

스마트 홈을 위한 상황기반 오류 복구 시스템 설계

유수진[○] 김성조

중앙대학교 컴퓨터공학부

Milkman93@konan.cse.cau.ac.kr, sjkim@cau.ac.kr

(A Design of A Context-Aware Fault Recovery System for Smart Homes)

Sujin Yu[○] Sungjo Kim

School of Computer Science & Engineering Chung-ang University

요 약

스마트 홈의 목적은 가정 혹은 사무실에서 사용자에게 보다 양질의 삶을 위한 서비스를 제공하는데 있다. 스마트 홈의 구성요소로는 여러 종류의 장치와 사용자 편의를 위해 제공되는 서비스가 있으며, 스마트 홈의 목적에 부합하여 서비스는 계속 증가할 것으로 전망된다. 스마트 홈에서 제공하는 서비스의 확대에 인하여 서비스가 이행되지 못하거나 장치와 서비스 간의 상충된 기능 수행으로 발생 되는 오류 또한 증가할 것으로 예상됨에 따라 보다 안정된 스마트 홈 서비스 제공을 위해 오류 복구 시스템 개발이 요구된다. 기존의 모델 기반의 스마트 홈 오류 복구 시스템은 스마트 홈에 새로운 장치가 동적으로 추가되거나 서비스가 변경 되는 경우와 같은 상황에서 발생하는 오류를 해결하지 못한다. 본 논문에서는 동적인 상황에서도 이러한 오류를 해결 하기 위하여 모델 기반이 아닌 상황인지 기반의 오류 복구 시스템을 제안한다. 스마트 홈은 가정에 오류가 발생했을 지라도 제안된 오류 복구 시스템을 이용하여 스마트 홈 상황에 적절한 서비스를 사용자에게 제공할 수 있다.

1. 서 론¹²

유비쿼터스 개념은 1988년 미국 제록스사의 Mark Weiser에 의해 컴퓨팅에 적용되어 유비쿼터스 컴퓨팅이라는 개념이 대두 되었다[1]. 특히 스마트 홈은 가정 내 정보가전기기가 유무선 네트워크로 연결되어 사용자에게 서비스를 제공함으로써 유비쿼터스 컴퓨팅을 실현한다. 스마트 홈은 지능형 정보가전기기와 서비스로 구성되며, 가정 혹은 사무실에서 사용자에게 보다 질 좋은 삶을 위한 서비스 제공을 목적으로 한다[2]. 그러나 스마트 홈에 대하여 간과하고 있는 사실은 스마트 홈이 어느 순간 완성되어 사용자에게 보여지는 것이 아니기 때문에 스마트 홈의 발전과 더불어 다양한 문제점이 도출될 수 있다[3]. 특히 사용자에게 제공되는 서비스는 스마트 홈의 발전에 발맞춰 다양해지고 양적으로 증대될 것이며, 이는 사용자를 위한 양질의 서비스로 발전함과 동시에 스마트 홈에서 오류를 발생하는 원인이 될 수 있다. 가정의 지능형 가전 기기들과 유무선의 네트워크

장비들 그리고 사용자를 위한 서비스가 스마트 홈의 네트워크를 이루면서 예기치 못한 오류를 발생시킬 수 있으며 이를 해결 하기 위한 오류 복구에 대한 연구가 요구 된다. 현재까지의 오류 복구에 대한 연구는 많은 경우가 정보가전기기와 서비스가 모델화 되어 있는 모델 기반에서 이루어져 왔다. 그러나 모델 기반의 오류 복구 시스템은 스마트 홈에 새로운 장치가 추가되거나 서비스가 변경 되는 경우와 같은 동적인 상황에서 발생하는 오류를 해결하지 못한다. 본 논문에서는 이러한 오류를 해결 하기 위하여 스마트 홈에 존재하는 정보가전기기들과 사용자 요구를 대변하는 서비스가 동적으로 추가되고 제거되는 상황을 반영하여 스마트 홈에 오류가 발생 했을 경우 이를 대처하기 위한 오류 복구 시스템을 제안한다. 따라서 제안된 오류 복구 시스템은 스마트 홈에 구성요소가 추가, 제거 되고 동적으로 변화하는 상황에서 발생하는 오류에서도 사용자에게 만족된 결과를 제공하는 것을 목적으로 하는 스마트 홈을 완성 할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존에 연구된 스마트 홈의 오류 복구 시스템에 대한 문제점에 대하여 기술하고, 3장에서는 본 논문에서 제안하는 상황기반의 오류 복구 시스템을 설계한다. 마지막으로 4장에서 본 논문의 결론을 언급한다.

¹ 본 연구는 서울시 산학연 협력사업(CR070019) 지원으로 수행되었음

² 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터(홈네트워크연구센터) 육성·지원사업의 연구결과로 수행되었음

2. 관련연구

스마트 홈에서의 오류 복구 시스템은 일상생활 환경에 편재되어 있는 다양한 컴퓨팅 자원으로부터 얻어진 스마트 홈의 상태 정보를 토대로 현재 스마트 홈의 상황을 인지하여 오류가 발생 했을 시에도 사용자에게 적절한 서비스를 계속하여 제공 해야 한다. 현재 스마트 홈에서의 오류 복구에 대한 많은 연구가 이루어 지고 있으나, 이들 연구[4][5]는 스마트 홈의 구성요소들이 고정되어있는 모델기반에서 이루어져 왔다. 그러나 사용자는 스마트 홈에 장치를 추가 하거나 제거하는 등의 동적인 행동을 하는 존재로서 스마트 홈은 이러한 동적 상황을 반영하여 오류 복구에 대한 연구가 진행 되어야 한다.

3. 상황기반 오류 복구 시스템 설계

제안하는 상황기반 오류 복구 시스템은 스마트 홈에서 발생 할 수 있는 오류를 스마트 홈의 상태 혹은 사용자의 패턴을 적용하여 해결하는 시스템이다. 본 장에서는 상황기반 오류 복구 시스템의 구조, 구성요소와 사용되는 용어를 정의하고, 스마트 홈에서 발생 할 수 있는 오류를 분류한 후 분류된 오류를 탐지하고, 복구하는 방법에 대하여 기술한다.

3.1 상황기반 오류 복구 시스템의 구조

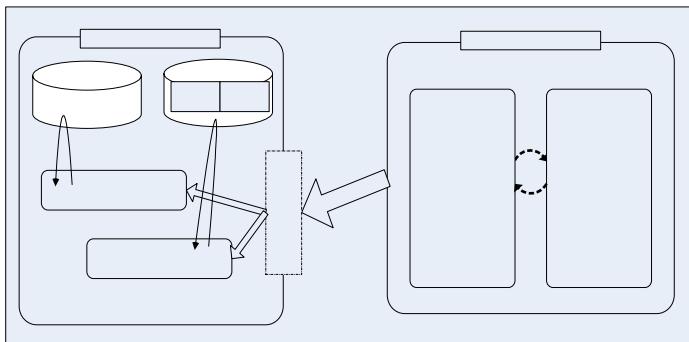


그림 1 상황기반 오류 복구 시스템

상황기반 오류 복구 시스템은 FaultDetector, FaultRecovery 그리고 Repository로 구성되어 있다.

- FaultDetector: 스마트 홈의 상황을 관찰하여 오류가 발생했을 경우, 이를 탐지하는 역할
- FaultRecovery: FaultDetector가 오류를 탐지 했을 경우, 이를 복구하는 역할

3.2 구성요소와 용어 정의

상황기반 오류 복구 시스템은 다음과 같은 구성요소를 포함한다.

Dev: 스마트 홈 내의 가전기기 혹은 센서

- Dev_attri: power 혹은 light와 같은 Dev의 속성
- Dev_val: Dev_attri의 값

- Env: 자연환경

- Env_attri: 온도, 습도 시간과 같은 Env의 속성
- Env_val: Env_attri의 값

- Server: 스마트 홈에서 변경되는 Dev와 Env 값들을 수집하고 관리하는 객체
- Situation: 스마트 홈의 상태
- Svc: 스마트 홈에서 사용자에게 제공되는 서비스
- Scenario: 사용자의 요구 사항을 정의한 구문으로, 조건에 대한 Dev의 실행으로 정의하고, XML을 이용하여 다음과 같이 표현한다.

Execute {Dev(Dev_func)} IF {Dev(Dev_state=value)}

상황기반 오류 복구 시스템은 다음과 같이 용어를 정의 한다.

- Base Factor: “온도가 증가한다” 혹은 “창문이 열렸다”와 같은 스마트 홈의 현재 상태를 표현하는 요소로, Rule, C, R을 구성한다. Base Factor는 아래와 같이 장치(Dev), 장치의 속성(Dev_attri) 그리고 장치의 속성에 대한 값(Dev_val)으로 정의 된다.

Dev(Dev_attri = Dev_val)

- Rule: Base factor 간의 관계를 정의하고 아래와 같이 표현된다. 표현된 형식은 “어떤 조건 C를 만족하였다면, 어떤 결과 R을 만족 해야 한다”와 같은 구문으로 해석된다.

Rule : C -> R

- C: Base factor로 구성되며, Rule이 만족 해야 하는 제약 조건(constraint)을 나타낸다. 또한, \bar{C} 는 C와 반대되는 상태를 나타낸다.
- R: Base factor로 구성되며, Rule에서 C가 만족 되었을 때의 결과(Result)를 나타낸다. 또한, \bar{R} 는 R과 반대 되는 상태를 나타낸다.

3.3 오류 분류

상황기반 오류 복구 시스템에서의 오류는 다음과 같이 분류 한다.

- Defect: 스마트 홈이 Rule의 어떤 조건 C는 만족 하였으나 결과 R을 만족 하지 못한 상태를 Defect로 정의하고 $C \cdot \bar{R}$ 로 표현한다.
- Conflict: 스마트 홈이 어떤 Rule1: $C1 \rightarrow R1$ 과 Rule2: $C2 \rightarrow \bar{R1}$ 을 동시에 수행하려 할 때 Conflict로 정의하고 $(C1 \rightarrow R1) \cdot (C2 \rightarrow \bar{R1})$ 로 표현한다.

3.4 FaultDetector

상황기반 오류 복구 시스템의 FaultDetector는 스마트 홈에서의 오류를 탐지 하기 위해 Rule의 수행 여부를 확인한다. 현재 수행되고 있는 Rule은 ActivateSet로 이동되고, 수행되지 않는 Rule은 DeactivateSet로 이동된다. 그림 2와 그림 3의 알고리즘은 오류 탐지 과정을 나타낸 것이고, 그림2의 $Fault_{cur}$ 와 $Fault_{Rule}$ 는 각각 현재 발생한 오류와 오류가 탐지된 Rule을 나타낸다.

```
Sa=NULL; /*ActivateSet*/ Sd=NULL; /*DeactivateSet*/
F = NULL; /*Base Factor*/
Sd <- 모든 Rule;
```

그림 2 상황기반 오류 복구 시스템 초기화 시

```
Faultcur = NULL;
FaultRule = NULL;

if(F){
    if( find ((Rule  $\ni$  F) in Sa )){
        if( (C in Rule)  $\ni$   $\bar{F}$  ) Sd <- Rule;
        if( (R in Rule)  $\ni$   $\bar{F}$  ) {
            Faultcur = Defect;
            FaultRule= Rule;
        }
    }
}

if( find ((Rule  $\ni$  F) in Sd )){
    if(((C in Rule) $\ni$ F) && (find((Rule' $\ni$  $\bar{R}$ ) in Sa))) {
        Faultcur = Conflict;
        FaultRule= Rule and Rule';
    }
}

If(Faultcur != NULL && FaultRule != NULL)
    send Faultcur and FaultRule to FaultRecovery;
```

그림 3 FaultDetector의 오류 탐지 과정

FaultDetector가 오류를 탐지 했다면, FaultDetector는 FaultRecovery에게 오류인 $Fault_{cur}$ 와 오류가 탐지된 Rule 인 $Fault_{Rule}$ 를 전달한다

3.5 FaultRecovery

상황기반 오류 복구 시스템은 사용자의 패턴과 스마트 홈에 존재하는 장치들을 파악하여 스마트 홈에 오류가 발생 했을 지라도 사용자에게 상황에 적절한 서비스를

제공한다. 오류 복구 방법은 탐지된 오류가 Defect 혹은 Conflict 에 대하여 다른 방법으로 이루어지며, 다음에서 각각에 대해 설명한다.

3.5.1 Detect에 대한 오류 복구

FaultRecovery가 FaultDetector로부터 Defect를 전달 받았다면, 현재 스마트 홈은 C를 만족 하였으나 R을 만족 하지 못한 $C \cdot \bar{R}$ 인 상태이다. FaultRecovery는 Defect에 대한 오류 복구 방법으로 R을 대신 만족 시킬 수 있는 Dev와 Dev_func를 추론하여 오류 복구를 수행한다. FaultRecover는 Dev와 Dev_func을 추론하기 위하여 표3에서 표현하고 있는 DevSubstitutionSet을 이용한다. DevSubstitutionSet은 스마트 홈의 상태 변경에 같은 역할을 하는 Dev와 Dev_func의 집합이다. DevSubstitutionSet는 같은 역할을 하는 Dev와 Dev_func의 인덱스를 나타내는 Idx, Defect가 발생 했을 때 대응할 수 있는 Dev를 나타내는 DevName, Dev_attri를 나타내는 DevAttri, Dev_func을 나타내는 DevFunction 그리고 Dev_func의 수행 후 DevAttri의 값의 변화를 표현하는 Variation으로 구성되어 있다.

스마트 홈은 Dev_func의 수행으로 변경되는 Env_attri의 변화 양상을 센서를 통해 관찰하고 기록한다. 기록한 정보를 통하여 표 1과 같은 DevSubstitutionSet을 생성한다. FaultRecovery의 Defect에 대한 오류 복구 과정은 그림 5의 알고리즘을 통해 이루어진다. 그림 4는 그림 5의 FaultRecovery의 Defect 오류 복구 과정에서 사용되는 용어를 정의한다

표 1 DevSubstitutionSet

DevSubstitutionSet				
Idx	DevName	DevAttri	Variation	DevFunction
1	Dev1	Dev_attri1	Variation1	Dev_func1
	Dev2	Dev_attri2	Variation2	Dev_func2

- Rule_{def}: C_{def}→R_{def}
 Rule_{def}: Defect가 발생한 Rule
 C_{def}: Rule_{def}의 제약 조건
 R_{def}: Rule_{def}의 결과
- SSet: DevSubstitutionSet
- (Dev1, Dev_func1):
 R_{def}의 Dev1, Dev_attri1 그리고 Dev_val1 을 이용하여 찾은 SSet로부터의 Dev 와 Dev_func
- (Dev2, Dev_func2):
 SSet로부터 찾은 (Dev1, Dev_func1) 대응의 Dev 와 Dev_func

그림 4 용어 정의

```

If(Defect && Ruledef){
  get (Dev1, Dev_attr1, Dev_val1) in Rdef;
  if(find(Dev1 in SSet){
    if((Dev_attr1 == value of Dev_attr1 in SSet)
      && (Dev_val1 == value of Variation in SSet)){
      Dev_func1 <- Dev_func in SSet;
      get Idx in SSet;
      get (Dev2, Dev_func2) in Idx;
    }
    send (Dev2, Dev_func2) to Server;
  }
  else
    notify the user;
}
    
```

그림 5 FaultRecovery의 Defect 오류 복구 과정

3.5.2 Conflict에 대한 오류 복구

FaultRecover가 FaultDetect로부터 Conflict를 전달 받았다면, 현재 스마트 홈은 Rule1: C1->R1과 Rule2:

C2-> R1 을 동시에 수행 하려 했기 때문에 2개의 상충된 Rule이 충돌이 난 상태이다. FaultRecovery는 Conflict에 대한 오류 복구 방법으로 Conflict를 일으킨 Dev들 중에서 사용자 패턴을 파악하여 하나의 Dev를 중지 시킴으로써 Conflict 오류에 대한 복구를 수행한다. 오류 복구 방법으로 스마트 홈에서의 사용자 패턴을 파악하여 이용함으로써 사용자에게 서비스를 계속하여 제공할 수 있다. 사용자 패턴은 사용자가 Env_attr에 대하여 Env_val을 어떻게 변경하려 하는지를 나타내는 사용자 환경패턴(User_EnvPattern)과 어떤 시간에 어떤 Dev를 동작 시킬지를 나타내는 사용자 시간패턴(User_TimePattern)으로 구성된다. 표2는 사용자패턴을 정의한다.

표 2 사용자 환경 패턴

User_EnvPattern					
Who	What	How	Where	Min	Max
Env를 변경한 대상	Env_attri	Env_attri 에 대한 Env_val 의 증/감	위치	Env_attri 에 대한 Env_val 의 최소값	Env_attri 에 대한 Env_val의 최대값

표 3 사용자 시간 패턴

User_TimePattern				
Who	What	Where	Start	End

Dev를 동작 시킨 대상	Dev	위치	Dev 동작 시작 시간	Dev 동작 중지 시간

FaultRecovery의 Conflict에 대한 오류 복구 과정은 그림 5와 같은 방법으로 이루어진다. 스마트 홈에 오류가 발생 했다면, Fault Recovery는 HOMI로부터 현재의 시간과 환경에 대한 Env_val을 읽어 온다. 읽어온 값들은 사용자패턴을 구성하고 있는 환경 패턴과 시간 패턴의 값과의 비교를 위해 사용된다.

Server로부터 읽어 온 Env_val이 사용자 환경 패턴의 Min과 Max 사이의 값이라면, 사용자는 현재 환경을 사용자 환경 패턴의 How와 같이 변경하려는 경향을 가지고 있는 것이다.

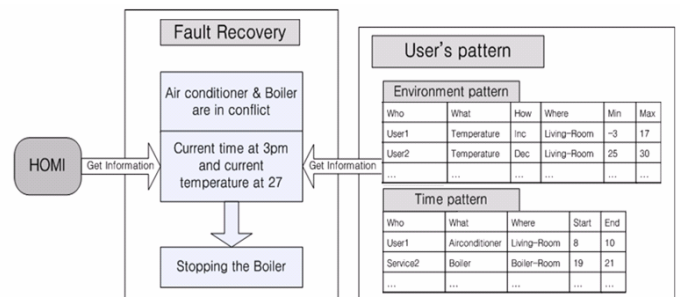


그림 6 FaultRecovery의 Conflict 오류 복구 과정

또한 Server로부터 읽어 온 현재의 시간이 사용자 시간 패턴의 Start와 End 사이의 값이라면, 사용자는 현재 시간에 What에 해당하는 스마트 홈의 Dev를 작동 시키는 경향이 있는 것이다. 이와 같이 사용자 패턴을 이용하여 스마트 홈에 발생한 Conflict를 해결 하는 것은 현재 시간과 환경에 사용자가 어떤 행동을 했는가를 기반으로 서비스를 제공할 수 있게 하여 스마트 홈에 오류가 발생한 상황에서도 사용자에게 적절한 서비스를 제공 가능하도록 하는 것이다.

4. 결론

본 논문에서는 스마트 홈의 동적인 상황에서 오류가 발생할 지라도 사용자에게 연속적인 서비스를 제공하기 위하여 오류를 탐지하고 복구하는 상황기반 오류 복구 시스템을 제안 하였다. 스마트 홈에서 Defect 오류에 대한 복구를 위해 대응 가능한 정보 가전 기기들을 집합화 해놓은 DevSubstitutionSet을 이용하였다. 스마트 홈은 미들웨어를 통해 정보가전기기가 추가 혹은 제거 되는 상황을 인지 할 수 있고, 센서를 통해 인지된 정보가전기기의 상태를 계속하여 업데이트 하므로 DevSubstitutionSet은 스마트 홈의 동적인 상황을 반영한다. 결과적으로 오류 복구 또한 스마트 홈의 상황을 기반으로 Defect에 대한 오류 복구가 이루어 짐으로써 상황기반의 오류 복구를 통해 사용자에게 끊임없는 서비스를 제공할 수 있다. 스마트

홈에서 두번째 오류로 정의한 Conflict의 오류 복구를 위해 사용자 패턴을 사용한다. 사용자 패턴은 동적인 스마트 홈 내의 사용자가 환경을 변화 시키는 양상과 시간에 대한 장치 사용의 선호도를 나타내므로 사용자 패턴을 이용한 오류 복구 시 스마트 홈의 상황을 기반으로 오류를 복구 한다고 할 수 있다.

[참고 문헌]

- [1] 이일주, 최진원, "A Study on the Design Guideline Development for Smart Home Based on Ubiquitous Technology System classification", 대한건축학회, vol. 24, No.2, 2004.10
- [2] S. Lee, K. Han, K. Lim, S. Ahn, "A Study on Location Supporting System for Smart Home", 한국정보과학회, Vol.32, No.1, 610pp, 2005
- [3] 최재동, 윤원식, 민승욱, 조위덕, " 유비쿼터스와 스마트 홈 서비스 기술", 방송공학회지, vol.9, 34-46pp, 2004.3
- [4] Insuk Park, Soon J. Hyun, Dongman Lee, Seongwoon Kim, "Context-Conflict Management for Context-Aware Applications in a Shared Pervasive Computing Environment," International Conference on Wireless Networks, 66-70pp, 2004
- [5] Nicholas Hanssens, Ajay Kulkarni, Rattapoom Tuchinda, and Tyler Horton, "Building Agent-Based Intelligent Workspaces," In Proceedings of the International Workshop on Agent for Business Automation, June. 2002