유비쿼터스 환경에서 사용자 맞춤형 디바이스 협업 시스템*

박원익0, 이용대, 김영국
충남대학교 공과대학 컴퓨터공학과
{wonik780, racunda, ykim}@cnu.ac.kr

Personalized Device Collaboration System in Ubiquitous Environment

Won-Ik Park0, Yong-Dae Lee, Young-Kuk Kim
Department of Computer Engineering, Chungnam National University

요 약

오늘날 보이지 않는 수많은 장치들과 소프트웨어들이 서로 연결되어 각각의 사용자들에게 편리한 서비스를 제공하는 유비쿼터스 환경에서 모바일 단말기는 유저와 서비스간의 매개체 역할을 한다. 하지만 자원이 제한적인 모바일 단말기의 특성상 사용 시에 많은 제약이 있다. 이러한 문제를 해결하기위해 가용한 주변의 장치 자원을 연결하여 사용할 수 있도록 표준 인터페이스를 제공하는 기술들이 있다. 하지만 동적으로 변화하는 사용자의 선호도 정보와 장치 자원 정보와 같은 프로파일 정보를 효율적으로 적용하지 못하여 사용자 맞춤형 디바이스를 제공하지 못하는 문제점이 있다. 본 논문에서는 유비쿼터스 환경에서 이러한 프로파일 정보를 효율적으로 적용하여 사용자 맞춤형 자원 공유 서비스를 제공하는 시스템을 제안한다.

1. 서 론

최근 정보 통신 기술의 발전과 빠른 보급으로 다양한 모바일 단말기를 통해 다양한 컴퓨팅 자원을 언제 어디서나 누구든지 각종 단말기와 장치들을 통해서 광대역 네트워크에 접속하여 서비스를 받을 수 유비쿼터스 환경이 보편화 되었다. 유비쿼터스 환경은 보이지 않는 수많은 장치들과 소프트웨어들이 서로 연결되어 각각의 사용자들에게 편리한 서비스를 제공한다. 이러한 서비스를 제공 받기 위해서는 유저와 서비스간의 매개체 역할을 하는 모바일 단말기가 필요하다. 하지만 자원이 제한적인 모바일 단말기의 특성상 다양한 기능을 이용할 수는 없다. 따라서 본 논문에서는 제한적인 자원 문제를 해결하기 위해서 주변의 다양한 자원을 실시간으로 공유하는 사용자 맞춤형 디바이스 협업 시스템을 제안한다.

이러한 모바일 단말기의 제한적인 자원 문제를 해결하기 위해서 데이터를 필터링하여 상황에 맞는 데이터만을 제공해주는 정보 필터링 방법과 직접 다른 디바이스의 자원을 사용하여 해결하려는 방법이 있다. 하지만, 동적으로 변화하는 사용자 선호도 정보 및 자원 정보와 같은 프로파일 정보를 효율적으로 적용하지 못하는 실정이다.

예를 들면 문서편집 작업을 원하는 A라는 사람은 현재 모니터와 키보드 마우스 기능이 없는 제한적인 자원을 갖는 모바일 단말기를 소유한 상태이다. A라는 사람이 문서편집 작업을 하기 위해 유비쿼터스 환경에 도착한다면 제안한 시스템은 문서편집을 수행하기 위해 필요한 장치들을 분석하여 A에게 이용 가능한 장치들을 추천한다. 이때 A의 프로파일 정보를 이용하여 주변의 이용 가능한 장치들 중에서 사용자에게 맞는 모니터, 키보드, 마우스를 공유 할 수 있도록 서비스 한다면 모바일 단말기의 작은 화면과 불편한 입력장치 및 포인팅장치와 같은 제한적인 자원 문제를 보완할 수 있다.

모바일 단말기를 이용하여 위와 같은 자동적인 서비스를 구성하기 위해서는 서비스 요청 분석, 서비스에 해당하는 사용 가능한 디바이스 검색, 추론엔진, 사용자의 선호 정보 등을 알아야 한다. 그리고 이들은 명료하게 정의된 형식과 의미를 제공하는 온톨로지로 표준에 맞추어 기술하고 필요한 지식들을 규칙으로써 정의하였다. 그리고 이렇게 정의된 온톨로지와 규칙들을 이용하여 룰 기반 추론을 통한 서비스 요청분석을 수행하고 사용 가능한 디바이스 검색을 지원한다. 또한, 가중치 기반의 추천 알고리즘을 통해 사용자 맞춤형 자원을 추천한다.

본 논문은 다음과 같이 구성된다. 2장에서는 네트워크에 연결되는 기기들을 발견하고 관리하며, 사용자가 이들을 제어할 수 있도록 인터페이스를 제공하는 역할을 수행하는 UPnP, Jini, Havi 기술에 대해 알아보고 3장에서는 본 논문에서 제안한 사용자 맞춤형 디바이스 협업 시스템 구조를 살펴본다. 4장에서는 본 논문에서 구성한 온톨로지에 대해서 설명한다. 마지막으로 5장에서는 결론 및 향후 과제를 제시한다.

2. 관련 연구

디바이스들의 협력이 필수적인 유비쿼터스 환경에서 UPnP, JINI, HAVi와 같은 기술은 디바이스들간의 통신을 위해 제안된 대표적인 기술이다.

UPnP(Universal Plug and Play)는 정보가전, 무선통신장치, PC 관련 장비 등 여러 장소에 분산되어 있는 장치와 서비스 간의 쉽고 편리한 통신방법을 제공하고자 탄생하였다[1-2]. UPnP의 특징으로는 소규모에서 대규모의 네트워크로 확장이 용이하고, PnP를 지원하여 장비의 접속과 분리를 자동으로 인지하며, 개발이 용이하고, 작은 자원으로도 이용이 가능하며, 가전장비와 같이 IP가

* 본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅및네트워크원천기반기술사업의 08B3-O1-30S 과제로 지원된 것임.

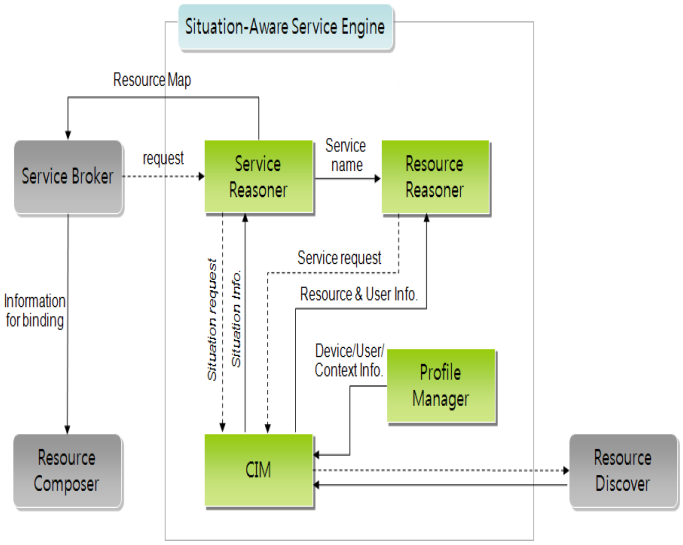
없는 장비에 대해서는 단순한 기능을 가진 SCP라는 프로토콜을 통하여 브릿지(Bridge: 네트워크 프로토콜 변환기)로 연결할 수 있도록 지원한다.

JINI는 Sun Microsystems에서 개발한 미들웨어로서 자바를 기반으로 하여 다양한 방식으로 네트워크에 접속된 장치나 소프트웨어를 동적으로 상호 작용하도록 하게 하는 기술이다[3-4]. 자바 환경에서 동작하므로 운영체제나 기타 하드웨어 플랫폼에 구애받지 않는다. JINI는 분산 환경에서 각각의 장치 또는 소프트웨어간의 상호운용을 위한 방법으로 정의되었을 뿐만 아니라, 자바 소프트웨어 기반의 기술이기 때문에 소프트웨어 서비스를 제공하는데 좀 더 적합한 구조를 갖고 있다.

HAVi(Home Audio Video Interoperability)는 가정 내의 오디오 및 비디오 가전 기기간의 상호운영성(장비간의 통신 및 제어 기능 등)을 위한 홈 네트워크용 표준으로 정의된다.

이러한 UPnP, JINI 및 HAVi 기술은 디바이스의 공유를 위한 기술이라는 점에서는 제안한 시스템과 비슷하다. 하지만, 동적으로 변화하는 사용자의 선호도나 자원에대한 정보를 고려하지 않기 때문에 각각의 사용자에게 상황에 적합한 디바이스를 자동으로 추천하지는 못한다는 점에서 제안한 시스템과 다르다.

3. 사용자 맞춤형 디바이스 협업 시스템 구조



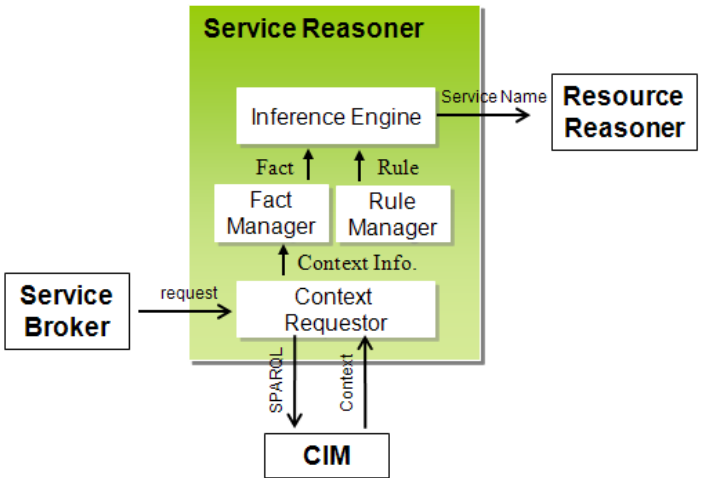
[그림 1] 사용자 맞춤형 디바이스 협업 시스템 전체 구조

사용자 맞춤형 디바이스 협업 시스템은 [그림 1]과 같이 Service Broker, Service Reasoner, Resource Reasoner, Profile Manager, Resource Composer, CIM, Resource Discover로 구성된다. Service Broker는 사용자의 서비스 요청 및 결과를 처리하는 역할을 하며 Service Reasoner는 사용자의 서비스 요청을 받아 현재 사용자에게 필요한 서비스(예를 들면, 문서편집, 프레젠테이션)를 분석하여 Resource Reasoner의 입력값으로 전달한다. 그리고 Resource Reasoner는 Service Reasoner로부터 전달 받은 서비스 실행을 위해 필요한 자원 리스트를 추천 해준다. 그리고 Profile Manager는 사용자의 선호도 및 자원 정보를 수집, 가공하여 CIM에 보내는 역할을 한다. CIM은 Profile Manager로부터 얻은 Profile 정보를 저장하여 Service Reasoner에서 컨텍스트에 대한 검색 요청이나 Resource Reasoner에서 자원이나 유저에 대한 검색 요청이 있을 때 정보를 제공해주는 역할과 저장된 프로파일 정보의 업데이트 기능을 수행한다. Resource Discover는 이용 가능한 외부의 자원을 탐색한다. 마지막으로, Resource Composer는 추천된 자원의 실제 바인딩을 위한 작업을 수행한다.

본 논문에서는 [그림 1]에서 Service Reasoner, Resource Reasoner, CIM, Profile Manager로 구성된 Situation-Aware Service Engine을 중심으로 설명한다.

3.1 Service Reasoner

Service Reasoner의 역할은 사용자의 현재 상황을 인지하여 현재 사용자가 필요한 서비스를 찾는 것이다. 룰 기반의 추론엔진인 JESS를 이용한다. 이를 위해서 Service Reasoner는 질의를 통해 CIM에 저장되어 있는 사용자의 스케줄 정보와 시간과 위치정보를 Fact값으로 얻어온다. 여기에 정의된 룰을 적용하여 최종적으로 사용자에게 필요한 서비스를 찾게 된다. 추론된 서비스명은 Resource Reasoner의 입력값으로 넘겨진다.



[그림 2] Service Reasoner 구조

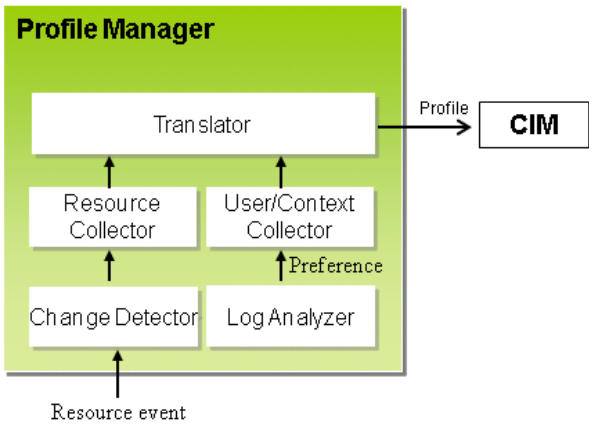
본 논문에서 추론을 위한 모듈로 사용되는 Inference Engine(JESS)은 Sun Microsystems에서 제작한 룰 기반의 전문가 시스템 저작 도구이다. JESS는 본래 CLIPS 전문가 시스템 저작 도구에서 유래되었고, 온톨로지와 룰을 기반으로 추론한다. 본 논문에서는 JESS를 사용하여 정의한 온톨로지와 룰을 입력으로 현재 상황을 추론한다.

3.2 Profile Manager

Profile Manager는 Service Reasoner와 Resource Reasoner에서 필요한 정보인 프로파일 정보의 생성 및 관리하는 역할을 한다. 프로파일 정보에는 자원관련 정보와 사용자의 선호도 정보를 포함한다. 동적으로 변화하는 자원 정보의 갱신을 위해 Change Detector 모듈을

두어 가용한 로컬 자원 정보를 관리한다.

Profile Manager에서 관리하고 있는 정보들은 JENA의 RDF Writing Function을 이용하여 RDF 형태로 변환되어 CIM에 전달 저장된다.



[그림 3] Profile Manager 구조

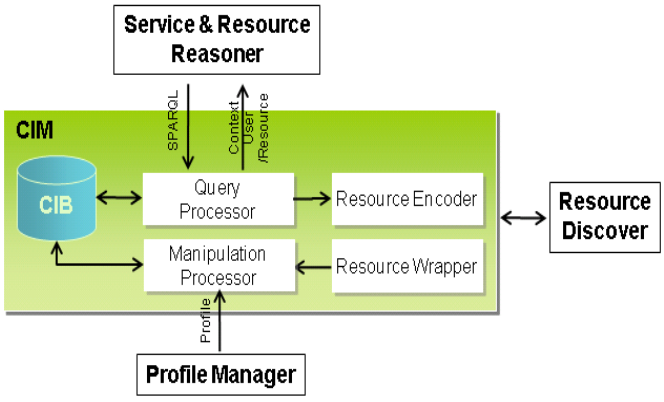
3.2.1 Resource Collector

모바일 단말기의 자원 상태 정보를 수집한다. 즉, 모니터, 키보드, 마우스와 같은 자원의 속성 정보를 수집한다.

3.2.2 User/Context Collector

사용자 로그 기록을 분석하여 사용자의 자원 선호정보와 시간과 위치정보 같은 컨텍스트 정보를 수집한다.

3.3 CIM (Context Information Management)



[그림 4] CIM 구조

CIM은 로컬 디바이스 정보 및 주변의 가용한 디바이스에 대한 상세 정보를 포함하며 각 모바일 단말기들이 제공하는 공유 자원들에 대한 속성 및 부가기능에 대한 상세 기술 정보들을 포함한다. 또한 이러한 정보들은 동적으로 추가, 삭제 등의 변경 작업이 일어나며 다양한 형태의 자원들에 대한 세부 정보를 표현한다. 또한 이러한 세부 정보에 대한 편리한 질의 방법이 제공되어야 한다. 이러한 다양한 표현력, 확장성에 대한 요구에 의해 CIM은 JENA를 이용하여 RDF 형태로 저장되는 CIB를 구성하며 정보에 대한 질의는 SPARQL을 사용한다. CIB에 저장되는 정보의 형태는 RDF 스키마로 정의된다. 정의한 온톨로지를 기반으로 디스플레이 자원을 검색하기 위한 SPARQL 질의의 예제는 아래와 같다.

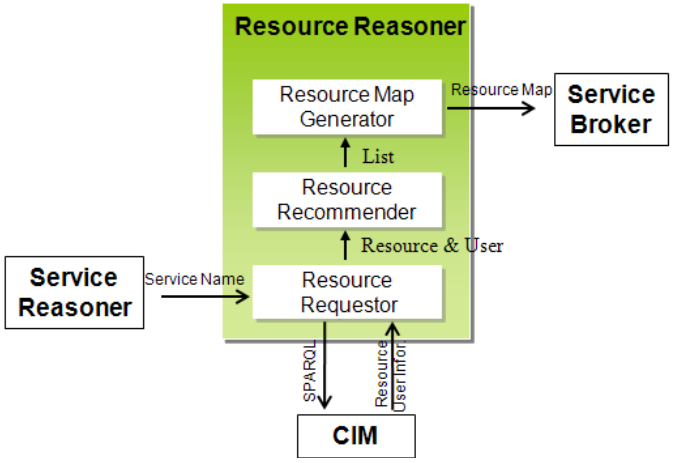
```
PREFIX umo:http://www.keti.re.kr/ubicom/umo#>
PREFIX ns:http://www.w3c.org/1999/02/22-rdf-syntax-ns#>
SELECT ?x
WHERE {
        ?x ns:type umo:Monitor.
}
```

CIB에는 자원에 대한 정보가 다음과 같은 RDF 형태로 저장 되어있다.

```
<umo:Monitor rdf:about="&umo;monitor_01"
   umo:ID="monitor_01"
   umo:Monitor_Respond_Time="200"
   umo:Monitor_brightness="900"
   umo:Monitor_horizon_resolution="1024"
   umo:Monitor_is_CRT="false"
   umo:Monitor_is_LCD="true"
   umo:Monitor_size="20"
   umo:Monitor_vertical_resolution="1024"
   umo:annotation="monitor_04"
   umo:model="monitor_01"
   rdfs:label="monitor_01">
<umo:IsLocalResourceOfrdf:resource="&umo;Loc
alResource_01"/>
</umo:Monitor>
<umo:Monitor rdf:about="&umo;monitor_02"
   umo:ID="monitor_02"
   umo:Monitor_Respond_Time="100"
   umo:Monitor_brightness="300"
   umo:Monitor_horizon_resolution="2048"
   umo:Monitor_is_CRT="true"
   umo:Monitor_is_LCD="false"
   umo:Monitor_size="18"
   umo:Monitor_vertical_resolution="2048"
   umo:annotation="monitor_02"
   umo:model="monitor_02"
   rdfs:label="monitor_02">
<umo:IsLocalResourceOfrdf:resource="&umo;Loc
  alResource_01"/>
</umo:Monitor>
```

3.4 Resource Reasoner

Resource Reasoner는 Service Reasoner를 통해 얻은 서비스명을 기반으로 서비스 수행 시 필요한 디바이스를 사용자의 성향에 맞게 추천 해주는 역할을 한다. Resource Requestor모듈이 CIM에 저장되어 있는 서비스에 관련된 자원과 사용자의 선호도 정보를 SPARQL을 통해 얻어온다. 이 정보를 이용하여 사용자의 성향에 맞는 디바이스를 추천하여 Service Broker에게 전달한다.



[그림 5] Resource Reasoner 구조

다수의 디바이스 중에서 사용자의 성향을 고려한 디바이스를 추천 하기위한 모듈인 Resource Recommender는 다음과 같은 방법으로 추천을 한다. Profile Manager의 Log Analyzer 모듈을 통해 전달받은 디바이스 속성에 대한 우선순위와 유저의 디바이스에 대한 선호 속성값을 이용한다. 예를 들면, 어느 한 사용자 A의 디바이스 속성에 대한 우선순위는 브랜드, 크기, 밝기 순이라 하고 사용자의 선호하는 속성값은 삼성, 19인치, 300cd라 하면 실제 검색된 세 개의 다른 속성값을 갖는 모니터들의 속성값과 사용자의 선호하는 속성값을 비교하여 [표 1]처럼 나타 낼 수 있다. 모니터들에 대한 속성값에 대한 점수는 선호하는 속성값과 일치하면 1의 값을 주고 그렇지 않으면 0의 값을 부여한다.

[표 1] 속성값 점수 부여

	모니터1	모니터2	모니터3
브랜드	1	0	0
크 기	0	1	1
밝 기	1	1	0
합 계	2	2	1

하지만 [표 1]은 속성값의 우선순위를 반영하지 못하는 문제점이 있다. 본 논문에서는 속성값의 우선순위에 따른 가중치 w를 다음과 같이 정의 하여 추천 정확도를 높였다.

$$w(j) = \frac{N - R_j + 1}{\sum_{j=1}^{N}(N - R_j + 1)}$$

여기에서, w(j)는 j속성에 대한 가중치를 의미하고, N은 디바이스의 속성 개수를 의미하며 R_j는 j속성에 대한 우선순위를 의미한다. [표 1]에 가중치를 부여하면 [표 2]와 같다.

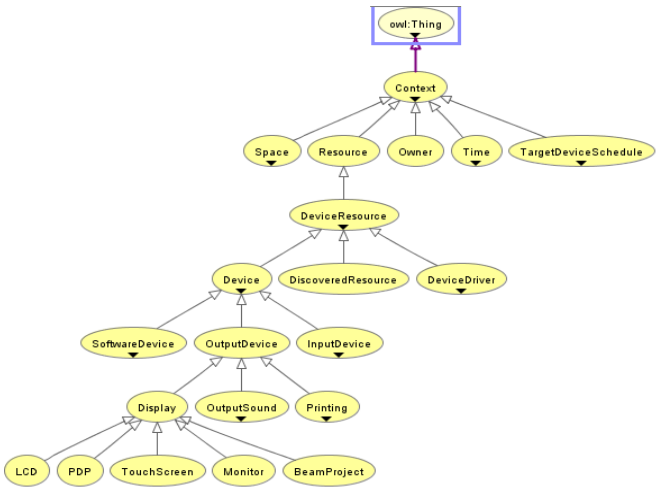
[표 1] 가중치 부여

	모니터1	모니터2	모니터3
브랜드	1*0.5	0	0*0.5
크 기	0*0.333	1*0.333	1*0.333
밝 기	1*0.166	1*0.166	0*0.166
합 계	0.666	0.5	0.333

[표 1]에서 보듯이 가중치를 적용하지 않았을 때 모니터1과 모니터2의 속성값의 합은 같았다. 하지만 선호 속성값에 따른 가중치를 부여한 결과 모니터1이 가장 높은 수치를 보였다.

4. 온톨로지

본 논문에서 지능공간의 다양한 Situation을 표현하기 위해 제안한 온톨로지 중 자원 온톨로지를 설명한다. 자원 공유를 위해 DiscoveredResource, LocalResource, VirtualResource, DeviceDriver, 그리고 Device의 서브 클래스들을 두어 보다 풍부한 자원의 정보들을 표현한다. 이렇게 한번 분류된 디바이스들은 각각의 특성에 맞게 세분화하여 분류하였다. 자원을 세분화하여 표현함으로써 자원 공유의 관점에서 볼 때 상황에 부합하는 자원을 찾고, 공유하기 위한 목적에 집약되는 온톨로지가 될 것이다.



[그림 6] 자원 온톨로지 클래스 다이어그램

5. 결론 및 향후과제

본 논문에서 제안하는 사용자 맞춤형 디바이스 협업 시스템을 이용하여 유비쿼터스 환경에서 모바일 단말기의 제한적인 자원 문제를 어느 정도 해결 할 수 있었다. 따라서 모바일 단말기는 사용자와 서비스간의 매개체 역

할을 수행함에 있어서 좀 더 기능적으로 확장되었다고 볼 수 있다.

향후에는 좀 더 다양한 상황을 고려한 디바이스 추론 알고리즘에 대해 알아보고 동적으로 변화하는 자원 정보 및 사용자의 선호도 정보 즉, 프로파일 정보를 실시간으로 반영하기 위한 프로파일 관리 기법에 대해 알아본다.

참 고 문 헌

[1] T.R. Halfhill, "Sun's Jini: Science, Not Magic", Microprocessor report, pp.10-13, March, 1999.
[2] G. Bhatti, Z. Sahinoglu, K. A. Peker, J. Guo, and F. Matsubara, "A TV-Centric Home Network to Provide a Unified Access to UPnP and PLC Domains," Proceedings of the 2002 IEEE 4th International Workshop on Networked Applications, pp.234-242, Jan., 2002.
[3] Rahul Gupta, Sumeet Talwar, Dharma P. Agrawal, "Jini Home Networking: A Step toward Pervasive Computing," IEEE Computer, Vol.35, No.B, pp.34-40, Aug., 2002.
[4] J. Waldo, "Alive and Well: Jini Technology Today,"IEEE Computer, Vol.33, No.6, pp.107-109, June, 2000.