

PDA용 소프트웨어를 위한 무선 에이전트 클래스 설계 및 구현

김종완^u 김태환 류성열

승실대학교 컴퓨터학과

wany69@hitel.net, thkim126@selab.ssu.ac.kr,

syrhew@computing.soongsil.ac.kr

Design and Implementation of Wireless Agent Class for PDA Software

Jong-Wan Kim^u Tae-Whan Kim Sung-Yul Rhew
School of Computing, Soongsil University

요 약

PDA를 이용한 무선통신 단말의 구성은 이동성, 편리성이 강조되어야 하며, 언제 어디서나 사용자가 원하는 정보를 받아 볼 수 있어야 한다. 이러한 필요성에 의해 정보를 자동으로 다운로드(download)한다면 사용자의 편리성은 더욱 증대될 것이다. 따라서 PDA용 소프트웨어를 개발할 때 필요한 구성요소로서 무선 에이전트(wireless agent) 개념을 추가하여 소프트웨어 계층을 구성하면 별도의 소프트웨어 개발단계 없이 무선 에이전트가 추가된 소프트웨어를 쉽게 구성할 수 있다. 본 논문에서는 PDA를 위한 소프트웨어에 에이전트개념을 도입하여 무선이동단말기의 이용효율과 편리성을 극대화하고자 무선 에이전트 클래스를 설계하고 구현하였다.

1. 서론

PDA(personal digital assistant)에서의 소프트웨어는 데스크탑 소프트웨어 구성과 유사하지만 하드디스크가 아닌 램 IC에 직접 저장되고 실행된다는 의미에서 임베디드(embedded) 소프트웨어라 한다.[9] PDA는 개인정보관리와 같이 스탠드얼론(stand alone)으로 사용되는 소프트웨어도 있지만 최근 이동의 편리성과 통신의 필요성이 결합하여 무선 이동(wireless mobile) 단말기로서의 이용가치가 증가되고 있다.

최근 인터넷 네트워킹 기술이 아날로그에서 디지털로 변화하면서 데이터, 음성, 영상등 개별 서비스에서 멀티미디어로 탈바꿈하고 있다[1]. IMT-2000이 대두되면서 이동통신과 함께 무선 통신에 대한 관심도 많이 높아지고 있다. 따라서 PDA와 같은 이동 단말기를 사용하여 무선 인터넷을 이용할 때 정보제공자(contents provider)의 서버에서 원하는 정보를 직접 찾아보거나 서버에서 보내주는 서비스를 일방적으로 제공받기 보다는 원하는 정보를 서버에서 자동으로 찾아주는 기능이 이동 단말기에 포함되어 있다면 편리할 것이다. 이러한 기능은 PDA에서 개발되는 소프트웨어에 무선 에이전트기능을 포함시킴으로써 가능하다.

본 논문은 PDA장비에서 사용 가능한 무선 통신용 소프트웨어를 개발할 때 소프트웨어 구성 아키텍처에 인터페이스 에이전트를 한 부분으로 추가하여 무선 에이전트가 하나의 객체로 생성될 수 있도록 클래스의 설계 및 구현을 보임으로서 PDA용 소프트웨어 아키텍처를 확장하였으며, 무선이동단말로서의 기능을 갖도록 소프트웨어 구성을 극대화하였다.

2. 관련연구

2.1 에이전트 구성

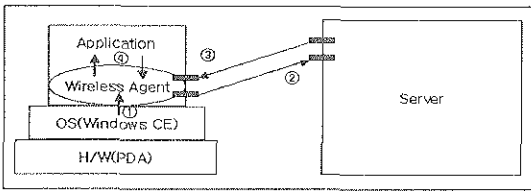
에이전트란 사용자가 해야할 작업이나 반복적인 작업을 자동적으로 수행해주는 소프트웨어이다 [2][3].

에이전트는 기본적으로 지시된 작업을 수행하고 조정하기 위한 제어지식과 추론능력을 갖는 에이전트 엔진(engine)과 수행할 작업에 따른 에이전트의 역할을 규정하게 되는 영역지식(domain knowledge), 그리고 에이전트들 사이에 프로시저 콜(procedure call)과 메시지 교환(message passing)을 담당하는 통신모듈(communication module)로 구성된다. 이 모듈은 한 에이전트가 정보를 찾지 못한 경우 링크되어 있는 다른 에이전트의 도움을 요청할 때 사용되며, 에이전트들은 상호통신하여 협조한다.

2.2 분산환경의 에이전트

분산환경 하에서 에이전트는 인터페이스(interface), TASK(task), 정보(information) 에이전트로 구분하며, 인터페이스 에이전트는 사용자와 직접적으로 상호작용 하면서 사용자의 요구사항을 분석, 처리한다. TASK에이전트는 영역지식을 기반으로하여 사용자의 요구사항을 직접 수행한다. 또한 여러곳에 흩어져있는 정보에 접근할 수 있는 기능을 가진 경우 정보에이전트라 한다.

현재 여러 분야에서 에이전트 개념이 적용되어 사용하고 있지만 특히, 무선 인터넷에서 사용되는 에이전트는 [그림1]에서 보는 바와 같이 사용자가 필요로 하는 정보를 스스로 찾아 일정한 시간에 보여 줄 수 있어야 한다.

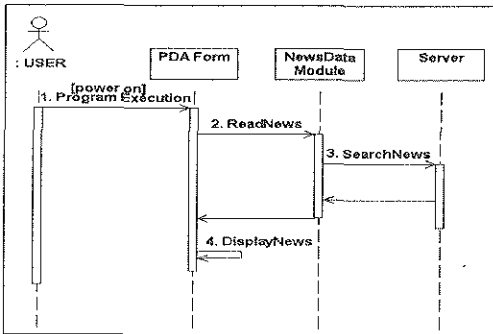


[그림1] 무선 인터페이스 에이전트

3. 클래스 설계 및 구현

3.1 기본 개념

무선 모바일 장비 사용자는 언제 어디서나 원하는 정보를 쉽고 빠르게 요구하여 습득할 수 있어야 한다. 또한 사용자가 정보를 찾을 수 없는 상황일 때 이를 대신할 수 있는 대리인이 있으면 정보의 습득시간이 많이 단축될 것이다. 그러나 기존의 모바일 소프트웨어는 구성은 [그림2]와 같이 사용자가 직접 실행하여 NewsDataModule의 기능을 호출하면 되돌은 해당 사이트에 접속하여 정보를 찾고 이 과정에 사용자는 계속 참여해야한다. 또는 사용자가 지정한 옵션에 따라 서버에서 일방적으로 전송해주는 형식을 갖는다.



[그림2] PDA소프트웨어 작동 원리

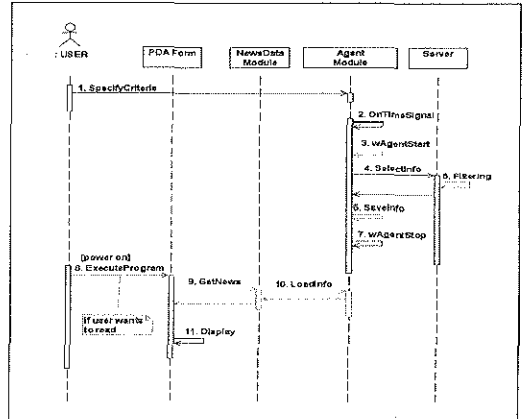
무선 인터페이스 에이전트는 하드웨어에 임베디드 된 상태에서 운영체제로 부터의 시그널을 계속 검사

하며, 소프트웨어를 실행하지 않은 경우에도 조건이 만족되면 에이전트는 자동으로 액티브(active)되어 [그림3]과 같이 서버 접속을 시도하고 접속후 사용자가 지정한 데이터를 자동으로 가져온다. 이때, 데이터의 업그레이드 유무를 스스로 판단하여 필요한 데이터를 다운로드 한다.

무선 에이전트로 넘어온 데이터는 응용프로그램에 푸시되거나 응용프로그램의 요청에 따라 정보를 넘겨준다. 요청이 없는 경우 취득된 정보는 메모리에 저장되어 관리되며, 사용자는 서버에 직접 접속하지 않아도 원하는 시간에 저장된 정보를 볼 수 있다. 정보 취득과정이 완료된 경우 7번의 메소드를 호출하여 다음 시그널이 발생하기를 대기한다.

이러한 과정에서 에이전트는 클라이언트에게 일관된 서비스의 가용성을 보장해주며, 그렇지 못하더라도 명확한 오류조건을 제공한다. 사용자는 서비스가 제공하는 객체를 사용하여 서비스와 상호작용을 할 수 있다[6][7].

이러한 작업은 별도의 독립된 소프트웨어로 개발하여 사용돼 왔지만 본 논문에서는 이를 소프트웨어 아키텍처에 삽입하여 보다 편리한 소프트웨어를 구성하도록 한다.



[그림3] 에이전트의 역할

3.2 개발 사례

클래스의 기능은 무선 인터페이스 에이전트를 설계 대상으로 하므로 일정한 시간에 지정된 조건에 의해 서버로부터 데이터를 가져올 수 있도록 설계한다. 이러한 개념은 대부분의 무선 에이전트에 공통된 기능이므로 동일하게 적용될 수 있다.

이 연구에서 설계하는 에이전트는 인터페이스와 TASK 기능을 갖는 클라이언트 측 단독 에이전트를 대상으로 하며, 이는 PDA에서 리얼타임으로 동작하는 임베디드 소프트웨어 객체이므로 UML(Unified Modeling Language)과 객체지향 리얼타임 모델링방법인 ROOM(Real-Time Object Oriented Modeling)을 적용하여 클래스를 설계하였다. 구현환경으로는 MIPS CPU에서 구동되는 운영체제 윈도우즈CE3.0 을

기반으로 하였으며, 임페디드 비주얼C++을 사용하였다[8].

ROOM에서 사용하는 설계방법에서는 기존의 클래스 개념과 비교하여 크게 두 가지 차이점이 있다.

첫째, 객체지향 프로그래밍에서 사용되는 클래스는 캡슐(capsule)로 표현되며, 캡슐은 클래스의 모든 기능과 함께 고유의 상태머신(state machine)을 갖는다[4].

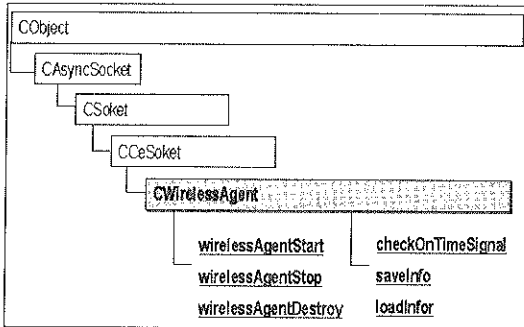
둘째, 실제 캡슐사이의 통신은 포트(port)라는 인터페이스를 통한, 메시지 전달(message passing)에 의해 이루어진다. 모든 메소드는 공용(public) 접근권한으로 선언되었을지라도 다른 캡슐에서 직접 접근할 수 없다[4].

에이전트 캡슐의 메소드 멤버는 에이전트 생성, 소멸을 포함하여 [표1]과 같이 구성된다.

[표1] 무선 에이전트 클래스의 메소드

메소드	세부기능
CWirelessAgent() CWirelessAgent(CString strCriteria)	생성자. 기본 생성자와 초기 설정조건을 갖는 생성자
void wirelessAgentStart(CString strCriteria)	시그널이 만족되면 에이전트를 실행
void wirelessAgentStop()	에이전트 작업을 중지시키고 wait상태로 전환
void wirelessAgentDestroy()	에이전트 객체의 서비스를 중단하고, 객체 소멸
void checkOnTimeSignal()	지정된 조건에 근거하여 시스템 시그널 확인
Boolean saveInfo(INFO infoData)	서버가 제공한 뉴스정보 저장
INFO loadInfo()	조건에 의해 미리 저장한 데이터 로드

설계된 클래스는 [그림4]에서 보는 것과 같이 소프트웨어 아키텍처의 상위 클래스인 Object로부터 상속된 소켓(CSocket) 클래스를 상속하여 최하위 클래스를 구성한다. 이는 다시 하위 클래스로 상속될 수 있다[5].



[그림4] 에이전트 클래스 계층도

새로운 소프트웨어 개발을 위해 프로젝트를 구성하면 [그림5]와 같이 무선 에이전트 클래스를 기반(base)으로 하는 프로젝트를 생성할 수 있으며, 해

당 메소드는 필요에 따라 오버라이딩(overriding)하여 사용한다.

```

class CWirelessAgentProjectView : public CWirelessAgent
{
    // Operations Overriding
public:
    void wirelessAgentStart(CString strCriteria);
    void wirelessAgentStop();
    void wirelessAgentDestroy();
    void checkOnTimeSignal();
    Boolean saveInfo(INFO infoData);
    INFO loadInfo();
protected:
    //{{AFX_MSG(CPdaView)
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};
    
```

[그림5] 상속된 에이전트 클래스

4. 결론 및 향후 연구 과제

본 논문에서 설계한 무선 에이전트 클래스는 무선기능이 있는 단말기용 소프트웨어를 구성할 때 개발시점부터 에이전트를 포함할 수 있도록 하였다. 이는 소프트웨어 개발자나 사용자 측면에서 보다 편리한 개발과 사용이 가능하도록 제시한 것이다.

향후에는 추론능력을 갖는 에이전트 엔진의 개발과 함께 플랫폼 및 운영체제 환경에 독립적인 무선 에이전트 개발에 대한 연구를 계속할 것이다.

5. 참고문헌

- [1] 송관호, "향후 인터넷의 발전전망," 정보처리학회지, Vol.7, No.3, pp.3-5, 2000.5.
- [2] S. Franklin, A. Graesser, Is it an agent, or just a program? : A taxonomy for autonomous agents, Proc. 3rd International Workshop on Agent Theories, Architectures, and Languages, 1996.
- [3] 이재호, "에이전트 시스템의 연구 및 개발 동향," 정보과학회지, 제18권 5호, PP. 4-5, 2000.5.
- [4] Bran Selic, Garth Gullekson, Jim McGee, Ian Engelberg, "ROOM: An Object-Oriented Methodology for Developing Real-Time Systems," Bell-Northern Research Ltd., Ottawa, CANADA, July, 1992.
- [5] Microsoft, http://msdn.microsoft.com/library/wcedoc/wcemfc/compare_32.htm
- [6] 홍성수, "정보가전을 위한 실시간 운영체제 및 미들웨어," 정보처리학회지, 제8권 1호, 2001. 1.
- [7] K. Arnold, B. Osullivan, R. Scheifler, J. Waldo, and A. Wollrath. The Jini Specification. Addison Wesley Publishing Co. Jul. 1999.
- [8] Nick Grattan, Marshall Brain, Windows CE 3.0, Prentice Hall PTR, pp. 185-235, 2001.
- [9] Michael Barr, Programming Embedded Systems, O'Reilly & Associates, Inc., pp.20-24, 1999.