

모델 기반의 자가 적응형 소프트웨어 설계

이상희⁰, 이은석
 성균관대학교 컴퓨터공학과
 {neomine⁰, eslee}@selab.skku.ac.kr

Design Of Model based Self-Adaptive System

Sanghee Lee⁰, Eunseok Lee
 Dept. of Computer Engineering, Sungkyunkwan University

요 약

최근 컴퓨팅 환경의 급격한 발전과 다양화는 소프트웨어 외부 환경의 복잡성과 변화를 야기시켰으며, 소프트웨어는 예상하지 못한 외부 환경 변화에 직면하였을 때, 지속적으로 서비스의 제공과 동시에 동적 변경에 대한 적응 능력을 필요로 하게 되었다. 본 논문에서는 외부 환경의 변화를 인식하고, 대안을 선택하기 위해 '어떻게' 적응하도록 할 것인가에 대해 모델을 이용한 적응 매커니즘을 제안하며, 시스템이 고려해야 할 컨텍스트 정보들을 모델링을 통하여 소프트웨어가 행위를 변경할 때 이용할 수 있도록 모델링 기법과 모델 기반 자가 적응형 프로세스를 제안한다. 또한, 프로세스를 이용하여 동작하는 모델 기반 자가 적응 시스템의 일반적 모듈들을 제시하고, 그 유효성을 확인하기 위하여 원격 화상 회의 시스템에 적용하였다.

1. 서 론

유비쿼터스 컴퓨팅 환경의 도래와 정보통신의 급격한 발전에 따른 다양한 디바이스와 플랫폼, 네트워크 환경이 등장하였고, 사용자의 요구 사항 또한 다양하게 요구되고 있다. 시스템은 이러한 동적인 환경의 변화를 반영하고, 스스로 적응 할 수 있는 자가 적응성이 요구 되고 있으며, 따라서 소프트웨어는 실행 중인 시스템의 현재 상태를 정확한 표현과 시스템의 변경에 대한 모델링을 수행하고 적응하려는 시도가 이루어지고 있다.

본 논문에서는 동적인 시스템의 환경과 상태를 모델로 제시하고, 모델을 기반으로 하여 시스템 스스로가 환경에 적응할 수 있는 모델 기반 자가 적응형 시스템을 제안한다[1]. 또한, 모델 기반 자가 적응형 시스템의 유효성을 확인하기 위하여, 여러 사용자의 다양한 실행 환경을 반영할 수 있는 원격화상회의 시스템[2]에 제안 시스템을 적용하여 설계하고 구현하였다. 시스템을 구현하기 위하여 우리는 환경 정보를 표현할 수 있는 컨텍스트 모델링 방법과 모델을 이용한 자가 적응 프로세스를 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 살펴보고, 3장에서는 모델을 기반으로 한 시스템에서 고려해야 할 컨텍스트의 정의와 모델링 방법, 프로파일을 제안한다. 4장은 모델 기반의 적응화를 위한 모듈을 제안하며, 5장에서는 프로토타입을 구현과 시스템 평가를 하였으며, 6장에서는 결론을 기술하였다.

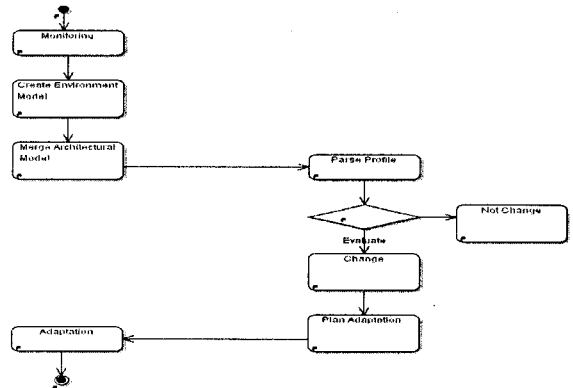
2. 관련 연구

2.1. Architecture based on Self-Adaptive Software

구조 기반의 자가 적응형 소프트웨어는 사용자의 실행 환경과 변화에 맞추어 동적인 재배치와 재설정이 가능하게 하기 위해 구조(Architecture)를 이용하여 변경하는 소프트웨어 이며, 실행 시에 스스로 적응하게 하기 위한 자가 적

응형 소프트웨어에 대한 접근 방법 중에 하나이다. 상호 작용하는 컴포넌트들을 그래프 형태로 표현하는 소프트웨어 구조를 변경함으로써 환경 적응은 효율적이지만 사용자의 모든 주변 자원을 인식하고, 반영하지 못하는 단점이 있다 [3].

3. 모델 기반 자가 적응 시스템의 동작 프로세스



(그림 1) 프로세스를 기술한 활동 다이어그램

본 장에서는 모델 기반 자가 적응 시스템의 프로세스를 살펴 본 후, 모델링 하기 위해 고려해야 할 컨텍스트들에 대한 정의와 모델링 방안을 제시한다. (그림 1)는 모델 기반 적응 시스템의 전체적인 흐름을 표현한 활동 다이어그램이다. 프로세스의 전체 사항으로, 초기 시스템 요구 사항 분석 시 도메인 전문가로부터 시스템 아키텍처 모델은 미리 정의되어져 있어야 한다. 프로세스 흐름은 다음과 같다.

Step 1. 사용자(Client) 측은 환경 컨텍스트들의 모니터링을 통

하여 자원 정보를 수집한다.

Step 2. 수집된 정보를 통하여 메타 모델을 이용하여 컨텍스트 모델을 생성한다.

Step 3. 미리 정의된 아키텍처 모델과 병합하여 새로운 상황 모델을 정의하고, 서버측에 모델을 기반으로 하여 작성한 CC/PP 프로파일을 전송한다.

Step 4. 서버 측은 정의된 상황 모델의 분석을 통하여 사용자 환경을 분석한다.

Step 5. 변경의 필요 여부를 파악한 후, 변경이 필요하다고 판단될 경우, 적응 정책을 계획한다.

Step 6. 계획된 적응 정책은 사용자 측에 전송된다.

Step 7. 적응 모듈(Adaptation Module)을 통하여 사용자 환경에 적합하게 적응한다.

3.1. 컨텍스트

모델링을 위해 고려해야 하는 컨텍스트들을 정의하였다. 컨텍스트는 정적, 동적 컨텍스트 두가지로 카테고리로 분류된다. 정적 컨텍스트는 사용자의 운영체제 및 사용 소프트웨어 및 플레이어 등 환경이 적은 요소 등을 고려할 수 있다. 동적 컨텍스트에는 현재 CPU 및 메모리의 사용량, 카메라 및 오디오 디바이스 상태, 네트워크 대역폭, 사용자의 컨텐츠 선호도 등으로 지속적으로 동적 변경을 수행하는 요소들이다. 이러한 환경 자원들을 수집하여 모델링을 통하여 시스템의 적응에 효율적으로 사용을 제안하는 것이 본 논문의 목적이다.

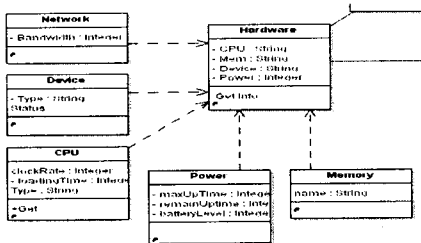
3.2. 모델링

모델을 기반으로 적응할 수 있도록 모델링 기술을 제시한다. 시스템을 구성하고 상호 작용하는 컴포넌트들을 그래프 형태로 표현해 높은 소프트웨어 아키텍처의 모델과 사용자의 동적 환경을 모니터링한 정보를 포함하는 컨텍스트 모델을 이용한다.

3.2.1. 아키텍처 모델링

아키텍처 모델링은 시스템의 요구 분석을 통하여 도출되는 시스템의 전체적인 구조에 대한 아키텍처를 기술한 것이다.

3.2.2. 컨텍스트 모델링



(그림 2) 컨텍스트 메타 모델

컨텍스트를 표현하기 위해 이용하는 메타 모델은 사용자, 사용자의 하드웨어 환경, 소프트웨어 환경 크게 3가지 카테고리로 분류되며, 3가지 엘리먼트는 루트 엘리먼트로 표현되며, 하위의 여러 개의 서브 엘리먼트들이 존재한다. (그림 2)는 사용자의 환경에 영향을 주는 요소들을 구분해 내고, 이들간의 관계를 객체 지향 방법을 적용하여 일부

표현한 것이다. 수집된 컨텍스트 정보는 메타 모델의 애트리뷰트로 기술된다.

정의된 컨텍스트 정보들을 표현하기 위하여 '(S,V,O)'의 Triple 구조를 사용하여 RDF(Resource Description Framework) 구조로 표현한다. S는 컨텍스트의 주체가 되는 대상에 해당하며, V는 컨텍스트 타입, O는 실제 엔티티의 값에 해당한다. 예를 들어, (운영체제, 타입, Linux)로 표현한다.

위에서 표현된 컨텍스트 모델은 소프트웨어가 인식할 수 있는 기술하는 방법으로 변환되어야 하며, 이를 위해 본 논문에서는 CC/PP 기술을 적용하였다.

하나의 메타 모델 클래스와 애트리뷰트로 표현된 사용자 환경 자원은 CC/PP 프로파일에서 하나의 엘리먼트로 표현되며, 다음 장에서 소개되는 컴포넌트로 CC/PP Processor는 이러한 변환 동작을 해주는 컴포넌트가 존재한다. (그림 3)는 CC/PP Processor를 통하여 생성된 사용자 환경 정보를 기술한 있는 프로파일의 일부이다[4].

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prf="http://www.testProfile.com/profiles/ccppschemata-20030226#">
  <rdf:Description rdf:ID="MyProfile">
    <prf:component>
      <rdf:Description rdf:ID="HardwarePlatform">
        <prf:ScreenSize>84x30</prf:ScreenSize>
        <prf:Type>PDA</prf:Type>
        <prf:CPU>PPC</prf:CPU>
        <prf:Memory>16MB</prf:Memory>
      </rdf:Description>
    </prf:component>
  </rdf:Description>
</rdf:RDF>
```

(그림 3) CC/PP로 기술된 사용자 컨텍스트 프로파일

4. 모델 기반 적응화 모듈

모델을 시스템에 적용하기 위해 실행하기 위해 일반적으로 적용할 수 있는 모듈이다.

- 1) **Monitoring Module** : 지속적으로 사용자 디바이스와 환경정보를 수집한다. 수집할 대상은 사용자의 운영체제나 각종 디바이스 상태 정보와 같은 정적 정보와 네트워크나 메모리 사용량 등의 동적 정보가 포함된다.
- 2) **Model Manager** : 제시한 메타모델을 이용하여 수집된 정보를 표현하는 환경 모델의 생성과 적절한 컴포넌트 구성 정보 및 관계를 나타낸 미리 정의된 아키텍처 모델과 병합하여 상황 모델을 생성한다.
- 3) **CC/PP Processor** : 생성된 상황 모델을 소프트웨어가 인식할 수 있는 디스크립션을 생성한다. CC/PP를 사용하며, 컨텍스트 스키마를 통하여 수집된 정보 저장한 환경 모델 프로파일을 생성한다.
- 4) **CC/PP Interpreter** : 사용자 자원 및 환경 정보를 담은 프로파일 해석한다. 해석된 정보는 변경 여부를 결정하기 위해 Change Evaluator에 전송된다.

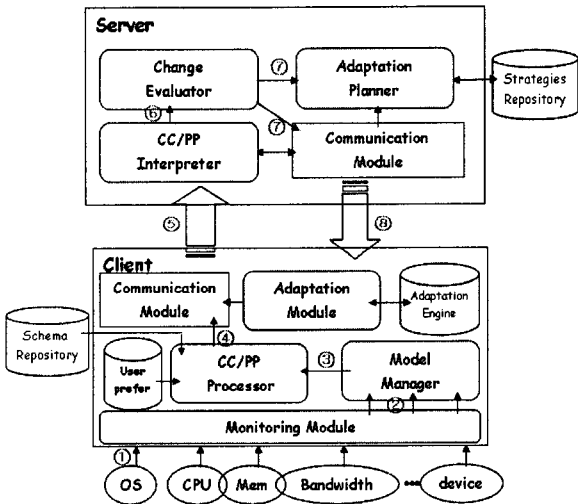
5) Change Evaluator : 접속한 사용자의 환경에 대하여 변경이 필요한지 아닌지를 분석하고, 변경이 필요하다고 판단될 경우, 적응 계획을 수립하기 위해 Adaptation Planner에 정보를 전송한다.

6) Adaptation Planner : 미리 정의된 적응 전략을 이용하여 전송된 사용자 환경 정보에 맞는 정책을 검색하고, 적응 계획을 수립한 후, Description을 작성한다.

7) Adaptation Module : 서버로부터 전송 받은 Description을 분석하고, 분석된 적응 정책을 실행하기 위해 Adaptation Engine에서 필요한 라이브러리들을 검색한다. 검색된 라이브러리와 컴포넌트 정보를 이용하여 적용한다.

8) Adaptation Engine : 적응에 필요한 실행 라이브러리 저장하는 저장소의 역할을 한다.

(그림 4)는 위의 각 모듈을 화상회의 시스템에 적용시킨 아키텍처이다.



(그림 4) 모델 기반의 원격화상회의 시스템 아키텍처.

5. 구현 및 평가

5.1. 구현

제안 시스템은 유효성을 확인하기 위해 원격화상회의 시스템에 적용하여 프로토타입을 구현하고, 테스트하였다. Java SDK를 기반으로 하여 구현하였으며, JMF(Java Media Frame)을 이용하여 화상을 전송하는 핵심코드를 제작하였다.

5.2. 평가

기존의 원격화상회의 시스템은 접속한 사용자의 환경 및 자원에 대하여 정책적으로 일부 정보들은 고려하나, 자원들의 정확한 반영과 표현을 제공하지 못하고 있다. 그러나, 시스템의 지속적인 변경과 함께 보다 많고 다양한 자원으로부터 얻어지는 정보들은 효과적으로 관리할 필요가 있으며, 본 논문에서는 그러한 환경 및 자원 정보들을 정확하게 표현하고 반영하기 위한 도구로서 모델링 기법을 이용하여, 이를 통해 기존 시스템에서 고려하지 않는 정보까지도 관리한다. (표 1)은 모델 기반 원격

화상회의 시스템에서 관리하는 컨텍스트들을 일부 나열한 표이다.

<표 1> 컨텍스트 정보 표현

고려한 컨텍스트	기존원격화상회의	모델기반화상회의
사용 인코더	O	O
포트	O	O
운영체제	O	O
디바이스 타입	X	O
스크린 사이즈	X	O
CPU 사용률	X	O
네트워크 밴드위스	X	O

위에서 제시한 외부 환경들에 대한 모델링을 통하여 시스템의 현재 상태를 정확하게 반영하고 최적의 적응 정책을 계획할 수 있다. 이를 통해, 본 시스템은 기존 원격화상회의 시스템보다 미디어 콘텐츠의 최적의 QoS를 보장하며, 데스크탑 PC와 PDA 등 이종의 환경에서 스스로에 맞는 다른 플레이어나 인코더, 렌더러 등을 적절하게 선택적 적응이 가능하고, 영상의 사이즈를 줄이거나, 원격화상회의에 필요한 영상, 음성, 텍스트 중 사용자의 선호도를 반영하여 개인 선호에 따른 서비스도 가능하다.

6. 결론

본 논문은 사용자의 정적, 동적 환경 변화에 스스로 적응하기 위하여 컨텍스트 개체를 정의하고, 이를 바탕으로 카테고리 형태의 구조로 표현된 메타 모델을 이용하여 표현하는 컨텍스트 모델링을 제안하였다. 제안된 아키텍처 모델과 환경 모델을 도입하여 확장된 모델을 생성하고, 이 모델을 이용하여 스스로 환경에 적응할 수 있는 프로세스 제시하고, 지속적으로 변경되는 환경을 어떻게 보다 효과적으로 표현하는지와, 어떻게 적응되는가를 보였다.

향후 연구는 시스템간 공통적으로 재사용할 수 있는, 지식 공유를 위한 온톨로지 컨텍스트 모델에 대한 연구가 있으며, 현재 진행 중에 있다.

참고 문헌

[1] Self adaptive software, DARPA, BAA 98-12, Proper Information Pamphlet, 1997.
 URL: http://www.darpa.mil/ito/Solicitations/PIP_9812.htm
 [2] Wenbiao Zhu, etc. al. 'JQOS: a QoS-based Internet videoconferencing system using the Java media framework (JMF)' ECE2001 Canadian Conference Pages: 625 - 630 vol. 1 May 2001
 [3] Peyman Oreizy, etc. al. "An Architecture-Based Approach to Self-Adaptive Software" IEEE Educational Activities Department, Volume 14, Pages: 54 - 62, May 1999.
 [4] W3C, Composite Capability / Preference Profiles (CC/PP), <http://www.w3.org/Mobile>, 2004.