유비쿼터스 환경에서 개인정보보호를 위한 오토노믹 컴퓨팅 아키텍쳐

임정은*, 이택, 인호, 백두권
*고려대학교 정보통신대학 컴퓨터학과
e-mail:{jelim, comtaek0, hoh_in, baik}@korea.ac.kr

Autonomic Computing Architecture for Privacy Protection in Ubiquitous Environment

Jung-Eun Lim, Taek Lee, Hoh Peter In, Doo-Kwon Baik
Department of Computer Science and Engineering
College of Information and Communications, Korea University

요 약

기존의 컴퓨팅환경에서 유비쿼터스 컴퓨팅환경으로의 전환이 이루어지면서 사용자는 언제 어디서나 네트워크에 연결될 수 있다. 이는 역으로 말하면 언제 어디서나 사용자의 정보가 네트워크에 유출될 수도 있다는 것을 의미한다. 컴퓨팅 환경에서 개인정보의 보호와 편리성의 추구는 서로 상반되는 문제이다. 이 논문에서는 오토노믹 컴퓨팅(Autonomic Computing)의 개념을 이용, 정해진 개인정보 보호정책(Privacy Policy)에 기반 하여 개인정보를 보호하고 서비스 사용자의 환경 변화에 대한 상황인지(Situation-Aware)를 통해 유연한 개인정보 보호정책을 적용 할 수 있는 아키텍쳐를 제안함으로서 유비쿼터스 환경이 주는 편리성과 개인 정보 보호를 극대화 할 수 있을 것으로 기대된다.

1. 서론

기존의 컴퓨팅환경에서 유비쿼터스 컴퓨팅환경으로의 전환이 이루어지면서 사용자는 언제 어디서나 네트워크에 연결될 수 있다. 이는 역으로 말하면 언제 어디서나 사용자의 정보가 네트워크에 유출될 수도 있다는 것을 의미한다. 컴퓨팅 환경에서 보안성과 편리성의 추구는 서로 상반되는 Trade-off 문제이다. 이 논문에서는 오토노믹 컴퓨팅(Autonomic Computing)의 개념[1,2]을 이용, 정해진 개인정보 보호정책(Privacy Policy)에 기반 하여 개인정보를 보호하고 서비스 사용자의 환경 변화에 대한 상황인지(Situation-Aware)를 통해 유연한 개인정보 보호정책을 적용 할 수 있는 아키텍쳐를 제안함으로서 보안성(개인정보보호)과 편리성(사용자 개입의 최소화)의 문제를 다루어보았다.

2. 오토노믹 컴퓨팅

다양한 아키텍처의 컴퓨터 시스템들이 상호간에 연결됨에 따라 시스템 설계자들은 대형화되고 복잡해져가는 시스템 컴포넌트들 간의 상호작용을 디자인하고 예상하기가 더욱 힘들어져 가고 있다. 따라서 시스템의 인스톨, 설정, 최적화, 유지관리, 통합 등과 관련된 일련의 작업들은 전문가들에게도 여간 어려운 일이 아니다. 이에 대한 대안으로 IBM에서는 오토노믹 컴퓨팅이라는 새로운 비전을 제시하고 있다.

오토노믹 컴퓨팅이라 함은 시스템의 모든 관리를 사람의 개입을 최소화하고 자동화하여 시스템 컴포넌트들 간에 자체적으로 해결하도록 하려는 새로운 하나의 패러다임이라 할 수 있다.[1]

오토노믹 컴퓨팅의 기본단위를 일반적으로 “오토노믹 엘리먼트”라고 한다. 오토노믹 엘리먼트는 혼자서 작동할 수 있으며, 여러 오토노믹 엘리먼트들이 상호협상을 통한 상호작용을 할 수 있다.

오토노믹 컴퓨팅 시스템에서의 Self-Management는 Self-Configuration, Self-Healing, Self-Optimizing,

Self-Protecting의 세부항목들이 있으며[1] 이런 항목들은 오토노믹 시스템이 추구하는 궁극적인 목표들이 된다. 기능적인 면에서 보자면 시스템은 스스로 모니터링을 통해 환경인식과 자기인식을 이루어내고 이를 바탕으로 자기 조정을 하게 된다.[2] 시스템은 스스로 조정할 수 있는 기능을 이용하여 현재 시스템이 처해 있는 상황에 맞도록 자신을 최적화하게 된다.

개인 프라이버시 문제들에 대한 접근 또한 오토노믹 컴퓨팅 환경에서의 자기조정 기능을 이용한다면 보다 손쉬워 지게 된다. 시스템 상황을 시시각각 모니터링하고 그에 맞도록 개인정책을 조정함으로써 유비쿼터스 환경에서 프라이버시 문제를 보다 효율적이고 융통성 있게 강화시킬 수 있다.

3. 개인정보보호를 위한 오토노믹 엘리먼트 아키텍쳐

본 논문에서는 개인정보보호를 위한 사용자 에이전트 기능 구현을 오토노믹 컴퓨팅 환경의 오토노믹 엘리먼트 구조[2,3] 내에 접목시키는 방안을 검토해 보았다. [그림 1] 오토노믹 엘리먼트 안의 각각의 에이전트들을 설명하면 다음과 같다.

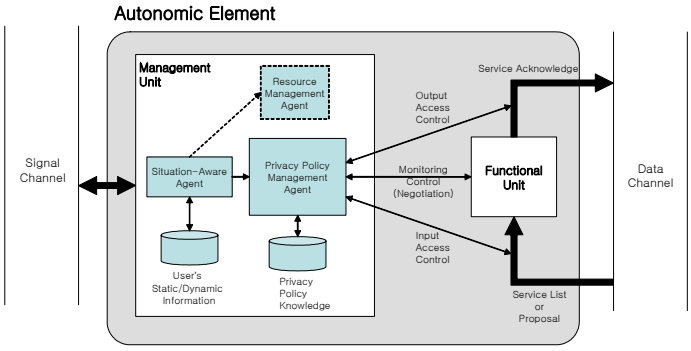

그림 1. 개인정보보호를 위한 오토노믹 엘리먼트 아키텍쳐

3.1 Privacy Policy Management Agent

사용자가 서비스를 제공받을 때 노출될 수 있는 프라이버시 정보에 대해서 미리 Knowledge에 수립된 개인정보 보호 정책에 따라서 개인정보를 노출 혹은 차단하게 한다. 또한 사용자의 현재 상황에 따라 제공받는 서비스 형태가 달라지므로 Situation-Aware Agent에서 추론된 결과를 이용하여 유연하게 개인정보 보호 정책을 수립하도록 한다.

Input/Output Access Control 라인에서는 Functional Unit에 출입하는 개인정보에 대한 접근 권한 여부를 사용자 정보 보호 정책에 기반 하여 검사하고 Monitoring & Control (Negotiation) 라인에서는 사용자 정보 보호 정책과 서비스 제공자의 정보사용 정책을 비교하여 서로 상충하는 항목이 있는지 검토하며 최종적으로 서비스에 대한 승낙/거부 의사를 결정하게 된다.

3.2 Situation-Aware Agent

사용자의 개인정보에는 정적인 저장 정보(예:이름, 성별, 주소)와 동적인 Context정보(예:현재위치, 체온)가 포함된다. Situation-Aware Agent는 사용자의 문맥(Context) 정보를 인지하고 사용자의 환경에 따라 변하는 위치, 시간과 같은 동적인 정보를 실시간으로 추적하여 저장관리(User's Dynamic Information)한다. 이는 또한 사용자의 현재 상황 정보를 추론하되 사용자의 개입을 최소화 하면서 개인정보 보호 정책을 오토노믹하게 결정하도록 하는데 도움을 준다. 오토노믹 엘리먼트의 특징 중에 Self-Management 기능과 연관되는 부분이라 할 수 있겠다.

3.3 Resource Management Agent

현재 사용자가 처해 있는 상황을 Situation-Aware Agent의 도움을 받아 모니터링하고 사용자가 가장 손쉽게 접근할 수 있는 장치 자원들을 관리하는 역할을 한다. 제안서가 통과되고 서비스가 개시되면 특정 디바이스를 선택하여 실제적으로 서비스 제공자와 사용자간의 서비스 연결을 성사시키는 역할을 한다.

4. 시나리오(홈서버를 통한 서비스 주문)

실제적으로 어떤 과정을 통해 서비스가 제공되고 그 과정 속에서 어떻게 개인정보가 보호되는지를 간단한 예를 통해 설명하도록 하겠다. 그림 2는 서비스 요청이 발생하고 서비스가 최종적으로 제공되기까지를 간단하게 설명한 상황도이다. 각각의 단계별로 설명을 하도록 하겠다.

단계1) 사용자는 원하는 서비스를 Home Server(이하 HS)에게 의뢰하게 된다. (여기서는 음식 주문을 예로 들겠다.) 단계2) HS는 현재 등록되어 있는(신용도가 보장되는) 음식점(Service Provider:이하 SP)에게 음식 서비스에 대한 제안서를 제출할 것을 요청한다. 단계3) 음식점(SP)은 HS에게 음식관련 제안서를 제출한다. 제안서의 내용은 손님에 맞는 맞

춤형 메뉴를 제공하기 위한 손님 기호(개인정보)와 기타 사항들*(가격, 배달시간, 결제방법 등)을 묻는 내용으로 이루어져 있다.

다음은 XML형태로 표현한 제안서의 간단한 예이다.

```
<proposal>
 <SP-specification>
  <sp-id>FastFood001</sp-id>
  <service-name>hamburger</service-name>
  <contact-info>Tel)123-1234</contact-info>
  …
 </SP-specification>
 <read-info>
  <user-info>
   <needed-info>age</needed-info>
   <needed-info>sex</needed-info>
   <needed-info>favorite food style</needed-info>
  </user-info>
  …
 </read-info>
 <write-info>
  <price>2500</price>
  <delivery-time>08:00pm</delivery-time>
  <pay>cash</pay>
   …
 </write-info>
 …
</proposal>
```

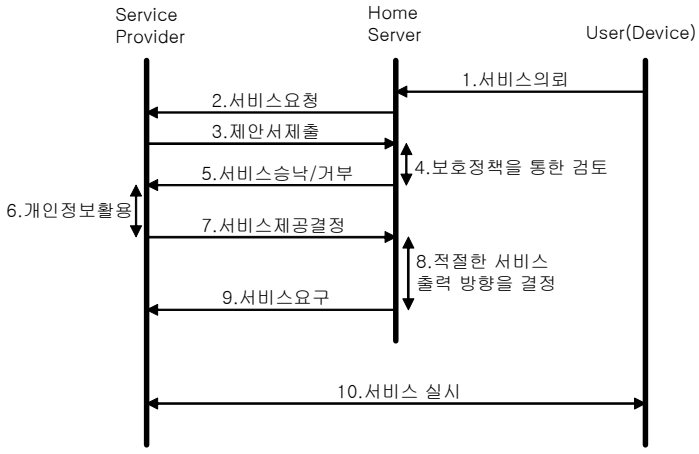

그림 2. 서비스 제공을 위한 각 개체들의 상호작용

단계4) 사용자 보호 정책(SP의 인증/승인/신용도 검사, 개인정보 입출력 접근 권한 검사 등의 내용으로 구성)에 의거 현재 제안서 내용을 검토하고 서비스를 승낙할 것인지 거부할 것인지를 결정한다. 만약 해당 서비스를 거부할 경우 단계2로 돌아가 새로운 음식점(SP)을 찾게 된다. (물론 이 예에서는 개인정보 보호정책 뿐만 아니라 상술한 기타 사항들*에 대한 선별을 위해 기타 부가적인 정책 또한 필요할 것이다.) 다음은 개인정보 보호정책의 간단한 예이다.

```
<protection-policy>
 <accept-condition>
  <trusted-SP-ID>
   FastFood001, FastFood002
   …
  </trusted-SP-ID>
  <permitted-info>
   name, sex, age, hobby, favorite-food-style
   …
  </permitted-info>
  <acceptable-SP-trust-level value="7"/>
  …
 </accept-condition>
</protection-policy>
```

단계5) 제안서의 내용과 사용자 보호 정책의 항목들을 비교하여 최종적으로 서비스의 승낙/거부 의사를 음식점에게 밝힌다. 단계6) 음식점(SP)은 수집된 정보들(개인정보 및 기타사항들)을 바탕으로 손님에게 가장 적절한 음식 서비스를 결정한다. 단계7) 결정된 서비스 내용을 HS에게 통보한다. 단계8) 지금 사용자가 처해 있는 상황을 HS가 감지하고 가장 적철한 장치를 선택하여 서비스 제공을 알린다. 처음에 사용자가 핸드폰을 통해 음식을 시켰지만 현재는 핸드폰을 꺼놓고 TV를 시청하고 있다면 핸드폰으로 음식 서비스 결정 여부를 알려줄 수는 없다. 따라서 현재 상황을 인지하고 TV를 통해 음식 서비스 제공의 성사 여부를 알리게 된다. 단계9) 손님(User)의 최종 의사결정이 내려지고 서비스를 음식점(SP)에게 요구하게 된다. 단계10) 실제적인 음식 서비스가 손님에게 제공되게 된다. 이 단계 이후에는 서비스의 종료가 이루어질 때 까지 더 이상 HS의 개입은 일어나지 않게 된다.

상기 기술한 시나리오의 경우 예외상황이 발생할 수도 있다. 예를 들어 손님(User)이 음식을 시킨 사실을 깜빡 잊고 집밖으로 나갈 수도 있고(User의 서비스 불가 지역으로의 이동) 갑작스럽게 약속에 없던 손님들이 찾아와 주문 내용을 변경해야 하는 경우(새로운 리소스의 추가, 사용자 정책의 변화)가 생기게 된다. 그밖에 여러 예외상황이 발생할 수 있으며 대부분 2장에서 설명한 에이전트들에 의해 커버되어야하는 경우가 가장 이상적인 상황이라 하겠다. 그렇지 못한 경우는 현재 단계를 취소하고 단계2부터 다시 밟아내려 가야 할 것이다. 2장에서 설명했던 에이전트들의 단계별 개입 여부를 보면 Policy Management Agent는 단계 3~5를, Situation-Aware

Agent는 단계 4~8을, Resource Management Agent는 단계 8~10을 각각 커버하게 된다.

5. 결 론

본 논문에서는 오토노믹 컴퓨팅에서 오토노믹 엘리먼트가 갖는 Self-Management 특징을 활용하여 유비쿼터스 환경의 컴퓨팅 디바이스들 간의 정보 유통 과정에서 노출될 수 있는 개인정보의 취약성을 해결하는 방안을 모색해 보았다. 개인정보 보호에 대한 의사 결정 과정중 사용자의 개입을 최소화 할 수 있다는 취지에서 오토노믹 컴퓨팅 환경이 제공해 주는 Self-Management 개념의 활용은 훌륭한 선택이 될 수 있다.

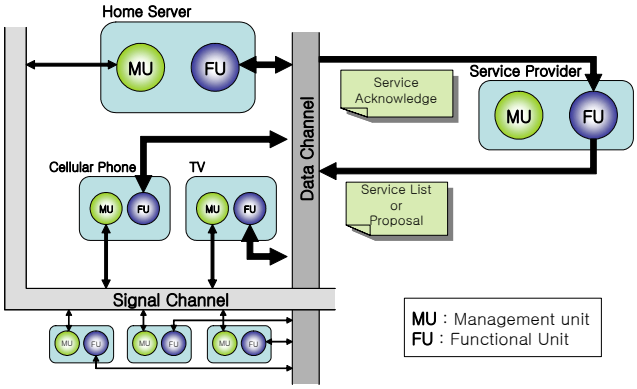

그림 3. 다른 오토노믹 엘리먼트들 간의 상호작용

향후 더욱 연구되어야하는 Management Unit 내의 주요 기능들은 다음과 같다.

- 현재 의존관계가 있는 다른 오토노믹 엘리먼트들에 대한 표현, 다른 엘리먼트들을 얼마나 신용할 수 있는가의 신용정도 표현
- FU에서 현재 실행중인 S/W의 상태(보안레벨)에 대한 표현, 그에 따른 적용 가능한 보안정책 선택
- 연관된 오토노믹 엘리먼트들 간에, 또는 관리자와의 통신 방법 제공
- 보호되어야 하는 개인 정보에 대한 접근권한 정책
- 정보 요청 엘리먼트들에 대한 빠른 인증 처리 방법 제공
- 서비스 제공자의 개인정보 이용 정책과 사용자의 개인정보 보호정책이 협상단계에서 서로 상충 될 때의 해결방안

현재 유비쿼터스 환경에서 개인정보 보호 기술 개발을 위한 다양한 방법들이 제시되고 있다. 이러한 차원에서 본 논문은 오토노믹 컴퓨팅 기반 구조의 활용을 생각해 보았다. 앞으로의 연구 과제는 Self-Management(여기서는 Situation-Aware Agent)에 대한 기능을 얼마나 구체화 시키고 조직화 시켜 다른 여러 엘리먼트들 간의 자율적인 연동을 이루어 내느냐 일 것이다. [그림 3]

여러 오토노믹 엘리먼트 내의 Management Unit들 간의 연동을 통해 분산된 정보 보호 정책들을 활용하여 계층적으로 구조화된 상위 레벨의 정보 보호 정책을 수립할 수도 있으며 각각의 오토노믹 엘리먼트들이 제공하는 리소스들의 관리와 활용을 통하여 시스템 성능을 극대화 할 수 있을 것으로 기대된다.

참고문헌

[1] J. O. Kephart and D. M. Chess, “The Vision of Autonomic Computing”, IEEE Computer, 36(1), pp. 41-50, 2003

[2] Roy Sterritt, Dave Bustard, University of Ulster, “Towards an Autonomic Computing Environment”, 14th International Workshop on Database and Expert Systems Applications (DEXA′03)September 01 - 05, 2003

[3] David M. Chess, Charles C. Palmer, Steve R. White, “Security in an autonomic computing environment”, IBM System Journal, March, 2003

[4] Philip Robinson and Michael Beigl, "Trust Context Spaces: An Infrastructure for Pervasive Security in Context-Aware Environments", First International Conference on Security in Pervasive Computing, Boppard, Germany, March 12 - 14, 2003