프트웨어 산업에 대한 학계, 업계의 주요 관심사
  - 마. 기타 관련 사항
2. 투고자는 원칙적으로 본 소사이어티의 회원으로 한다. 다만 공동 또는 초청 기고자는 예외로 한다.
3. 논문은 원칙적으로 한글로 작성한다.
4. 원고는 한글(hwp), 워드(MS Word), PDF 형식 중 하나를 택하여 A4용지에 작성하며, 그림과 표를 포함하여 10쪽 이내로 한다.
5. 논문 내용에 직접 관련이 있는 문헌에 대해서는 이들 문헌에 관련이 있는 본문 중에 참고 문헌 번호를 쓰고 그 문헌을 참고문헌 난에 인용 순서대로 기술한다. 참고문헌은 학술지의 경우 저자, 제목, 학술지명, 권, 호, 쪽수, 발행 연도의 순서로, 단행본은 저자, 서명, 쪽수, 발행처, 발행 연도의 순서로 기술한다.

[1]Cole, R., "Parallel Merge Sort", SIAM Journal of Computing, vol.17, No.4, pp.770-785, 1988.  
[2]김수형, 강명호, 조형재, 송주석. "안전하고 효율적인 침입자 역추적 시스템", 정보과학회논문지, 제25권, 제10호, pp.1123-1131, 1998
6. 논문은 소프트웨어공학 소사이어티(<http://www.sigse-kiss.or.kr/>)의 온라인 투고 시스템을 통해 제출한다.
7. 논문투고신청서를 반드시 작성하여 이메일([hjyoon@uhs.ac.kr](mailto:hjyoon@uhs.ac.kr))로 제출한다.
7. 원고 접수는 수시로 하며, 접수일은 온라인 접수일로 한다.
8. 기타 자세한 사항은 한국정보과학회 논문지 투고 요령을 따른다.



## 한국정보과학회 소프트웨어공학소사이어티 임원명단

직책	성명	소속	
회 장	박 수 용	서강대학교	
부 회 장	권 기 현	경기대학교	
부 회 장	민 경 호	LG전자	
부 회 장	전 진 옥	비트컴퓨터	
부 회 장	최 병 주	이화여자대학교	
부 회 장	한 혁 수	상명대학교	
감 사	이 금 해	항공대학교	
	차 성 덕	고려대학교	
자 문 위 원 회	강 교 철	포항대학교	
	권 용 래	KAIST	
	배 두 환	KAIST	
	성 기 수	KISTI 고문	
	신 규 상	ETRI	
	양 승 민	송실대학교	
	우 치 수	서울대학교	
	이 경 환	중앙대학교	
	이 단 형	KAIST	
	정 기 원	송실대학교	
	황 선 명	대전대학교	
	협 력 기 관 위 원 회	권 경 룡	국방품질기술원
박 찬 규		국방SW산학연합회	
신 석 규		SW시험인증센터	
유 법 민		지식경제부	
윤 태 권		SW기술진흥협회	
이 상 은		한국소프트웨어진흥원	
운 영 위 원 회	운 영 위 원 장	권 기 현	경기대학교
	총 무 이 사	이 병 정	서울시립대학교
	기 획 이 사	백 종 문	KAIST
	조 직 이 사	이 관 우	한성대학교
	학 술 이 사	홍 장 의	충북대학교
	편 집 이 사	고 인 영	KAIST
	편 집 이 사	윤 회 진	협성대학교
	협 력 이 사	인 호	고려대학교
기 술 위 원 회	홍 보 이 사	이 정 원	아주대학교
	기 술 위 원 장	최 병 주	이화여자대학교
	요 구 공 학 분 과 장	박 수 진	서강대학교
	아 키 텍 처 분 과 장	조 은 숙	서일대학
	재 사 용 분 과 장	이 관 우	한성대학교
	정 형 화 분 과 장	김 문 주	KAIST
	테 스 팅 분 과 장	김 영 철	홍익대학교
프 로 세 스 분 과 장	이 세 영	NIPA	

	직책	성명	소속
산 학 위 원 회	고신뢰성분과장	배현섭	슈어소프트테크
	재공학분과장	김진태	SEEG
	실시간분과장	이찬근	중앙대학교
	산학위원회장	전진옥	비트컴퓨터
	자동차분과위원장	유영수	오토에버시스템(주)
	국방분과위원장	윤희병	국방대학원
	금융분과위원장	김정아	관동대학교
	전자분과위원장	이근	삼성전자
	의료분과위원장	최호진	KAIST
전력분과위원장	이장수	한국원자력 연구소, MMIS팀 책임연구원	
이사	강성원	KAIST	
이사	계승교	삼성SDS	
이사	권원일	STA컨설팅	
이사	남영광	연세대학교	
이사	김문주	KAIST	
이사	김상기	현대자동차	
이사	김영철	홍익대학교	
이사	김정아	관동대학교	
이사	민상윤	솔루션링크	
이사	박복남	핸디피엠지	
이사	백종문	KAIST	
이사	손세창	인천공항공사	
이사	손진규	삼성탈레스	
이사	양상욱	KAI	
이사	염근혁	부산대학교	
이사	오재원	카톨릭대학교	
이사	유준범	건국대학교	
이사	윤형진	케피코	
이사	윤희병	국방대학원	
이사	이근	삼성전자	
이사	이병걸	서울여자대학교	
이사	이성남	방위사업청	
이사	이우복	삼성전자	
이사	이우진	경북대학교	
이사	이은주	경북대학교	
이사	이정원	아주대학교	
이사	이해서	솔루션링크	
이사	장주수	모아소프트	
이사	정연대	N3SOFT	
이사	정인상	한성대학교	
이사	조병인	국방과학연구소	
이사	조상윤	다한테크	
이사	최승훈	덕성여자대학교	
이사	최종무	단국대학교	
이사	현창문	탐라대학교	



## 2010-2011 소프트웨어공학소사이어티 논문지 편집위원회 ...

편집위원장 윤희진 교수(협성대학교)

편 집 위 원 고인영 교수(KAIST)

강성원 교수(KAIST)

김문주 교수(KAIST)

김정아 교수(관동대학교)

이우진 교수(경북대학교)

이찬근 교수(중앙대학교)

이 근 박사(삼성전자)

이세영 박사(정보통신산업진흥원)



## 소프트웨어공학소사이어티 논문지 제24권 제1호 (통권 89호) ...

발 행 일 || 2011년 5월 31일

발 행 인 || 박수용

편 집 인 || 윤희진

발 행 처 || 사단법인 한국정보과학회 소프트웨어공학소사이어티

연 락 처 || 서울특별시 마포구 신수동 1번지, 서강대학교 신과학관 202호

전 화 : 02-705-8928, 팩 스 : 02-704-8273

홈페이지 : <http://www.sigse-kiss.or.kr/>

인 쇄 처 || (주)참기획 (전화 : 042-861-6380, 팩스 : 042-861-6381)

Copyright© 2011 한국정보과학회 소프트웨어공학소사이어티(비매품)