OSGi 지능형 홈 네트워크 서비스를 위한 효율적인 자원 관리 방안

[1] 리건, [2] 최영락, [1,2] 홍원기
[1] 포항공과대학교 컴퓨터공학과, [2] 포항공과대학교 정보전자융합공학부

{gunine, dkby, jwkhong}@postech.ac.kr

An Efficient Method for Managing Resources of OSGi based Intelligent Home Network Services

[1]Jian Li, [2]Yeongrak Choi and [1,2]James Won-Ki Hong
[1]Dept. of Computer Science and Engineering, POSTECH
[2]Division of IT Convergence Engineering, POSTECH

요 약

첨단 기술과 다양한 서비스들의 융합이 가속화되면서 지능형 홈 네트워크를 구성하는 장치 및 이를 활용한 서비스의 종류 및 개수 또한 증가하고 있다. 비교적 매우 한정된 자원량을 가진 지능형 홈 네트워크 장치들 상에서 동작하는 서비스들을 최대한 수용하고 관리하기 위한 방안으로 서비스를 동적으로 활성화 및 비활성화 가능한 OSGi 개방형 프레임워크가 제시되고 있다. 본 연구에서는 OSGi 개방형 프레임워크에 서비스 관리자를 설계하고 구현하는 방식을 소개하고, 서비스 관리자에 적용 가능한 자원 유틸리티 함수를 설계하여 임의의 자원 최대량을 초과하지 않으면서 보다 효율적으로 서비스를 활성화 및 비활성화하는 방안을 제시하고자 한다.

Ⅰ. 서 론

지능형 홈 네트워크를 구성하는 가전기기, 센서 등 대부분의 장치들은 제한된 양의 컴퓨팅 파워, 메모리 공간, 스토리지 공간, 배터리를 가지고 있다 [1]. 현재, 지능형 홈 네트워크 기술의 발전과 더불어 해당 네트워크를 구성하는 장치들 및 제공되는 서비스의 수가 급증하는데 반해, 해당 서비스들을 한정된 자원에 모두 적재하기에 많은 어려움을 겪고 있다. 따라서 지능형 홈 네트워크 상에 존재하는 장치들을 효율적으로 관리하기 위해 여러 프레임워크들이 제안되고 있다.

OSGi(Open Service Gateway initiative) 개방형 프레임워크는 자바의 플랫폼 독립성과 실행코드의 네트워크 이동성을 고려하여 제정된 표준이다 [2]. 따라서 OSGi 프레임워크는 소용량의 메모리 장치를 위한 동적으로 서비스 활성화 및 비활성화를 지원한다. 동적인 서비스 활성화 및 비활성화를 위해 제안되는 방식으로는 ERA (Eager Resource Allocation)가 있다 [3]. ERA 는 시스템 초기화 과정에서 장치 내 자원을 최대한 소모하도록 가능한 많은 서비스들을 할당하는 방식을 사용한다. 그러나 이 방식은 시스템을 초기화 하는데 많은 시간을 필요로 하며, 서비스 수가 증가함에 따라 자원의 불필요한 소모 또한 증가하는 단점이 있다.

본 연구에서는 자원 유틸리티 함수 기반의 지연된 로딩(lazy-loading) 방식으로 OSGi 서비스 관리자를 확장 및 구현하여 가능한 적은 숫자의 서비스들을 메모리상에 로드하는 방식을 제안한다. 자원 유틸리티 함수는 메모리 사용량과 지역성(Locality)를 고려하여 시스템의 성능을 향상시키며, 메모리를 효율적으로 사용하는 데 초점을 두었다.

본 논문의 구성은 다음과 같다. 2 장에서는 구현할 서비스 관리자를 위해 확장된 OSGi 전반적인 시스템 구조와 함께 서비스 관리자에 적용할 자원 유틸리티 함수를 살펴본다. 3 장에서는 결론을 제시하고 향후 연구에 대해 기술한다.

Ⅱ. 본 론

이번 장에서는 구현되는 서비스 관리자에 따른 OSGi 프레임워크 구조를 설명하고, 본 연구에서 제안한 자원 유틸리티 함수에 대해 알아보고자 한다.

1. 확장된 OSGi 프레임워크 구조

OSGi 서비스 관리자는 서비스 관리자 내에 실행되는 서비스 및 서비스에 대한 번들, 그리고 번들 컨텍스트로 구성된다 [4]. 하나의 어플리케이션은 여러 개의 서비스와의 상호작용을 통해 실행되며, 런타임 시에 필요한 서비스를 추가 요청 가능하다.

지연된 로딩을 적용하기 위해 OSGi 상의 번들을 크게 서비스를 Export 하는 인프라스트럭처 번들, Export 된 서비스를 호출하여 사용하는 어플리케이션 번들, Export 된 서비스들을 관리하는 서비스 관리자로 구분하였다. 인프라스트럭처 번들은 서비스를 등록하는 역할을 담당한다. 서비스 관리자는 등록된 번들을 관리하는 역할을 담당하며, Export 된 서비스들의 활성화 여부를 모니터링하는 RSM (Resource & Service Monitor)과 자원 유틸리티 함수를 기반으로 활성화된 서비스 중 제거할 서비스를 결정하는 ISR (Intelligent Service Remover)로 구분된다. 그림 1 은 각 종류의

본 연구는 한국과학재단을 통해 교육과학기술부의 WCU 사업(WCU, World Class University)으로부터 지원받아 수행되었습니다(과제번호: R31-2008-000-10100-0)

번들과 서비스 사이의 관계를 도식화한 시스템 구조를 나타낸다.

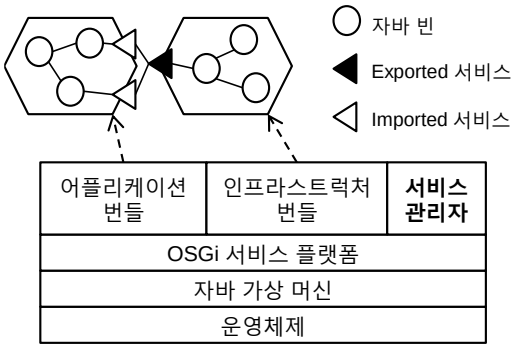

그림 1 시스템 구조

동작과정은 다음과 같다. 서비스 관리자는 RSM 을 통하여 모든 서비스 이벤트들을 모니터링한다. 이 때, 활성화 되려는 서비스가 메모리에 로드되는 경우, 메모리에 로드된 전체 서비스들이 차지하는 메모리 량이 우리가 미리 설정한 threshold 값 보다 크면 ISR 을 통하여 제거할 서비스들을 선정하여 메모리에서 제거한다. ISR 은 다음 4 가지 단계를 거쳐 제거할 서비스를 결정한다. 1) 새로운 서비스를 메모리에 로드하기 위해 필요한 메모리 량을 계산한다. 2) 자원 해제가 필요한 메모리 량을 기반으로 제거할 서비스 후보 목록을 만든다. 3) 각 목록에 대해 자원 유틸리티 함수로 유틸리티 값을 계산한다. 4) 계산된 유틸리티 값을 비교하여 제거할 서비스를 결정한다.

2. 자원 유틸리티 함수

제안하는 자원 유틸리티 함수는 각 서비스들이 소비하는 메모리 사용량 및 지역성(Locality)을 고려하였다. 각 서비스들이 소모하는 메모리 사용량이 많을수록 소비하는 에너지가 많아져 한정된 자원을 가진 장치에 오래 적재시킬 수 없다. 또한, 각 서비스의 지역성 값에 따라 서비스들이 빈번하게 교체되는지의 여부를 파악할 수 있으며, 빈번한 교체가 발생하는 서비스들은 시스템 전체 성능에 큰 영향을 미친다. 따라서 해당 두 요소를 유틸리티 함수에 고려하여, 이를 기반으로 총 자원 유틸리티 함수를 설계하였다 [5].

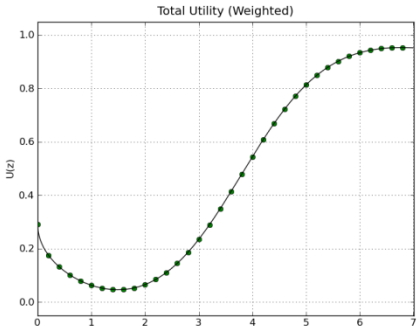

그림 2 총 자원 유틸리티 함수

총 자원 유틸리티 함수는 지역성 유틸리티 함수 U_A 와 메모리 사용량에 따른 유틸리티 함수 U_B 를 기반으로 계산된다. 각 유틸리티 함수를 모두 고려하기 위해 정규화 함수(N_A, N_B, & N_C)가 사용되며, w_A 와 w_B 를 통해 각 유틸리티의 가중치를 고려하여 총 자원 유틸리티 값을 계산한다. 따라서 계산된 총 자원 유틸리티를 통해 메모리를 적게 사용하는 동시에 지역성을 높이는 최적의 값을 찾을 수 있다. 그림 2 는 총 자원 유틸리티 함수 식을 적용한 그래프를 나타낸다.

$$U_{Total} = N_C(w_A(N_A(U_A)) - w_B(N_B(U_B)))$$

그림 3 은 임의의 시나리오를 기반으로 총 자원 유틸리티 함수를 적용한 구현과 교체할 서비스를 선택할 때 적용 가능한 Random 및 Greedy 알고리즘과 비교한 메모리 사용량을 나타낸다. 총 자원 유틸리티 함수를 적용했을 때, 다른 알고리즘보다 많은 메모리 량을 지속적으로 사용하도록 메모리 사용 효율을 증가시켜 OSGi 서비스 활성화 및 비활성화의 성능 및 효율을 향상시킬 수 있음을 확인하였다.

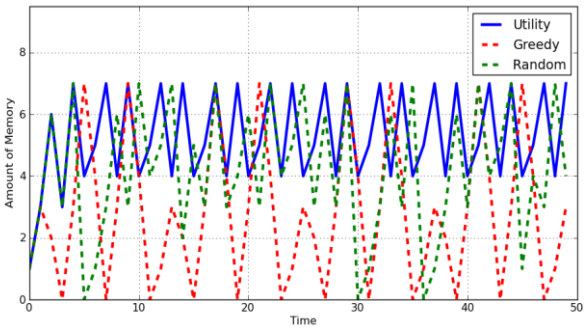

그림 3 유틸리티 함수와 기타 알고리즘을 적용 하였을 때 메모리 사용 효율 비교

Ⅲ. 결론

본 연구에서 지능형 홈 네트워크에 존재하고 있는 장치들의 한정된 자원을 효율적으로 사용하기 위하여 해당 자원들을 사용하고 있는 서비스들의 활성화 및 비활성화 과정을 자동화 시켜주는 서비스 관리자를 소개하고, 서비스 관리자에 적용 가능한 자원 유틸리티 함수를 제안하였다.

향후 연구로는 제안한 자원 유틸리티 함수에서 몇몇 성능에 영향을 줄 수 있는 계수들에 대한 튜닝을 통하여 좀 더 나은 성능을 보여주는 유틸리티 함수를 만들고 서비스 관리자에 적용하여 제안한 함수가 좀 더 나은 성능을 발휘함을 보여줄 것이다.

참 고 문 헌

[1] Moller, M. Akerholm, J. Fredriksson, and M. Nolin, "Evaluation of component technologies with respect to industrial requirements, ", Proc. of EUROMICRO’04, pp. 56-63, 2004.

[2] Open Service Gateway initiative Alliance, (http://www.osgi.org/).

[3] Lazy Activation Policy, OSGi Service Platform Core Specification, Release 4, section 4.

[4] R. S. Hall and H. Cervantes, "Challenges in building service-oriented applications for OSGi, ", IEEE Communications Magazine, vol. 42, no. 5, pp. 144-149, May 2004.

[5] Annie P Foong, Yu-Hen Hu, Dennis M Heisey, "Web caching: Locality of References Revisited", ICON’00, 2000.