

## 확장된 UML 상호작용 다이어그램을 이용한 이동 에이전트의 이동성 표현

유 문 성\*

### *Mobility Representation of Mobile Agents using Extended UML Interaction Diagrams*

Yoo, Moon Sung

#### 〈Abstract〉

A mobile agent system is the most common software paradigm for distributed applications. A software development model to develop efficiently for these systems is needed. Unified Modeling Language (UML) is a widely used software development model to develop a system for Visualizing, Specifying, Constructing, Documenting. However, UML can not describe in explicit way the necessary mobility of the mobile agent-based software systems. In this paper, the interaction diagrams of UML (sequence diagrams and communication diagrams) are extended and used to express the mobility of the mobile agents in three ways. For a case study, we applied the extended diagrams to a distributed file searching using mobile agents, and showed these diagrams can describe the mobility very well.

Key Words : Mobile Agents, UML, Language Extensions, Interaction Diagrams

## I. 서론

이동 에이전트는 한 기계에서 다른 기계로 이동할 수 있는 소프트웨어로 때와 장소를 선택하여 이동할 수 있다[1-2]. 이동 에이전트는 인코딩된 연산이 데이터가 있는 곳에 이동하기 때문에 데이터를 가져와 계산하는 것보다 상당히 네트워크 트래픽을 줄인다. 이동 에이전트는 다양한 네트워크 환경과 응용프로그램의 출현에 대해, 플랫폼 독립성을 지니면서 또한 통합성과 유연성이 뛰어나, 분산응용 분야에 있어서 유용한 기술의 하나로 기대되고 있다. 이동 에이전트 시스템은 이동성을 기반으로 작업을 수행하므로 플랫폼 독립적인 언어와 이동성

을 제어, 관리하는 환경을 기본으로 요구한다. 이동 에이전트는 그 응용의 폭을 넓혀가고 있어, 이를 효율적으로 개발하기 위한 도구가 필요하다.

UML 기법은 기존의 모델링 기법인 Booch 방법론, OMT, Jacobson 방법론을 집대성하여 향상시킨 객체지향 분석/설계방법이다[3]. UML 기법은 객체지향 모델링이 가져야 할 필수 기능으로 일반화된 표기법을 지원하고 모델링 요소간의 관계를 명시해 주는 등의 기준을 잘 갖추고 있다.

UML에서 제공하는 많은 다이어그램 중 이동 에이전트의 이동성을 가장 잘 표현할 수 있는 것이 상호작용 다이어그램(Interaction Diagram)이다. 상호작용 다이어그램에는 순차 다이어그램(Sequence Diagram)과 통신

\* 상지대학교 컴퓨터정보공학부 교수(교신저자)

다이어그램(Communication Diagram)이 있다.

본 논문에서는 이동 에이전트의 이동성을 효율적으로 표현하기 위하여 상호작용 다이어그램을 세가지 방법으로 확장하였다. 그리고 확장된 다이어그램들을 이동 에이전트를 이용한 분산형 파일 검색에 적용하여 효율적으로 나타낼 수 있음을 보였다.

## II. 관련 연구

### 2.1 이동 에이전트

인터넷의 급속한 보급과 발달은 정보량의 증가를 가져왔고 이는 사용자를 대항할 에이전트에 대한 관심을 높이게 하였다. 에이전트는 특정 목적을 수행하기 위하여 사용자를 대신하여 작업을 수행하는 자율적인 프로세스이다. 에이전트가 가지는 특성 중에는 자율성(Autonomy), 지능성(Intelligence), 이동성(Mobility), 사교성(Social Ability) 등이 있다[4].

이동 에이전트는 에이전트의 하나로서 다른 에이전트와 달리 자신이 직접 네트워크를 통해 이동하면서 실행되는 프로그램의 일종이다[5-6]. 이동 에이전트는 에이전트의 특성 중 특히 이동성이 강조된 에이전트이며 에이전트 스스로의 판단에 의해 네트워크간의 호스트들을 이동하면서 사용자를 대신하여 작업을 수행한다. 이동 에이전트는 원격 통신, 네트워크 관리, 전자상거래, 이동 컴퓨팅 등의 분야에서 활용된다.

이동 에이전트는 현재의 네트워크에서 자동으로 실행될 수 없고 이동 에이전트를 위한 플랫폼이 각 호스트마다 설치되어 있어야 하는데 Aglet[5], JADE[7], Mobile-C[8]등이 대표적인 플랫폼이다. 이동 에이전트 사용이 증가하고 있으므로 이를 효율적으로 개발하기 위한 모델이 필요하다.

### 2.2 UML

UML(Unified Modeling Language)은 사실상 소프트웨어 산업 표준으로 시각화, 사양화, 구축, 시스템의 요소를 문서화하는 모델링 언어이다. UML은 문법과 의미가 잘 정의되어 있으며 요구 사항 수집에서부터 소프트웨어 구성 요소의 배포까지 객체 지향 소프트웨어 시스템을 개발하는데 하향식 정련기법을 사용하며 이를 위하여 많은 그래픽 도구들을 제공한다.

UML 다이어그램은 그림에 의해 표현되는 클래스, 개체, 노드, 구성 요소 및 관계와 같은 모델 요소를 포함한다[3, 9-10]. UML 다이어그램에는 요구 분석 과정에서 시스템과 외부와의 상호 작용을 묘사하는 사용사례(Use Case) 다이어그램, 시스템의 구조적인 모습을 그리는 클래스(Class) 다이어그램, 업무의 흐름을 모델링하거나 객체의 생명 주기를 표현하는 활동(Activity) 다이어그램, 객체 간의 메시지 전달을 시간적 흐름에서 분석하는 순차(Sequence) 다이어그램, 객체와 객체가 주고받는 메시지 중심으로 작성한 동적 다이어그램인 통신(Communication) 다이어그램(또는 협력(Collaboration) 다이어그램), 소프트웨어 구조를 나타내는 컴포넌트(Component) 다이어그램, 기업 환경의 구성과 컴포넌트들 간의 관계를 그린 배포(Deployment) 다이어그램 등이 있다. 이 중 순차 다이어그램(Sequence Diagram)과 통신 다이어그램(Communication Diagram)을 합쳐 상호작용 다이어그램(Interaction Diagram)이라고 하며 이동 에이전트의 이동성을 가장 잘 표현할 수 있는 다이어그램이다.

### 2.3 기존의 연구 동향

UML을 이동 에이전트를 비롯한 에이전트에 적용하려는 몇 가지 연구가 있었다. 먼저 일반적인 에이전트를 위한 전반적인 UML을 확장된 연구로는 [11]이 있는데, 여기서는 에이전트에 관한 다이어그램을 3가지 level로 나누어서 표현하였다. 주로 순차(Sequence) 다이어그램, 협력(Collaboration) 다이어그램, 활동(Activity) 다이어그

램, 패키지(Package) 다이어그램, 배포(Deployment) 다이어그램을 에이전트에 맞게끔 확장하였으나 이동 에이전트의 이동성에 대한 것은 고려하지 않았다.

이동 에이전트에 대하여 전반적인 UML 다이어그램 확장을 연구한 것이 [12]이다. 이 연구에서는 전반적인 UML 다이어그램, 즉 사용사례(Use Case) 다이어그램, 클래스(Class) 다이어그램, 활동(Activity) 다이어그램, 순차(Sequence) 다이어그램, 협력(Collaboration) 다이어그램, 컴포넌트(Component) 다이어그램, 배포(Deployment) 다이어그램을 이동 에이전트에 맞게끔 확장하였으나 대부분 이동 에이전트에 관계된 개체(클래스, 사용사례, 패키지 등)에 모바일을 뜻하는 M을 표시하는 것으로 확장했다. 이 연구에서는 위치에 대한 것이 명시적으로 나타지지 않아 특히 상호작용 다이어그램(즉 순차 다이어그램 (Sequence Diagram)과 통신 다이어그램(Communication Diagram))에서 현재 이동 에이전트가 어느 위치(노드)에서 작동하는가를 파악하기 어렵다. 이동 에이전트에 대한 활동 다이어그램 확장에 대한 연구로는 [13-14]이 있다.

이동 에이전트뿐만 아니라 모바일 하드웨어를 포함한 일반적인 이동 객체(mobile objects)에 대한 순차 다이어그램을 제시한 연구는 [15]이다. 여기서는 위치 정보를 표현하기 위하여 객체의 한 구성요소로 태그를 사용하여 이동 객체의 위치를 표현하였으며 이동 객체와 관련된 <<create>>, <<copy>>, <<become>> 등 여러 스테레오타입과 중첩된 구조를 사용하였다. 이 연구에서는 표준적인 객체 표현과 순차 다이어그램을 사용하지 않고 중첩된 구조를 가지는 등 이동 에이전트를 위한 다이어그램 으로서는 지나치게 복잡하고 비효율적이다.

### III. 이동성 표기를 위한 확장된 상호작용 다이어그램

이동 에이전트의 이동성과 관련한 상호작용 다이어그

램에서는 다음과 같은 요소를 나타낼 수 있어야 하는데 원래의 UML 다이어그램 표기법으로서는 표현할 수 없었다.

- 이동에이전트의 생성
- 이동 에이전트의 명시 및 위치
- 이동에이전트의 이동 경로

기존의 관련된 연구[13-16]에서는 이를 표현하기 위하여 여러 가지 스테레오 타입을 사용하였는데 본 연구에서는 이동 에이전트를 생성할 때는 <<create>> 스테레오 타입을 사용하고, 이동 에이전트의 명시는 <<mobile agent>> 스테레오 타입을 사용하여 나타내며, 이동 에이전트의 이동 경로는 화살표를 사용하며 <<move>> 스테레오 타입 메시지로 표현한다.

또한 기존의 연구에서는 에이전트나 객체의 위치에 대한 표현이 미흡하였는데 에이전트의 위치를 나타내는 것은 다음과 같은 이유에서 모바일 에이전트의 모델링에서는 필수적으로 명시되어야 한다.

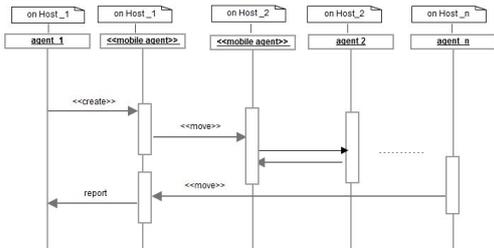
첫째, 이동 에이전트는 정적 에이전트와 달리 여러 노드(혹은 호스트)에서 실행해야 되는데 각 노드마다 환경이 상이할 수 있음을 전제로 해야 한다. 노드마다 하드웨어, 운영체제, 사용가능 서비스와 자원이 상이하므로 이동 에이전트가 어느 위치에 있는가를 명시적으로 표시해야 한다[17]. 둘째, 이동 에이전트는 각 노드의 지역 에이전트와 교류해야 하므로 현재 이동 에이전트가 교류 가능한 에이전트를 알기 위해서 이동 에이전트의 현재 위치를 반드시 알아야 한다. 셋째, 네트워크 관리자는 이동 에이전트가 현재 어느 노드에 있는가를 파악하고 있어야 하며 이는 이동 에이전트가 주어진 작업을 마치지 못하고 오류가 났을 때 그것을 복구하기 위해서도 이동 에이전트의 위치는 반드시 명시되어야 한다. 본 연구에서는 이동 에이전트의 위치를 노트(note)나 상태(state)를 이용해 표시한다.

스테레오 타입과 노트(note)나 상태(state)를 이용해

각 에이전트의 위치를 명시하여 이동성을 효율적으로 표기하기 위한 확장된 상호작용 다이어그램으로 세 가지 방식을 제안한다. 첫째는 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 순차 다이어그램, 둘째는 이동 에이전트의 위치를 상태(state)로 나타낸 확장된 순차 다이어그램, 셋째는 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 통신 다이어그램을 제시한다.

### 3.1 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 순차 다이어그램

이동 에이전트의 이동성을 순차 다이어그램에 나타내기 위해 먼저 이동 에이전트가 지금 어느 호스트에 있는가를 표시해야 하는데 UML의 노트를 사용하여 각 이동 에이전트의 위치를 표시한다. 이동 에이전트는 스테레오 타입을 사용하여 <<mobile agent>>로 표현하며 이동 에이전트가 한 호스트에서 다른 호스트로 이동할 때 사용할 메시지로 스테레오 타입을 사용한 <<move>> 메시지를 이용한다. 이 표현을 사용하면 순차 다이어그램은 <그림 1>과 같은 형식으로 표현된다.

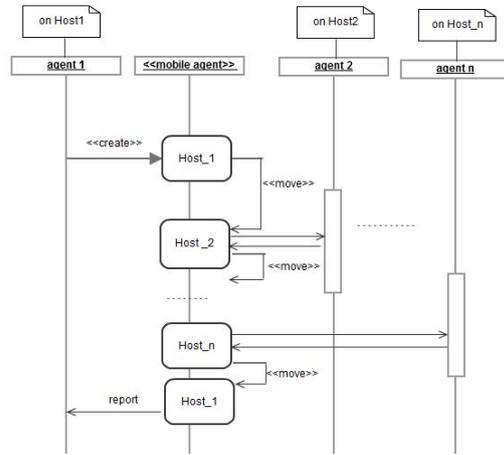


<그림 1> 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 순차 다이어그램

### 3.2 이동 에이전트의 위치를 상태(state)로 나타낸 확장된 순차 다이어그램

이동 에이전트가 지금 어느 호스트에 있는가를 표시하기 위하여 이동 에이전트의 상태에 호스트 위치를 상

태를 사용하여 나타낸다. 이동 에이전트는 스테레오 타입을 사용하여 <<mobile agent>>로 표현 한다. 이동 에이전트는 첫 번째 방법과 달리 하나의 객체로 표현되며 이동할 때 상태가 바뀌는 것으로 표현되며 이를 스테레오 타입을 사용한 <<move>> 메시지를 이용하여 표현한다. 이동 에이전트가 아닌 고정 에이전트나 객체는 노트로 어느 위치에 있는지를 나타낸다. 이 표현을 사용하면 순차 다이어그램은 <그림 2>와 같은 형식으로 표현된다.

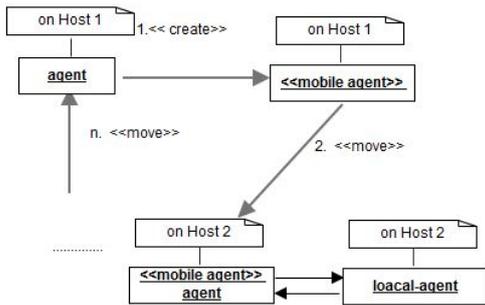


<그림 2> 이동 에이전트의 위치를 상태(state)로 나타낸 확장된 순차 다이어그램

### 3.3 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 통신 다이어그램

노트(note)와 스테레오 타입을 이용한 확장된 순차 다이어그램과 같이 지금 어느 호스트에 있는가를 표시해야 하는데 UML의 노트를 사용하여 각 이동 에이전트의 위치를 표시한다. 이동 에이전트는 스테레오 타입을 사용하여 <<mobile agent>>로 표현하며 이동 에이전트가 한 호스트에서 다른 호스트로 이동할 때 사용할 메시지로 스테레오 타입을 사용한 <<move>> 메시지를 이용하는 것은 첫 번째 방법과 같으며 메시지 시행 순서는 메시지 앞의 번호에 따라 시행된다. 같은 위치(노트 또는

호스트)에 있는 에이전트들은 바로 옆에 표시하는 것이 전체적인 구조를 파악하는데 도움이 된다. 이 표현을 사용하면 확장된 통신 다이어그램은 <그림 3>과 같은 형식으로 표현된다.



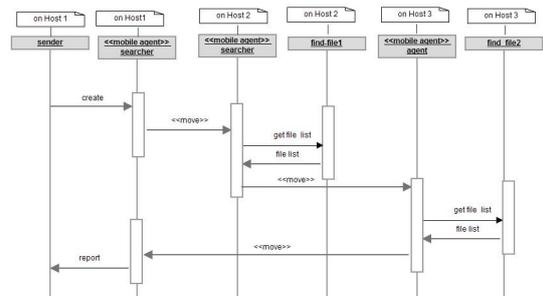
<그림 3> 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 통신 다이어그램

#### IV. 사례 연구

분산형 파일 검색은 호스트 목록에 있는 모든 호스트들을 방문하여 우리가 찾는 조건에 맞는 파일이 있는가를 조사한다. 본 논문에서는 Host1에서 이동 에이전트를 파견하여 Host2, Host3를 방문하는 것으로 한다. 에이전트는 점검해야 할 네트워크상의 호스트들의 주소 목록을 지니고, 각 호스트를 방문하여 점검하고 그 결과를 관리자에게 넘겨주게 되고, 그 결과를 바탕으로 관리자는 분산형 파일 검색을 찾아낸다. 각 호스트에는 파일의 목록을 검색할 수 있는 지역 에이전트가 있다. 이동 에이전트는 각 호스트로 이동한 다음 각 호스트의 지역 에이전트로부터 조건에 맞는 그 호스트의 파일 목록을 얻는다. 방문할 호스트 목록 순서에 따라 다음 호스트를 방문하고 동일한 작업을 계속한 후 모든 호스트에 대한 검색이 끝났으면 원래의 호스트로 돌아와 그 결과를 보고한다.

<그림 4>는 노트(note)와 스테레오 타입을 이용한 확장된 순차 다이어그램을 보여준다. 먼저 필요한 에이전

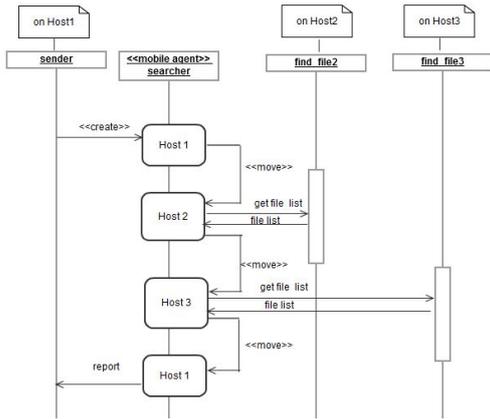
트(또는 객체)들을 조사하고 각 에이전트들이 어디에 위치하는가를 파악한 다음 각 에이전트들이 위치하는 호스트(또는 노드)별로 순서대로 나열한다. Host1에서 sender가 이동 에이전트를 생성하고 방문할 네트워크상의 호스트들의 주소 목록을 전달하면 이동 에이전트는 목록상의 처음 호스트인 Host2로 <<move>> 메시지를 사용하여 이동하고 그 호스트상의 지역 에이전트(또는 객체) find\_file2를 호출하여 조건에 맞는 파일을 검색한 다음 결과를 얻는다. 그리고 다음 호스트인 Host3로 이동하여 동일한 작업을 계속하고 모든 호스트 방문이 끝났으면 자기의 기본 플랫폼으로 돌아가서 결과를 보고한다.



<그림 4> 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 순차 다이어그램을 이용한 분산형 파일 검색

<그림 5>는 상태(state)와 스테레오 타입을 이용한 확장된 순차 다이어그램을 보여준다. 먼저 필요한 에이전트들을 조사하고 각 에이전트들이 어디에 위치하는가를 파악한 다음 각 객체들이 위치하는 호스트별로 순서대로 나열한다. 단 여기서 이동 에이전트는 <그림 4>와 같이 각 호스트별로 따로 그리지 않고 이동 에이전트의 생명선에 방문한 순서대로 상태로서 호스트를 나타낸다. 생성된 이동 에이전트는 상태로 표시한 Host1에서 처음 방문할 호스트인 Host2 상태로 <<move>> 메시지를 사용하여 이동하고 Host2 상태로 이동한 이동 에이전트는 Host2의 지역 에이전트(또는 객체) find\_file2를 호출하여 조건에 맞는 파일을 검색한 다음 결과를 얻는다. 그리고 다음 호스트인 Host3으로 이동하는 것도 상태로 표시하

여 동일한 작업을 계속하고 모든 호스트 방문이 끝났으면 자기의 기본 플랫폼인 Host1 상태로 돌아가서 결과를 보고한다.

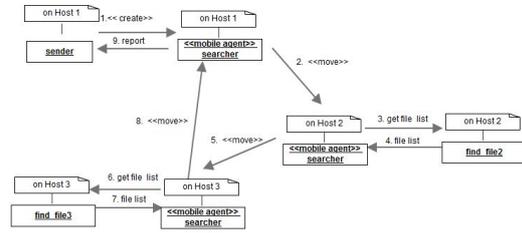


<그림 5> 이동 에이전트의 위치를 상태(state)로 나타낸 확장된 순차 다이어그램을 이용한 분산형 파일 검색

<그림 6>은 스테레오 타입을 이용한 확장된 통신 다이어그램을 보여준다. 먼저 필요한 에이전트들을 조사하고 각 객체들이 어디에 위치하는가를 파악한 다음 각 에이전트들이 위치하는 호스트별로 분류하여 근접한 곳에 위치시킨다. 이동 에이전트가 이동할 때 마다 다른 객체로 표시하고 작동하는 방식은 첫 번째 방식과 같다. 이 다이어그램은 시스템의 전체적인 구조를 파악하는데 유용하다.

## V. 결론

본 논문에서는 이동 에이전트의 이동성을 UML 다이어그램으로 효율적으로 표현하려 하였으며 이동 에이전트의 이동성에 중점을 두었다. UML의 여러 다이어그램 중 이동성은 상호작용 다이어그램이 가장 잘 표현할 수 있기 때문에 여기에 이동성을 나타낼 수 있는 표기법을 추가하였다.



<그림 6> 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 통신 다이어그램을 이용한 분산형 파일 검색

본 논문에서는 이동 에이전트의 이동성을 상호작용 다이어그램을 확장된 표기법에 기반하여 세 가지 방법으로 제안하였다. 첫째는 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 순차 다이어그램이다. UML의 노트를 사용하여 각 이동 에이전트의 위치를 표시하며 시간에 따라 이동 에이전트들이 어떻게 이동하는가를 잘 보여준다.

둘째는 이동 에이전트의 위치를 상태(state)로 나타낸 확장된 순차 다이어그램이다. 이동 에이전트의 상태에 호스트 위치를 명기한 상태를 사용하여 이동 에이전트가 어떻게 이동하는가를 그 이동 에이전트의 상태의 변화를 점검함으로써 파악할 수 있다.

셋째는 이동 에이전트의 위치를 노트(note)로 나타낸 확장된 통신 다이어그램이다. 첫번째 방법과 같이 UML의 노트를 사용하여 각 이동 에이전트의 위치를 표시하며 전체적인 구조를 파악하는데 유용하다.

사례 연구로 분산형 파일 검색을 제안한 세 가지 방법으로 이동성을 표현하였다. 사례 연구 결과 제안한 이동 에이전트의 이동성 표기를 위한 확장된 상호작용 다이어그램은 분산형 파일 검색의 이동성을 효율적으로 표현할 수 있었다.

## 참고문헌

- [1] A. Genco, "Mobile agents: principles of operation and applications," MIT Press, Boston, 2008, pp. 21-24.
- [2] D. Lange, M. Oshima, "Seven good reasons for mobile agents," *Communications of the ACM* 42 (3), 1999, pp. 88-89.
- [3] G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*(2nd ed.), Addison-Wesley, Reading, MA, 2005, preface p.18.
- [4] T. Magedanz, K. Rothermel, S. Krause, *Intelligent agents: An emerging technology for next generation telecommunications?*, INFOCOM'96. San Francisco, 1996, pp. 464-472.
- [5] D. Lange, M. Oshima, *Programming and deploying Java mobile agents with Aglets*, Addison -Wesley, 1998.
- [6] D. Chess, C. Harrison, A. Kershenbaum, "Mobile Agents: Are They a Good Idea?," *Lecture Notes in Computer Science*, Vol. 1222, SPRINGER VERLAG, 1997, pp. 25-45.
- [7] F. Bellifemine, A. Poggi, and G. Rimassa, "JADE: a FIPA2000 compliant agent development environment," In *Proceedings of the fifth international conference on Autonomous agents (Agent'1, Montreal, Canada, 2001*, pp. 216-217.
- [8] B. Chen, H. H. Cheng and J. Palen, "Mobile-C: a mobile agent platform for mobile C/C++ agents," *Software-Practice & Experience* 36 (15), 2006, pp. 1711-1733.
- [9] OMG, "Formal Specifications: Unified Modeling Language," <http://www.omg.org/spec/UML/>.
- [10] M. Fowler, K. Scott, *UML Distilled*(3rd ed.), Addison-Wesley, Reading, MA, 2003.
- [11] J. Odell, H. Parunak, and B. Bauer, "Extending UML for agents," in G. Wagner, Y. Lesperance, and E. Yu, (eds.), *Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence*, TX, 2000, pp. 3-17.
- [12] K. Saleh and C. El-Morr, "M-UML: an extension to UML for the modeling of mobile agent-based software systems," *Information and Software Technology*, 46, 2004, pp. 219-227.
- [13] M. Kang, L. Wang, K. Taguchi, "Modelling Mobile Agent Applications in UML 2.0 Activity Diagrams," *Proceedings of the 3rd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELAMAS 2004)*, May 24-25, Edinburgh, United Kingdom, 2004, pp. 104-111.
- [14] H. Baumeister, N. Koch, P. Kosiuczenko and M. Wirsing, "Extending activity diagrams to model mobile systems," *Intl. Conf. NetObjectDays, 2002. Revised Papers*, LNCS Vol. 2591, Springer, 2003, pp. 278-293.
- [15] P. Kosiuczenko, "Sequence Diagrams for Mobility," Krogstie J. (ed.): *Proc. of MobIMod workshop*, Tampere, Finland, October 2002, Revised Papers, LNCS Vol. 2784, Springer, 2003, pp. 147-155.
- [16] M. R. Bahri, R. Mokhtari, and A. Chaoui, "Towards an extension of UML2.0 to model mobile agent-based systems," *International Journal of Computer Science and Network Security*, VOL. 9 No. 10, October 2009, pp. 124-131.
- [17] E. Belloni and C. Marcos, "MAM-UML: An UML Profile for the Modelling of Mobile-Agent Applications," in *Proceedings of the XXIV International Conference of the Chilean Computer Science Society (SCCC'04)*, 2004, pp. 3-13.

■ 저자소개 ■



유 문 성  
Yoo, Moon Sung

2000년 9월~현재  
상지대학교 컴퓨터공학부 교수  
1996년 12월 루이지애나대학교  
컴퓨터학과(박사)  
1992년 인디애나대 컴퓨터학과(석사)  
1978년 서울대 수학과 (학사)  
관심분야 : 소프트웨어공학, 분산컴퓨팅,  
웹 소프트웨어, 모바일 소프트웨어  
E-mail : msyoo@sangji.ac.kr

논문접수일 : 2011년 2월 17일
수 정 일 : 2011년 2월 27일
게재확정일 : 2011년 3월 03일