적응성 지원을 위한 메타 레벨 기반의 이동 에이전트 프레임워크

김 수 중[†] · 윤 용 익[††]

요 약

멀티미디어, 이동 컴퓨팅 등 최근 등장하고 있는 새로운 기술들은 이기종 환경에서의 보다 유연한 서비스를 제공하기 위해서 높은 적응성을 지원하는 미들웨어 플랫폼을 요구하고 있다. 분산 유무선 환경에서 높은 서비스 품질을 제공하기 위해서는, 응용 프로그램과 미들웨어가 사용자 요구사항의 변화 뿐만 아니라 환경의 상태 변화를 인지해야 할 필요가 있으며 변화에 따라 시스템 동작을 적응시킬 수 있어야 한다. 이러한 미들웨어에서의 적응성 지원 요구에 따라, 본 논문에서는 리플렉션(Reflection) 기법을 적용하여 컴포넌트 기반의 이동 에이전트 프레임워크를 베이스 레벨(base level)과 메타 레벨(meta level)로 구조화하고 이동 에이전트에 의해 발생하는 변경 사항을 시스템에 반영할 수 있도록 하는 메타 에이전트(meta agent) 및 메타-서비스 에이전트(meta-service agent)를 제안한다. 프레임워크의 메타 레벨에서 메타 에이전트는 이동 에이전트의 실행을 감시하며, 메타 에이전트와 메타-서비스 에이전트를 통해 동적인 사용자 요구 사항 반영 및 응용 서비스 배치, 서비스 맞춤 구성을 제공할 수 있다.

Mobile Agents Framework for Adaptability Support based on Meta Level

Soo-Joong Ghim[†] · Yong-Ik Yoon[††]

ABSTRACT

Emerging technologies, such as multimedia and mobile computing, require that middleware platforms can support high adaptability in order to provide more flexible services in heterogeneous environments. To support high quality of service in distributed wired/wireless environments, it will be necessary for applications and middleware to be aware of changes in users requirements as well as environmental conditions, also to be able to adapt their behaviour as such changes. According to the needs of adaptability supporting in middleware, we structure a component-based, mobile agents framework in base level and meta level by using reflection. We propose concepts of meta agents and meta-service agents that are able to reflect changes made by mobile agents to the system. At the meta level of our framework, meta agents monitor execution of mobile agents and it is possible to provide dynamic adaptation of users requirements, deployment of application services and service customization with meta agents and meta-service agents.

키워드 : 이동 에이전트(Mobile Agent), 적응성(Adaptability), 리플렉션(Reflection), 미들웨어(Middleware)

1. 서 론

유무선 응용 서비스를 위한 응용 서버 미들웨어는 유무선 통합 인터넷 환경에서 이동 비즈니스 서비스 뿐만 아니라 기존의 엔터프라이즈 비즈니스 서비스를 이동 단말기, 운영체제 및 통신 프로토콜에 대해 독립적인 자바 기반의 분산 객체 컴포넌트로 개발 및 배포하고 유지보수 및 재사용이 가능하도록 해야 한다. 그러나 현 단계의 이동 응용 서비스에 대한 연구는 이동 컨텐츠 개발과 데이터 동기화 등에 편중되어 있어 사용자 요구사항을 반영한 서비스는 아직 제공되지 못하고 있다.

유무선 환경에서 적응성을 갖춘 응용 서비스를 지원하기 위해서는 응용 서비스가 실행 환경의 변화를 검출할 수 있어야 하며 이러한 변경된 사항들이 수용될 수 있도록 하부의 미들웨어 기능 요소들을 재구성할 수 있어야 한다[1]. 현재 이러한 문제를 해결하기 위한 방안으로 리플렉션(Reflection) 기법 적용에 대한 연구가 미들웨어 분야에서 활발히 진행되고 있다. 차세대의 유무선 환경에서 요구하는 응용 서비스들은 기존의 요구 조건에 보다 정확한 시간에 사용자가 자신이 원하는 서비스만을 선택해서 제공받을 수 있도록 요구하고 있으며 특히, 멀티미디어 기술의 발달로 높은 품질의 서비스를 필요로 한다. 이러한 필요성에 따라서 리플렉티브 미들웨어[2, 3]는 사용자 개개인의 요구사항을 만족시키면서 이기종의 플랫폼을 사용하는 사용자가 요청하는 비디오, 오디오 등과 같은 멀티미디어 정보를 제공하는

※ 본 연구는 2003년도 숙명여자대학교 교비연구비 지원으로 수행됨.
† 준 회 원 : 숙명여자대학교 대학원 컴퓨터과학과
†† 종신회원 : 숙명여자대학교 정보과학부 교수
논문접수 : 2003년 6월 4일, 심사완료 : 2003년 11월 27일

응용 서버의 플랫폼으로 응용 가능한 기술이다.

본 논문에서는 이러한 리플렉션 기법을 적용하여 동적으로 적응성을 지원할 수 있는 이동 에이전트 기반의 미들웨어 프레임워크를 제안한다. 이는 적응성 지원에 있어서 기존 미들웨어 플랫폼들의 정적인 블랙-박스(black-box) 형식의 구조로 인한 취약점을 해결함과 동시에 컴포넌트 모델을 채택함으로써 재사용성과 확장성을 제공할 수 있다. 또한 이동 에이전트를 사용하여 보다 유연적인 이동 코드 기반의 이동 서비스를 지원할 수 있는 프레임워크 구조이다.

본 논문의 구성은 다음과 같다. 2장에서는 연구 배경으로서 유무선 환경에서의 적응성 지원 문제와 이동 에이전트에 대해 간략히 설명한다. 3장에서는 리플렉션 기법을 사용하여 적응성을 지원하기 위한 이동 에이전트 기반 프레임워크 구조를 제안하고 각 구성 요소에 대해 기술한다. 4장에서는 메타 에이전트 및 메타-서비스 에이전트의 알고리즘과 이동 에이전트 기반 프레임워크를 주문형 비디오 응용에 적용한 구현 예제를 설명하며, 끝으로 5장에서는 연구의 결론 및 향후 연구 과제에 대해 언급한다.

2. 연구 배경

2.1 적응성과 리플렉티브 미들웨어

분산 환경에서의 응용은 사용자의 요구사항 변경과 사용자의 서비스 환경 상태의 변화를 적응 시킴으로써 사용자에게 최선의 서비스를 제공할 수 있어야 한다. 또한 상태 변화에 따른 하부 플랫폼의 동작에 서비스의 동작을 적응시킬 수 있어야 하며, 시스템은 이러한 정보를 사용하여 시스템 자체의 동적인 구성이 분산된 형태로 이루어지도록 해야한다[4].

특히, PDA(Personal Digital Assistants)부터 워크스테이션에 이르는 다양한 종단 시스템이 연결된 유무선 통합 환경에서 요구되는 적응성은 상위 응용 레벨에서 하위 운영체제 레벨까지 시스템의 모든 측면에 적용된다. 이러한 적응성은 통신 상의 성능, 자원 사용, 위치, 응용 서비스의 선호도 등을 포함한다.

그러나 특정 레벨에 국한되는 적응성 지원은 몇 가지 문제점을 초래할 수 있다. 예를 들어, 운영체제 레벨에서의 적응성은 무결성 및 성능 상의 문제를 발생시킬 수 있으므로 매우 주의 깊게 처리되어야 할 것이다. 또한, 이러한 경우 적응성 획득을 위해서는 반드시 운영체제에 의존해야 하기 때문에 응용 프로그램의 이식성이 손상될 수 있다. 이와 반대로, 응용 레벨에만 의존할 경우에는 적응성 지원 메커니즘을 응용 프로그램 내에서 모두 구현해야 하므로 개발에 대한 막대한 부담이 가중될 것이다.

이러한 문제를 해결하기 위하여 미들웨어 레벨에서의 적응성 지원이 제기되면서 최근 리플렉티브 미들웨어에 대한 연구가 진행되고 있다. 리플렉티브 미들웨어란 응용이 실행되는 환경 내에서 발생한 변화가 하부 미들웨어에 반영되고, 응용이나 사용자가 명시적으로 제어할 필요 없이 코어 소프트웨어와 하드웨어 메커니즘을 동적으로 적응시킴으로써 응용의 자율적인 동작 변화를 가능하게 하는 기술이다[5].

분산 유무선 환경을 고려할 때, 미들웨어가 갖추어야 할 주요한 요건에는 적응성 지원 이외에 서비스 코드의 이동성(mobility)에 대한 관리를 들 수 있다. 차세대 유무선 응용 서비스에서는 사용자와의 원활한 상호작용 뿐만 아니라 사용자 요구사항에 대한 반영이 중요시 되며, 다양한 연결 형태를 가지고 있는 종단 사용자 시스템들에게 유연한 서비스 제공이 이루어져야 한다. 그러나 현 단계의 리플렉티브 미들웨어에서는 이동 코드를 기반으로 하는 응용 서비스와 그 환경에 대한 구체적인 연구가 진행되지 않고 있다. 따라서 적응성 지원을 위한 리플렉션 기법과 동시에 유무선 통합 환경의 특성을 고려한 효율적인 이동 서비스 기법을 결합한 미들웨어 플랫폼이 요구된다.

2.2 이동 에이전트

이동 에이전트 기술은 네트워크 부하를 감소시키고 네트워크 지연과 대역폭의 제약성을 극복할 수 있는 장점을 포함하고 있으며 이동 에이전트 기반으로 응용 서비스를 동적으로 구성할 수 있기 때문에 많은 다양한 응용 시스템을 구축하는 데 효율적이라 할 수 있다.

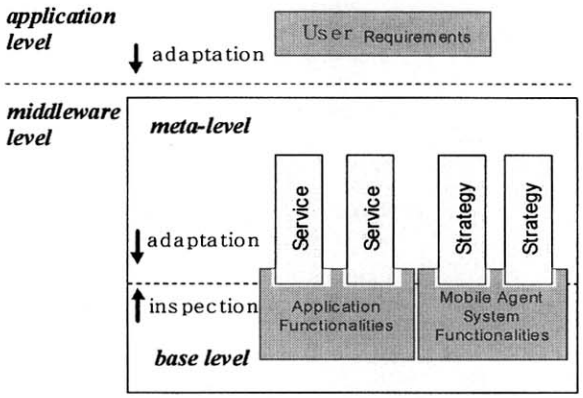
지금까지 국외를 중심으로 다수의 이동 에이전트 시스템들(Aglets[6], Ara[7], Concordia[8], D'Agent[9], Mole[10] 등)이 개발되었으나 실제로 이들 시스템들은 상이한 구조를 가지고 있으며 구현 상세 기술에 대해 매우 종속적이기 때문에 보다 유연한 응용 서비스를 지원하는데 있어서 제약이 따른다. 이러한 이동 에이전트 시스템은 이동 에이전트의 장점을 바탕으로 하나의 미들웨어 플랫폼으로서 활용될 수 있는 잠재력을 내포하고 있음에도 불구하고 이들 대부분이 전통적인 블랙-박스 기법에 따라 설계 및 구축되었기 때문에 적응성 지원은 ad hoc 방식으로만 가능하며 실제로 적응성 문제에 대한 세부적인 사항은 언급되지 않고 있다.

3. 이동 에이전트 기반 프레임워크

3.1 2-레벨 적응 모델

본 논문에서는 리플렉션 기법을 적용하여 적응성의 관점을 크게 두 가지로 분류한다. 첫째, 응용 레벨의 적응성은 사용자 요구사항을 반영한 서비스 제공과 관련된다. 사용자는 응용 서버가 제공하는 다양한 서비스에 대해 요구사항을 제시함으로써 필요한 서비스만을 원하는 형태(예 : 미디어 형식, 해상도 등)로 제공받을 수 있다. 이러한 정보는 이동 에이전트에 의해 수집 및 분석되어 서비스 제공 시에 동적으로 반영된다. 둘째, 미들웨어 레벨의 적응성은 플랫폼의 컴포넌트 변경에 따른 동적인 구성 및 관리와 관련된다.

미들웨어 레벨은 다시 세부적으로 베이스 레벨과 메타 레벨로 분리된다. 베이스 레벨에는 응용 및 시스템의 기본적인 기능 컴포넌트들이 위치하며 검사(inspection) 메커니즘을 통해 베이스 레벨을 구성하는 환경 정보가 메타 레벨에게 제공된다. 메타 레벨은 변경이 가능한 응용 서비스 컴포넌트와 이동 에이전트를 위한 전략 컴포넌트들로 구성된다. (그림 1)에서 본 논문의 개념적인 적응 모델을 나타내었다.



(그림 1) 2-레벨 적응 모델

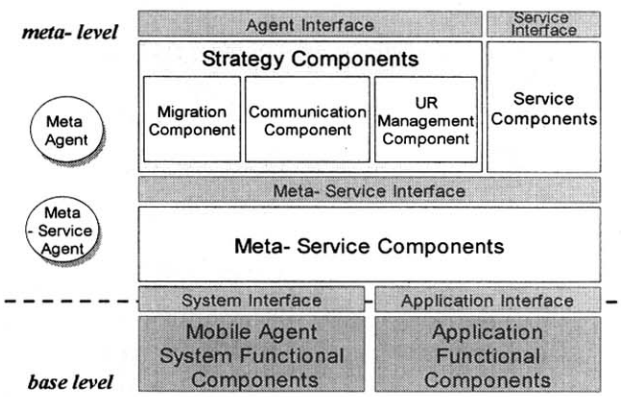
3.2 컴포넌트 모델

컴포넌트 기술은 소프트웨어 시스템이 변경된 사용자 요구사항을 반영하고 컴포넌트의 추가, 삭제 및 대체를 통해 시스템을 재구성할 수 있도록 하므로 적응성을 제공하기 위한 필수적인 기반 기술로 인식되고 있다. 그러나 현재의 컴포넌트 기술은 미들웨어 하부구조 상의 응용 레벨에만 적용되고 있기 때문에 컴포넌트 개발자가 미들웨어의 내부 구조에 대해 상세히 파악할 수 없다[11]. 그러므로 적응성에 대한 요구를 수용하기 위해서는 응용 뿐만 아니라 미들웨어 계층 자체가 컴포넌트 기반 구조로 이루어져야 한다. 따라서 본 논문에서는 적응 메커니즘의 수행 및 제어를 담당하는 두 가지의 에이전트, 즉 메타 에이전트와 메타 서비스 에이전트를 컴포넌트로 정의하고 다양한 응용 도메인에 대해서 시스템 구성이 용이한 컴포넌트 프레임워크를 제안한다. (그림 2)에서는 이 프레임워크의 구조를 나타내었다.

3.2.1 전략 컴포넌트(Strategy Component)와 서비스 컴포넌트(Service Component)

전략 컴포넌트는 이동 에이전트의 이동, 통신, 사용자 요구사항 관리 등 에이전트의 자율적인 수행과 관련된 메커니즘을 구현하는 컴포넌트이다. 예를 들어, RMI나 소켓 등이 구현된 통신 컴포넌트(Communication component)를 플러그-인 하여 이동 에이전트가 여러가지 방식으로 로컬 및 리모트 통신을 할 수 있도록 구성할 수 있다. 이동 컴포넌트(Migration component)로는 강 이동 방식과 약 이동 방식을 구현할 수 있으며, 사용자 요구사항 관리 컴포넌트(UR Management component)는 요구사항 정보의 수집 및 분석 알고리즘을 구현한다. 이외에도 전략 컴포넌트는 서비스 품질 관리, 실시간 관리 등 이동 에이전트에 요구되는 기능에 따라 컴포넌트를 추가함으로써 확장할 수 있다.

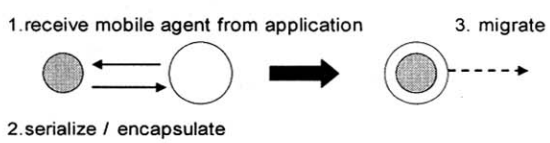
서비스 컴포넌트는 이동 에이전트의 중재나 에이전트 자체에 의해서 동적으로 다운로딩 가능한 컴포넌트이다. 새로운 응용 서비스의 배치나 기존 서비스의 업그레이드는 메타 레벨에서 베이스 레벨로의 서비스 컴포넌트 플러그-인을 통해 이루어진다.



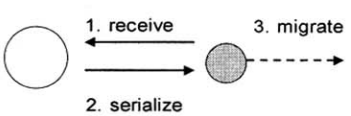
(그림 2) 이동 에이전트 컴포넌트 프레임워크

3.2.2 메타 에이전트(Meta Agent)

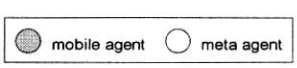
메타 에이전트는 자체적인 전략을 포함하고 있는 컴포넌트로서, 적응 처리에 대한 제어 및 관리를 위한 에이전트이다. 따라서 메타 에이전트의 주요 역할은 다른 이동 에이전트들의 실행을 모니터링하고 이들을 리모트 호스트로 전송하는 것이다. 메타 에이전트와 이동 에이전트의 연관은 일대일 방식과 일대다 방식으로 수행할 수 있다.



(a) one meta agent - one mobile agent



(b) one meta agent - many mobile agents



(그림 3) 메타 에이전트와 이동 에이전트의 연관

먼저 전자의 경우(그림 3)(a), 하나의 이동 에이전트에 대

하여 하나의 메타 에이전트가 생성되며 이동 시에는 메타 에이전트가 내부에 이동 에이전트를 포함하여 직접 이동하게 된다. 그러므로 이동 에이전트의 이동 경로를 따라서 메타 에이전트가 계속적으로 이동 에이전트의 행동을 감시할 수 있다. 후자의 경우에는(그림 3)(b), 하나의 메타 에이전트가 다수의 이동 에이전트를 관리한다. 그러므로 전자와는 달리 메타 에이전트에 대해 이동성은 지원되지 않는다. 이때 어떤 이동 에이전트를 관리할 것인가와 얼마나 많은 에이전트를 관리할 것인가는 메타 에이전트 내의 전략에 따라 조정될 수 있도록 한다.

3.2.3 메타-서비스 컴포넌트(Meta-Service Component)와 메타-서비스 에이전트 (Meta-Service Agent)

메타-서비스 컴포넌트는 베이스 레벨에 위치하는 응용의 기능 컴포넌트(Application functional component)와 이동 에이전트의 시스템 기능 컴포넌트(Mobile Agent System functional component)로부터 검사 메커니즘을 통한 추상 서비스를 제공한다. 또한 현재의 컴포넌트 구성 정보를 유지하며, 적응 수행을 위하여 메타-서비스 인터페이스(Meta-Service interface)를 제공한다. 이동 에이전트에 의한 구성 변경의 허용 범위는 메타-서비스 컴포넌트에 포함된 구성 규칙을 적용하여 제어될 수 있다.

메타-서비스 에이전트는 컴포넌트의 구성 및 재구성을 제어하는 역할을 담당하는 시스템 상주(stationary) 에이전트이다. 런-타임 시에 메타-서비스 에이전트의 제어 하에 베이스 레벨의 구성 정보에 대한 검사가 수행되고 이 정보를 이용하여 메타-서비스가 형성된다.

3.3 동적인 적응성 지원

앞서 설명한 바와 같이, 미들웨어에서의 적응성 지원을 위하여 본 논문에서 적용한 방안은 첫째, 리플렉션 기법과 둘째, 컴포넌트 기술이다. 이에 따라, 이동 에이전트 프레임워크의 베이스 레벨에는 이동 에이전트의 생성, 실행, 제거 등과 관련된 시스템 컴포넌트들이 위치하고, 메타 레벨에는 사용자 요구 사항 관리, 이동 에이전트의 이동, 통신 등의 전략적인 메커니즘을 구현하는 컴포넌트들이 위치한다. 이는 기존의 미들웨어 시스템들이 블랙-박스 형태로 은폐하고 있는 시스템의 내부 구조로부터 전략 메커니즘을 메타 레벨로 분리함으로써 메타 레벨 컴포넌트에 대한 명시적인 표현이 가능하고, 컴포넌트들 간의 동적인 상호작용을 유지하면서 시스템이 수행되는 동안 특정 전략들을 변경할 수 있도록 하였다. 새로이 요구되는 전략 컴포넌트들은 플러그-인 방식으로 추가하여 시스템의 기능을 확장할 수 있다. 또한 메타 레벨에 응용 서비스 컴포넌트를 포함시킴으로써 사용자 요구에 따른 서비스의 맞춤 구성 및 새로운 응용 서비스 배치를 동적으로 수행할 수 있다.

4. 구 현

4.1 메타 에이전트와 메타-서비스 에이전트

메타 레벨에서 발생하는 변화를 모니터링 및 관리하여 시스템에 반영될 수 있도록 하는 핵심적인 컴포넌트는 메타 에이전트와 메타-서비스 에이전트이다. 메타 레벨의 반영과 관련하여 이들 두 형태의 에이전트들은 notify(), inspect(), adapt() 메소드를 사용한다. 다음의 (그림 4)와 (그림 5)는 각각 이벤트 발생에 따른 메타 에이전트와 메타-서비스 에이전트의 알고리즘이다.

```
/* meta agent */
loop( ) {
    switch(event)
        event i : mobile agent received from application
            establish point-to-point channel with remote meta
            agent ;
            serialize mobile agent ;
            stop monitoring ;
            transfer mobile agent ;
            break ;
        event j : mobile agent received from remote place
            register mobile agent ;
            deserialize mobile agent ;
            start monitoring ;
            break ;
        event k : changes are detected
            notify to meta-service agent ;
            break ;
}
```

(그림 4) 메타 에이전트

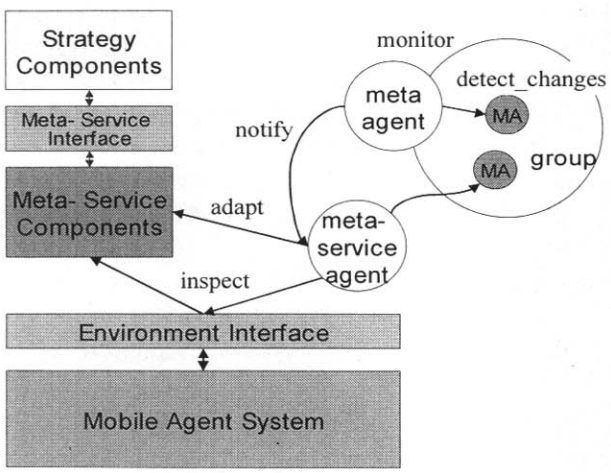
```
/* meta-service agent */
loop( ) {
    switch(event)
        event p : inspection
            do inspection ;
            construct meta-service ;
            break ;
        event q : changes are notified
            do adaptation ;
            break ;
}
```

(그림 5) 메타-서비스 에이전트

메타 에이전트와 이동 에이전트의 연관 관계가 1 : n으로 하나의 메타 에이전트가 여러 이동 에이전트를 모니터링한다고 가정할 때, 이동 에이전트가 리모트 호스트로의 이동을 요청하면 이와 연관된 메타 에이전트는 이동 전략에 따라 요청한 에이전트의 상태와 코드를 직렬화하고 목적지의 메타 에이전트와의 원격 통신을 통해 점-대-점 채널을 성립한다. 직렬화된 에이전트를 수신한 목적지의 메타 에이전트는 이를 복원하고 에이전트가 실행되기 전에 모니터링을 시작한다. 메타 에이전트가 이동 에이전트의 동작을 감시하는 동안 메타 레벨에서의 컴포넌트 변경이 감지될 경우, 메

타 에이전트는 notify()를 호출하여 메타-서비스 에이전트에게 이를 통보한다. 메타-서비스 에이전트는 inspect() 통해 컴포넌트의 구성에 대한 정보를 메타-서비스에 포함하며, 메타 에이전트로부터 변경을 통보 받은 후 adapt() 를 통해 메타서비스를 수정하여 컴포넌트의 변경을 반영시킨다.

또한 사용자 요구사항 반영에 있어서는, 우선 이동 에이전트가 사용자 요구사항 정보를 수집 및 분석한 후, UR 관리 컴포넌트에 구현된 전략에 따라 사용자에 대한 서비스 제공 시에 반영할 수 있다.



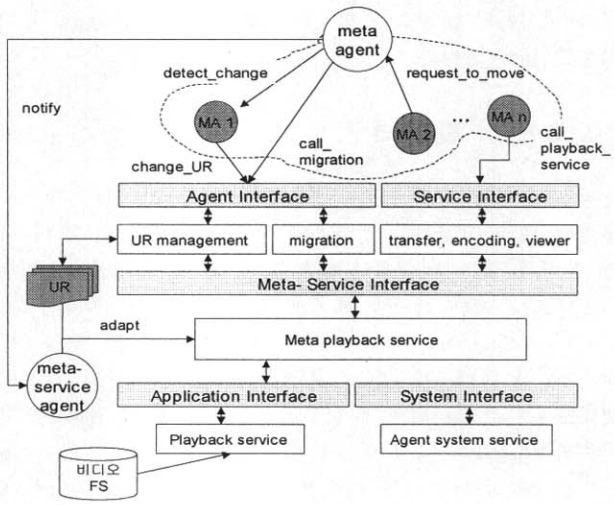
(그림 6) 컴포넌트 구성 변경에 따른 메타 에이전트와 메타-서비스 에이전트의 상호작용

4.2 구현 예제

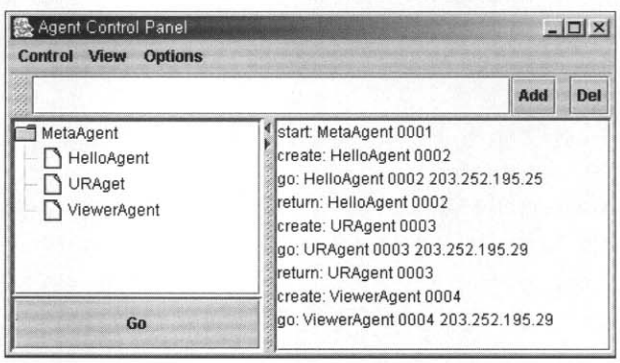
본 논문에서는 이동 에이전트 기반 프레임워크의 일차적인 응용 도메인을 유선 환경에서의 주문형 비디오 서비스에 맞추어 시험용 시스템을 구현하였다. 구현 환경은 10Mbps의 로컬 네트워크로 연결된 개인용 컴퓨터(750Mhz, 256MB, Windows 2000 Server/Windows XP)와 워크스테이션(Sun Sparc20, Solaris 2.6)들로 구성하였다.

주문형 비디오 서비스를 위한 시험용 시스템은 본 논문에서 제안한 리플렉티브 프레임워크에 따라 베이스 레벨과 메타 레벨로 설계하였다. 베이스 레벨에는 에이전트를 위한 기능 컴포넌트의 기본적인 재생 컴포넌트를 구현하고, 메타 레벨에는 사용자 요구사항 관리 전략 컴포넌트와 에이전트 이동 전략 컴포넌트 및 전송, 인코딩, 뷰어 컴포넌트를 플러그-인 가능하도록 구현하였다. (그림 7)은 시험용 시스템의 구성도이다.

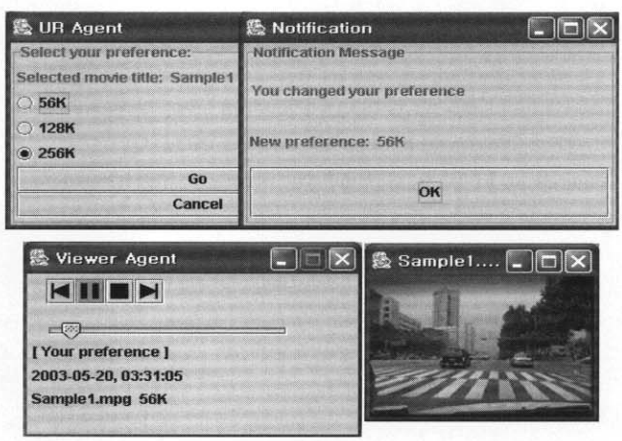
이 시스템에서 비디오 재생 서비스는 에이전트를 통해 제공되고 사용자 요구 사항은 메타 에이전트 및 메타-서비스 에이전트의 상호작용에 의해 반영된다. (그림 8)과 같이 에이전트 제어 프로그램을 통해 하나의 메타 에이전트의 감독 하에 있는 이동 에이전트의 생성/소멸과 목적지로의 이동 등을 관리할 수 있다.



(그림 7) 주문형 비디오 서비스 프레임워크



(그림 8) 에이전트 제어용 프로그램



(그림 9) UR 에이전트와 뷰어 에이전트

(그림 9)는 시험용 시스템에서 구현한 UR(User Requirements) 에이전트와 뷰어 에이전트의 사용자 인터페이스 화면이다. 사용자는 UR에이전트를 통해 새로운 요구사항을 추가하거나 수정할 수 있으며 이러한 정보는 이동 에이전트에 의해 분석되어 UR 관리 컴포넌트에 반영된다. 이때

메타 에이전트는 변화가 발생하였음을 통보(notification) 메시지를 통해 알린다.

5. 결론 및 향후 연구

본 논문에서는 현재의 분산 시스템이 안고 있는 적응성 문제에 대해 고찰하고, 기존 미들웨어 플랫폼들의 블랙-박스 형식과 달리 리플렉션 기법을 적용하여 시스템의 내부 컴포넌트를 메타 레벨과 베이스 레벨로 분리함으로써 시스템이 수행되는 동안 메타 레벨 컴포넌트의 구성 변경을 동적으로 반영시킬 수 있는 프레임워크를 제안하고 주문형 비디오 서비스를 위한 시험용 시스템을 구현하였다. 또한 기존의 미들웨어 플랫폼들이 다루고 있지 않은 이동 서비스 지원을 위하여 이동 에이전트 기법을 도입하고 메타 에이전트와 메타-서비스 에이전트라는 새로운 개념을 통해 사용자 요구사항의 반영, 서비스의 맞춤 구성, 동적인 서비스 배치 등을 가능하게 하는 이동 에이전트 기반의 능동적이고 적응적인 시스템 구성 방안을 제시하였다. 그러므로 본 논문의 연구는 보다 유연한 운용 및 관리가 요구되는 유무선 통합 환경에서의 효율적인 미들웨어 플랫폼으로 발전할 것으로 기대된다.

이에 더하여 사용자 시스템의 다양한 하드웨어적 특성과 연결 형태를 지원하고 이동 환경에서의 자원 가용성을 탐색하여 높은 서비스 품질을 보장하는 리플렉티브 이동 서비스 기술에 대한 연구가 향후 추진되어야 할 것이다.

참 고 문 헌

[1] F. Kon, R. H. Campbell, *Supporting Automatic Configuration of Component-Based Distributed Systems*, Proceedings of the 5th USENIX Conference on Object-Oriented Technologies and Systems, May, 1999.

[2] F. Kon, K. B. S., G. Blair, R. Campbell, *IFIP/ACM Middleware '2000 Worksop on Reflective Middleware*, New York, 2000.

[3] M. Roman, F. Kon, R. Campbell, *Reflective Middleware : From Your Desk to Your Hand*, IEEE Distributed Systems Online (Special Issue on Reflective Middleware), Vol.2, No.5, July, 2001.

[4] G. Blair, G. Coulson, A. Andersen, et al., *A Principled Approach to Supporting Adaptation in Distributed Mobile Environments*, Proceedings of the 5th International Symposium on Software Engineering for Parallel and Distributed Systems (PDSE '2000), Limerick, Ireland, IEEE, June, pp.3-12, 2000.

[5] J. A. Zinky, D. E. Bakken, R. Schantz, *Architectural Support for Quality of Service for CORBA Objects*, Theory and Practice of Object Systems, Vol.3, No.1, 1997.

[6] D. B. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents with Aglets*, Addison-Wesley, 1998.

[7] H. Peine, T. Stolpman, *The Architecture of the Ara Platform for Mobile Agents*, First International Workshop on Mobile Agents, MA '97, April, 1997.

[8] Concordia, *Mobile Agent Computing -A White Paper*, Mitsubishi Electric Information Technology Center of America, 1997.

[9] Robert S. Gray, *Agent Tcl : A transportable agent system*, In Proceedings of the CIKM Workshop on Intelligent Information Agents, Fourth International Conference on Information and Knowledge Management (CIKM 95), Baltimore, Maryland, December, 1995.

[10] J. Baumann, F. Hohl, K. Rothermel, M. Strasser, *Mole - Concepts of a Mobile Agent System*, Technical report 1997/15, Fakultaet Infor-matik, University of Stuttgart, August, 1997.

[11] N. Parlavantzas, G. Couison, M. Clarke, G. Biair, *Towards a Reflective Component-based Middleware Architecture*, Workshop on Reflective Middleware, April, 2000.

김 수 중

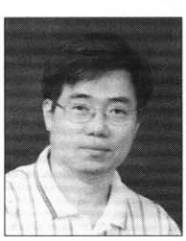
e-mail : sjghim@sookmyung.ac.kr

1995년 숙명여자대학교 전산학과(이학사)

1998년 숙명여자대학교 전산학과(이학석사)

1998년~현재 숙명여자대학교 컴퓨터과학과 박사과정

관심분야 : 유비쿼터스 컴퓨팅, 분산 미들웨어 시스템, 모바일 에이전트



윤 용 익

e-mail : yiyoon@sookmyung.ac.kr

1983년 동국대학교 통계학과(이학사)

1985년 한국과학기술원 전산학과(공학석사)

1994년 한국과학기술원 전산학과(공학박사)

1998년~현재 숙명여자대학교 정보과학부 교수

관심분야 : 유비쿼터스 컴퓨팅, 멀티미디어 시스템, 분산시스템, 실시간처리시스템, 미들웨어, 실시간 OS/DBMS