

HUMAN : 분산 프로그래밍을 위한 이동에이전트 플랫폼

강미연⁰, 문미경, 유전아, 정원호
 덕성여자대학교 컴퓨터과학부
 (mykang⁰, whchung)⁰@center.duksung.ac.kr

HUMAN : A Mobile Agent Platform for Distributed Programming

Mi-Yeon Kang⁰ Mi-Kyung Moon, Kyeon-Ah Yu and Won-Ho Chung
 School of Computer Science, Duksung Women's University

요 약

이동 에이전트는 분산 응용에 있어서 네트워크의 부하와 대기시간을 줄일 수 있는 좋은 기술 중 하나이며, 다양한 네트워크 환경에 대한 유연성과 보안성 좋아 향후 기대되는 분산 응용 기반 기술이라 할 수 있다. 본 논문에서는, 이동 에이전트 기반의 다양한 분산 응용에 효율적으로 적용할 수 있는 이동 에이전트 플랫폼인 HUMAN이 설계 구현되었다. 이동 에이전트 플랫폼, HUMAN은 파일 탐색, 그룹 어드레싱, 다양한 여정 모드, 유연 응답 기능 등의 고급 유틸리티 등을 지원하고 있어, 에이전트 프로그래밍에 있어서 높은 수월성을 제공하고 있으며, 불법파일 탐색 등과 같은 분산 응용에 특히 유용할 수 있다. 또한 에이전트 등록, 해지 등을 위한 관리 서버 등을 제공하고 있어, peer-to-peer 관련 분산 응용 시스템의 개발에도 용이하게 적용할 수 있다는 장점을 가지고 있다.

1. 서론

이동 에이전트는 네트워크 환경의 다양한 분산 응용을 위해 비교적 쉽게 대처할 수 있을 만큼, 통합성과 유연성이 좋아, 향후 기대되는 분산 응용 기술 중의 하나라고 할 수 있다. 이동 에이전트의 장점은 네트워크 부하(load)와 지연(latency)의 감소, 네트워크 오류 시 다른 시스템에 비해 신뢰도가 높다는 것을 들 수 있다[1]. 왜냐하면, 이동 에이전트는 일단 목적하는 노드로의 이동이 이루어지고 나면, 결과물 응답 전까지, 대부분의 작업이 해당 노드에서 로컬하게 수행되므로, 네트워크의 비접속 상태에서도 작업 수행이 가능하며, 작업 수행이 완료된 후 접속을 시도하여 결과물을 응답하기 때문이다. 특히, Java 기반의 이동 에이전트는 Java 언어의 폭넓은 사용과 더불어, 그 응용의 폭이 넓어, 다양한 형태의 분산 정보 관리[2], 병렬 및 분산처리[3], 정보의 전송 및 공유[4], SCM(Supply Chain Management)과 같은 전자상거래 분야에까지 그 응용 범위를 넓혀가고 있다[5-7].

본 논문에서는, 네트워크 상에서의 효율적인 분산 응용을 위해, Java 기반의 이동 에이전트 플랫폼인 HUMAN(High Utility Mobile Agent Network)을 설계, 구현하였다. HUMAN은 이동 에이전트 기반의 다양한 분산 응용에 효율적으로 적용할 수 있는 이동 에이전트 플랫폼으로, 분산 환경에 쉽게 대처할 수 있도록 유연성을 높이고, 손쉽게 에이전트 프로그래밍이 가능하도록, 고급의 유용한 기능을 지원하고 있다. 또한, 간단한 시스템 구조를 갖도록 하여, 원하는 이동 에이전트 응용에 손쉽게 적용할 수 있도록 하는 것을 목표로 하여 설계, 구현되었다.

본 논문의 구성은 다음과 같다. 2장에서는 HUMAN의 구성과 동작 및 메시지 구조에 대해 기술되며, 3장에서는 HUMAN의 주요 특징에 대해 기술된다. 그리고, 마지막으로 결론과 향후 연구가 기술된다.

2. HUMAN 플랫폼 구성 및 메시지 구조

HUMAN은 다양한 분산 응용에 효율적으로 적용하고, 분산 환경에 쉽게 대처할 수 있도록 유연성을 높이고, 손쉽게 프로그래밍이 가능하도록, 고급의 유용한 기능을 지원하고 있다.

2.1 플랫폼의 구성

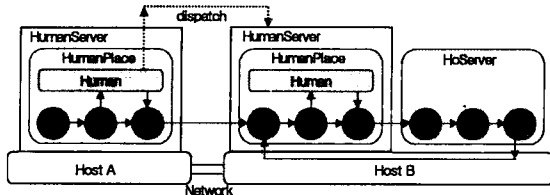
HUMAN은 hoserver, server, utility, 그리고 agents 4개의 패키지과 총 27 개의 클래스로 구성되어 있으며, 150여 개의 활용 가능한 메소드를 제공하고 있다. utility 패키지는 이동 에이전트 응용을 위한 기본 클래스로 구성되어 있으며, 분산 환경에서 에이전트 응용을 위한 다양하고 유용한 기능들을 지원하고 있다. 그리고 server, hoserver 패키지는 에이전트의 제어 및 관리를 위한 클래스들로 구성되어 있으며, agents 패키지는 HUMAN을 이용한 응용 에이전트 클래스들로 구성되어 있다. 패키지과 클래스의 구성은 (표-1)과 같다.

(표-1) HUMAN 플랫폼 구성

패키지	클래스
hoserver	HoServer, HoModel, HoModelEvent, HoModelListener
server	HumanServer, MenuPil, MenuPrI, MenuFile, RadioIcon
utility	Human
	HumanFile, HumanList, HumanTime
	HumanAddress
	HumanMessage, HumanMessageEvent,
	HumanMessageListener
	HumanPlace, HumanPlaceEvent, HumanPlaceListener
	HumanEvent, HumanListener
agents	HumanBox, HumanClassLoader, HumanClassManager
	HumanInputStream, HumanOutputStream
	IllegalFileSearch, FileSearchAgent

HUMAN은 네트워크 상에 분산된 각 에이전트의 등록, 동작 여부, 공유 환경 등에 관한 정보를 수집하고, 관리하는 HoServer(Human On-line Server), 에이전트의 생성 및 전송, 그리고 감시 등을 위한 사용자 인터페이스를 제공하는 HumanServer, 그리고 에이전트들간의 메시지 통신 및 에이전트의 수행 환경 역할을 하기 위한 HumanPlace 프록시를 제공하고 있다. 위 서버들을 통해 HUMAN 에이전트의 동작과 통신이 이루어질 뿐만 아니라, 분산된 에이전트들의 관리를 가능

하게 해줌으로써, 예를 들면, 분산 파일 공유, 메신저 등과 같이 peer-to-peer 관련 분산 응용에 용이하게 적용할 수 있도록 하고 있다. HUMAN은 서버의 환경 설정 및 보안을 위한 특정 파일로 server.config, hoserver.config 파일을 제공하고 있다. HUMAN 서버의 구성은 (그림-1)과 같다.



(그림-1) HUMAN 서버의 구성

2.2 메시지의 구조

서버 간 또는 에이전트들 사이의 모든 통신은 메시지를 통해 이루어진다. 메시지의 송수신 및 처리는 (그림-1)과 같이 HoServer와 HumanPlace에 의해 ReceiveMessageThread, ListenThread, 그리고 SendMessageThread가 생성, 운용되어 이루어진다. 그러나, HoServer의 경우는 활성화된 HumanPlace들의 등록 혹은 그들의 종료 시 해지를 위한 메시지 처리에 국한되어 사용되고, 그 외의 경우에는 HumanPlace를 통해서 모든 통신이 이루어진다.

HUMAN에서 정의되어 사용되는 메시지는 sender, receiver, kind, content, result, binary로 구성된다. sender와 receiver는 메시지를 송수신하는 노드의 이름(혹은 IP 주소)과 송수신 시에 사용하는 포트 번호, 송수신하는 에이전트의 이름으로 구성된 HumanAddress 객체이다. 메시지의 kind는 의미를 가지는 메시지 유형을 나타내며, SEL, ACKS, RESULT, POLL, ACKP, HUMAN, GET, CLASS, GET_CLASS, PLACE_ONLINE, PLACE_OFFLINE, PLACES으로 그 용도에 따라 12 가지가 정의되어 사용되고 있다. content, result, binary는 메시지에 포함될 내용으로, 각각 String, Object, byte[] 형으로 정의되어 있다. 정의된 메시지 유형과 그에 따른 각 메시지의 구성은 (표-2)과 같다.

(표-2) HUMAN에서 사용되는 메시지의 구성

구분	kind	content	result	binary
결과물 응답 시 사용하는 메시지	SEL	null	null	null
	ACKS	null	null	null
에이전트와 클래스의 이동을 위한 메시지	RESULT	result kind	Result	binary
	POLL	null	null	null
	ACKP	null	null	null
	HUMAN	agent name	null	value
	GET	agent name	null	null
	CLASS	class name	null	binary
HumanPlace의 온라인/오프라인 상태 전송을 위한 메시지	GET_CLASS	class name	null	null
	PLACE_ONLINE	null	null	null
	PLACE_OFFLINE	null	null	null
	PLACES	agent name	null	null

3. HUMAN의 특성

HUMAN은 분산 응용을 위한 고급 기능들을 제공함으로써, 응용 프로그래밍의 수월성을 높이고, 또한 이동성 및 네트워크

상황을 고려한 다양한 기능을 지원하여, 사용자가 상황에 맞춰, 적절하게 사용할 수 있도록 높은 선택성을 제공하고 있다.

3.1 프로그래밍의 수월성 제공

분산 응용을 위해 보편적이면서도 주요한 기능으로, 분산 정보 탐색 및 수집을 들 수 있는데, 이를 위한 기본 기술로, 주어진 특정 파일 혹은 파일 리스트 기반의 파일탐색 기능, 주어진 노드들의 파일 전송 기능, 그리고 탐색할 노드들의 주소 설정 기능 등을 들 수 있다. HUMAN은 이러한 기능들을 위한 인터페이스를 제공하고, 내장 API(메소드)로 지원하고 있어, 사용자가 필요시 손쉽게 사용할 수 있도록 하고 있다.

특히, 분산 정보의 탐색을 위해, 네트워크 상에 분산되어 있는 노드들을 대상으로 탐색 범위를 설정하는 2가지 방법을 탐색 주소 설정 기능으로 지원하고 있다. 하나는 개개의 노드 단위로 주소를 지정하는 경우로, 특정 노드들을 대상으로 작업을 수행할 경우 사용될 수 있는 주소 설정 방법이다. 또 다른 하나는 탐색 대상 노드들을 개별적 지정하는 데는 무리가 따르게 되는 경우 그룹 기반으로 탐색 대상 노드들의 주소를 설정할 수 있도록 하는 설정 방법이다. 이를 위해 HUMAN에서는 서브넷 기반의 노드들을 그룹 단위로 주소를 지정할 수 있도록 하고 있다. 예를 들면, 203.252.212.240부터 203.252.212.249까지 모든 노드들을 탐색 범위로 설정하고자 할 경우, 203.252.212.24x 하나의 주소로 대상 노드들을 설정할 수 있도록 하고 있다. 이는, 네트워크 상에서 광범위한 정보 탐색 및 액세스를 요하는 응용에서 유용하게 사용될 수 있다.

3.2 다양한 여정모드의 제공

HUMAN은 에이전트 이동 시점에서 고려될 수 있는 상황에 대해 유연하게 대처할 수 있도록, 다양한 여정 방식을 제공하고 있다. 단순 모드 여정(Simple-Mode Itinerary, SMI)과 다중 모드 여정(Multi-Mode Itinerary, MMI)이 그것들이며, 이들은 가져야할 노드들의 주소가 우선순위로 기록된, 우선순위 여정 리스트(Priority Itinerary List, PIL)에 기반을 두고 있으며, 항상 하나 이상의 주소를 포함하고 있다. 이러한 에이전트의 여정은 목적 노드들의 디스패치 방식에 따라 결정되며, HUMAN은 단순 디스패치(Simple Dispatch), 동적 디스패치(Dynamic Dispatch) 그리고 다중 디스패치(Multiple Dispatch) 3가지 디스패치 방식을 제공하고 있다. 단순 디스패치 방식은 SMI를 지원하고 있으며, 동적 디스패치와 다중 디스패치는 MMI를 지원하기 위해 제공되고 있다. 디스패치 방식에 대해 사용자가 특별히 지정하지 않는 경우 HUMAN은 동적 디스패치를 사용하는 MMI로 간주한다. 사용자는 HUMAN이 제공하는 인터페이스를 통해 단순히 자신의 상황에 필요한 디스패치 방식을 선택하고, PIL을 구성하기만 하면 된다. 그리고, 위에 기술된 3가지 디스패치 방식은 doItinerary()라는 하나의 API의 호출을 통해 이루어진다.

단순 디스패치: 단순 디스패치는 SMI의 경우를 위한 디스패치 방식이다. PIL 상의 특별히 정한 노드들의 이동인가 혹은 PIL 상의 이동 대상 노드들 중 어느 한 노드들의 이동인가는 PIL에 포함된 노드들에 의해 결정된다. PIL상의 첫 번째 엔트리에 지정된 노드가 정상 동작한다면, SMI의 경우 모두 PIL상의 첫 번째 노드로 이동이 일어나게 되고, 그렇지 않은 경우 PIL상의 노드들을 차례로 조사하여 이동이 가능한 첫 번째 노드로 이동이 일어나는 전략을 취하고 있다.

동적 디스패치: 동적 디스패치는 PIL에 기술된 노드들에 대

해 순차적으로 이동 가능성을 조사하면서 이동 가능한 노드로 스스로 먼저 이동하여, 주어진 작업을 마친 후 다음 노드로 이동한다. 이때, 이동 가능성 조사는 타이머를 기반으로 하고 있다. 먼저 이동 가능한 노드를 이동 대상 노드로 선정하여, 에이전트를 이동하기 때문에 주어진 여정에 따른 효율적인 에이전트의 이동이 이루어질 수 있다는 장점을 가진다.

다중 디스패치: 다중 디스패치는 이동 대상 노드가 2개 이상이며, 이동 대상 범위는 여정리스트 상의 모든 노드들인 경우이다. 다중 디스패치의 경우는 PILL상의 모든 노드들에게 POLL 메시지를 전송한 후 설정된 시간 내에 수신된 ACK 메시지를 송신한 모든 노드들로 이동을 하며, 에이전트의 작업 수행도 동시에 이루어진다.

3.3 다양한 응답모드의 계층

이동 에이전트는 자신에게 주어진 작업을 원격지 노드에서 수행하고 그 결과를 응답하여야 작업을 완료했다고 할 수 있으며, 다음 노드로 이동하여 작업을 시작할 수 있다. 결과물 응답 시점에서 고려될 수 있는 상황으로 여러 가지가 있을 수 있는데, 1) 부담스러운 결과물의 크기, 2) 작업 유형에 따른 결과물에 대한 다른 방식의 응답 필요, 3) 결과물 응답 대상 노드와의 접속 불가능 상황, 그리고 4) 여정을 맞추기 위한 작업 에이전트의 신속한 이동의 필요, 등이 있을 수 있으며, 다른 한편으로, 원격 작업에 대한 수행 여부의 파악이 있을 수 있다. HUMAN은 이러한 상황에 유연하게 대처할 수 있도록, 3가지 응답 기능으로 구성되는 동적 응답 기능을 제공하고 있으며, 응답 대상 노드들이 우선순위로 지정된 우선순위 응답 리스트(Priority Reply List, PRL)에 기반을 두고 있다. 이러한 응답 기능은 적시에 정보를 얻어내는 것이 중요하며, 또한 협동 작업의 유형이 많아서, 결과물을 응답해야 할 노드가 서로 다른 분산 응용의 경우 유연하게 대처할 수 있는 특징을 가지고 있다. HUMAN은 고정응답모드(Fixed Reply Mode), 동적응답모드(Dynamic Reply Mode), 그리고 다중응답모드(Multiple Reply Mode) 등 3가지 방식의 응답 기능을 지원하고 있다.

고정 응답: 고정 응답모드(FRM)는 응답 대상 노드가 특정한 하나의 노드로 지정되는 경우의 응답 형식이다. 그 특정 노드는 PRL 상에서 우선순위가 가장 높은, 즉, PRL의 첫 번째 엔트리에 지정된 노드이다. FRM의 경우 응답 대상 노드와의 접속이 가능하지 않으면, 가능할 때까지 결과물의 전송이 일어나지 않으며, 동시에 다음 노드로의 이동도 일어나지 않아, 작업 지연이 커질 수 있다는 단점이 있으나, 에이전트의 작업 수행 및 이동 여부를 파악하는 데는 유용한 응답 기법이다.

동적 응답: 동적 응답모드(DRM)는 우선 순위 차례로 접속 가능성을 조사하면서 가장 먼저 접속 가능한 노드를 응답 대상으로 선정하여 결과물을 전송한다. DRM은 설정된 시간 이내에 접속이 가능하지 않으면 다음 순위의 노드를 응답 대상으로 하여 접속 가능성을 조사하는 과정을 반복하면서, 가장 먼저 접속 가능한 노드를 응답 대상으로 선정하여, 결과물을 전송하기 때문에 응답 지연을 줄여 에이전트의 이동이 신속하게 이루어지게 할 수 있다는 장점을 가진다.

다중 응답: 다중 응답 모드(MRM)는 응답 대상 노드가 2개 이상이며 응답 대상 범위는 PRL 상의 모든 노드들인 경우이다. MRM의 경우는 PRL 상의 모든 노드들에 대해 한꺼번에 접속 가능성을 조사한다. 즉, PRL 상의 모든 응답 대상 노드들에게 SEL 메시지를 전송한 후 타이머를 구동시킨다. 그리하여 설정된 시간 내에 수신된 ACK 메시지를 송신한 모든 응답 대

상 노드들에게 결과물을 응답한다. 그러나, 설정된 시간 내에 k 개 이상의 ACK 메시지를 수신하지 못하게 되면 타이머는 재 구동되고 전과 같은 동작이 반복되게 된다.

본 연구에서 설계 구현된 HUMAN 플랫폼과 일반적으로 많이 사용되는 이동 에이전트 플랫폼인 AGLETS과, AJANTA와의 요약 비교가 (표-3)에 나타나 있다.

(표-3) AGLETS, AJANTA, HUMAN의 주요 기능 비교

주요 기능	AGLETS	AJANTA	HUMAN
고급 유틸리티	NO	NO	YES
유연한 여정 모드	YES	YES	YES
유연한 응답 모드	NO	NO	YES
에이전트 관리서버	NO	YES	YES
프록시 서버	YES	YES	YES
에이전트 생성과 전송을 위한 UI 서버	YES	YES	YES

중 더 자세한 사항에 대해서는 [7]의 Programmer's Guide와 System Document를 참조하기 바란다.

4. 결론 및 향후연구

이동 에이전트는 여러 가지 분산 응용에서 네트워크 부하와 지연을 극복할 수 있는 기법 중 하나로 최근 활발히 연구되고 있는 새로운 패러다임이다. 본 논문에서는 네트워크 상에서의 효율적 불법파일 탐색을 위한 이동 에이전트 시스템인 HUMAN이 설계, 구현되었다. HUMAN은 간단한 구조, 유용하며 풍부한 메소드, 그리고 유연성 있는 주소 지정 방법, 그리고 응답 방법 등을 제공하고 있어, 이동 에이전트 기반의 분산 응용에 쉽게 응용될 수 있다. HUMAN 시스템이 지원하는 기능을 바탕으로 기타 다른 응용에도 쉽게 대처할 수 있는데, 좋은 분야가 개인 대 개인(peer-to-peer) 정보 교환을 들 수 있다. 향후 HUMAN 시스템은 이러한 개인 대 개인 정보 통신 분야로 적용해 나갈 것이다.

참고문헌

- [1] D. B. Lange and M. Oshima, "Programming And Deploying Java Mobile Agents with Aglets," Addison Wesley, 1998
- [2] D. D. Roure et al., "Agents for Distributed Multimedia Information Management," Proc. of 1st Int'l Conf. on the Practical Application of Intelligent Agents and Multi-agents Technology, April, 1996.
- [3] L. M. Silva et al., "Using Mobile Agents for Parallel Processing," Project Report, Dept. of Engineering Information, Univ. of Coimbra, Portugal, 1998.
- [4] N. Minar et al., "Cooperating Mobile Agents for Dynamic Network Routing," MIT Media Lab., 1999.
- [5] N. Ivezić et al., "Agent-based Technologies for Virtual Enterprises and Supply Chain Management," A draft version for submission, IEEE Internet Computing, CTRC, Oak Ridge National Lab., 1999.
- [6] D. Brugali et al., "Inter-Company Supply Chains Integration via Mobile Agents," in The Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility, and the Virtual Enterprise, Kluwer Academic Pub., 1998.
- [7] 정원호, HUMAN 0.9.0 Programmer's Guide/System Documents, "http://www.duksung.ac.kr/~whchung/