

# 지능형 서비스 로봇을 위한 자가 치유 방법론

민동욱\*, 고재현\*, 김영덕\*, 서정범\*, 박정민\*, 이현숙\*, 김훈기\*, 이은석\*\*

\*동양공업전문대학 전산정보학부, \*\*성균관대학교 정보통신공학부

{appeal, jhko, episod2, arki85, ya23ma, hsrhee, kimhk}@dongyang.ac.kr, eslee@ece.skku.ac.kr

## Self-Healing Methodology for Intelligent Service Robot

Dongwook Min\*, Jaeheon Ko\*, Youngduck Kim\*, Jeongbeom Seo\*,

Jeongmin Park\*, Hyunsook Rhee\*, Hoonki Kim\*, Eunseok Lee\*\*

\*School of Computing and Information, DongYang Technical College

\*\*School of Information and Communication Engineering, SungKyunKwan University

### 요 약

서비스 로봇이 발전함에 따라 인간이 직접 해결해야했던 작업들이 로봇에게 위임가능하게 되었으나, 서비스 로봇이 잘못된 행동을 하게 되는 경우 작업에 심각한 혼란이 야기되고, 사람의 안전과 직결될 수 있다. 이러한 문제점을 극복하기 위해 지능형 서비스 로봇을 위한 자가 치유 방법론 및 자가 치유 시스템의 구조를 제안한다. 제안 방법론은 7단계로 설계 및 구현되며, 시스템 구조는 모니터링, 진단 및 평가, 치유전략 계획 및 실행 순으로 수행된다. 제안사항을 통해 로봇의 고장으로 인한 불편을 최소화 하며 관리를 위해 발생하는 비용 절감이 가능하다.

### 1. 서론

인간에게 보다 편리한 서비스를 제공하는 지능형 서비스 로봇을 통해, 인간은 좀 더 실용적이고 실리적인 효과를 추구하며 이러한 환경에서 인간이 직접 해결해야했던 작업들은 로봇에게 위임이 가능하게 되었다[1].

그러나, 서비스 로봇의 잘못된 동작으로 인해 오류가 발생하게 되면 큰 혼란을 야기 시킬 수 있다. 또한 서비스 로봇은 사람들의 생활과 가장 가까운 곳에서 작동하게 되기 때문에 로봇의 오작동이 발생 시, 사람의 안전과 직결될 수 있다.

따라서 본 논문에서는 스스로 오작동을 치유하거나 감내하여 사용자의 불편을 최소화 할 수 있는 자가 치유(Self-Healing) 방법론을 제안한다. 제안 방법론은 다음과 같은 7단계로 구성되어 있다.

#### 1단계. 오류 분석(Fault Analysis)

- 목표 시스템의 정상 동작을 분석, 발생 가능한 오류의 분석, 오류 내용 정의

#### 2단계. 오류 발생조건 정의(Fault Event Definition)

- 오류 발생 이벤트 추출 및 인과관계 정의

#### 3단계. 오류 모델링(Fault Modeling)

- 오류들의 인과관계에 의한 트리 모델링

#### 4단계. 오류 상황 해석(Faulty State Interpretation)

- 오류상태를 해석할 수 있는 조건을 정의

#### 5단계. 치유전략 결정(Strategy Decision)

- 치유 전략 및 행위자 정의

#### 6단계. 시스템 모니터링(System Monitoring)

- 제약 조건에 따라 오류를 감지

#### 7단계. 진단(Diagnosis)

- 감지된 오류의 평가 및 진단

#### 8단계. 치유(Healing)

- 평가된 오류의 치유전략 계획
- 소프트웨어의 재구성

1-5단계는 자가 치유를 위한 오류 정의 및 설계를 위한 단계 이고, 6-8단계는 설계한 자가 치유 시스템의 테스트를 위한 실행 단계이다.

제안 시스템의 평가는 오류 주입(Fault Injection)기법[2]을 통하여 로봇의 이동에 관해 테스트 하였고, 이를 통해 로봇의 오류를 최대한 감내하여 일반 사용자들은 로봇의 서비스를 제공받는 것에 있어 불편을 최소화하는 것이 가능하다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 설명하고, 3장에서는 제안 사항을 서술한다. 4장에서는 구현 및 테스트를 기술하고, 5장에서는 결론을 맺는다.

### 2. 관련연구

#### 2.1 네트워크 기반의 지능형 서비스 로봇

우리는 자가 치유 개발 방법론의 적용을 위한 선행 연구로 지능형 서비스 로봇[3]을 제안했다.

제안 로봇은 EZ-Robomaster[4]와 Lego Brick을 이용하여 제작하였고, 사용자 애플리케이션과 이동 로봇, 웹 카메라, RFID 시스템으로 구성되어 다음과 같은 기능을 수행한다.

- 1) RFID를 이용하여 로봇의 위치를 인식
- 2) 사용자 애플리케이션을 통해 로봇을 제어
- 3) 네트워크를 통한 URC(Ubiquitous Robotic Companion)[5] 기반 로봇의 프로세싱 분담 기능

그러나, 사람들과 가장 가까운 곳에서 작동하는 로봇이기 때문에 로봇의 오작동이 발생할 경우 안전과 직결될 수 있다. 이러한 문제점을 최소화하기 위해 본 논문에서는 우리의 이전 연구를 목표시스템으로 적용하여 자가 치유 기법을 실험한다.

#### 2.2 컴포넌트 기반의 자가 치유 기법

커넥터를 통한 소프트웨어 아키텍처의 문제를 감지하는 방법론[6]은 다음과 같은 장점을 가지고 있다.

- 1) 이 소프트웨어 아키텍처는 컴포넌트와 컴포넌트들 사이의 커넥터로 구조화되어 컴포넌트의 태스크와 태스크들 사이에 있는 커넥터의 비정상적인 동작을 감지할 수 있다.
- 2) 시스템 모니터(System Monitor)를 통해 컴포넌트 사이의 입출력 메시지 통신 문제를 감지하며, 컴포넌트 모니터(Component Monitor)를 통해 컴포넌트 내부의 객체들의 문제를 감지한다.
- 3) 모니터들 내부에 있는 상태차트를 캡슐화 하여 메시지 사이의 패싱 관계를 비교할 수 있다.

본 논문에서는 위의 기법을 기반으로 로봇 소프트웨어 내부의 오류를 감지하고, 오류의 위치를 인식할 수 있도록 적용 하였다.

### 2.3 컴포넌트의 신뢰성 측정 기법

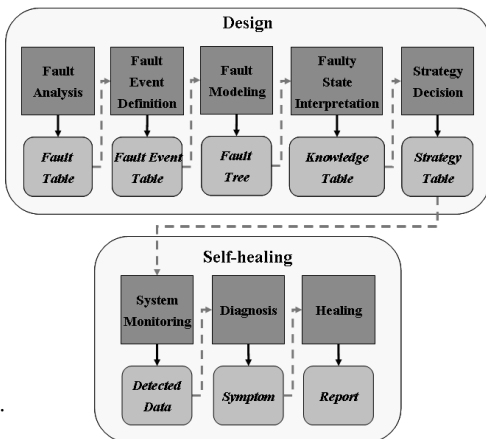
컴포넌트 신뢰성 측정 방법론[2]은 오류 주입(Fault Injection) 기법을 통해 시스템의 신뢰성을 측정하는 방법을 제안하였다.

오류 주입(Fault Injection)이란 소프트웨어의 신뢰성과 강건성을 테스트하는 방법으로, 발생 가능한 오류를 주입하여 소프트웨어가 오류에 대해 대처하는 방법을 분석하기 위해 사용되는 기법이다. 분석된 내용을 통해 소프트웨어에 존재하는 문제를 제거하고, 결함을 감내할 수 있는 다양한 방법들을 추가, 보완하여 강건한 소프트웨어를 만들려는 방법이다. 이러한 장점을 이용 가능하도록, 본 논문에서는 오류 주입 기법을 평가를 위해 적용하였다.

## 3. 제안 사항

### 3.1 자가 치유 방법론

본 절에서는 지능형 서비스 로봇의 오작동을 최소화하기 위해 (그림 1)과 같이 8단계로 자가 치유 방법론을 제안한다.



(그림 1) 자가 치유 프로세스

### 1단계- 오류 분석(Fault Analysis)

오류 분석 단계는 목표 시스템의 정상 동작을 분석하여 발생 가능한 오류를 추출하고 정의하여 자가 치유를 위해 요구되는 자료로 이용한다. 자가 치유 시스템 개발자는 시스템 개발에 앞서 적용할 목표 시스템의 정상 동작을 반드시 알아야 한다. <표 1>은 오류 분석 단계를 통해 생성된 오류 테이블(Fault Table)의 일부분이다.

<표1> Fault Table

구분	컴포넌트		오류
로봇 (센서노드)	센서	컴퍼스	자기장에 의한 측정값 오류
		초음파	특정 데이터 검출
	모터	스텝모터	양쪽 모터의 회전수가 다름
사용자 애플리케이션 (싱크노드)	통신	bluetooth	데이터 송수신 불가
	카메라	웹 카메라	영상 데이터 없음
	통신	인터넷	접속 불가
⋮	⋮	⋮	⋮

로봇과 UI에서 일어날 수 있는 오류로 크게 구분 지었으며, 컴포넌트별로 오류를 추출하여 정리한다.

### 2단계- 오류 발생조건 정의(Fault Event Definition)

작성된 오류 테이블에 기반을 두어 파악한 오류의 발생 조건(Fault Event)을 정의하고, 발생 원인이 되는 오류(Causative States)와 결과가 되는 오류(Resultant State)로 관계를 정의해 <표2>와 같이 오류 발생조건 테이블(Fault Event Table)을 작성한다.

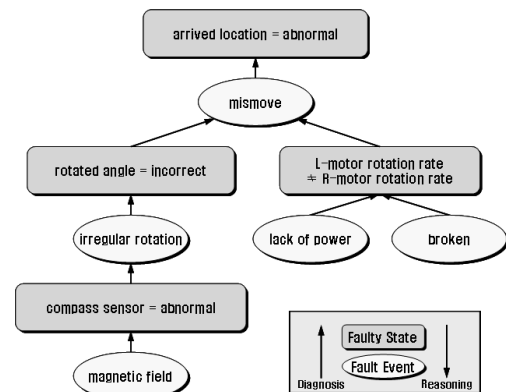
<표2> Fault Event Table

발생 조건	원인 오류상태	처리	결과 오류상태
mismove	rotated angle = incorrect OR L-motor rotation rate ≠ R-motor rotation rate	move()	arrived location = abnormal
irregular rotation	compass sensor = abnormal	turn()	rotated angle = incorrect
magnetic field		move()	compass sensor = abnormal
lack of power		move()	step motor = abnormal
broken		move()	step motor = abnormal

잘못된 이동(mismove)의 경우 ‘회전된 각도가 부정확’하거나 ‘왼쪽과 오른쪽모터의 회전수가 다름’인 상태에서 이동을 하게 되면 ‘도착한 좌표가 비정상’인 상태로 전이하게 된다.

### 3단계- 오류 모델링(Fault Modeling)

발생 조건 테이블에 기반을 두어 오류들의 연관성에 따라 (그림 2)와 같이 오류 트리를 작성한다.



(그림 2) Fault Tree

이를 통해 특정 오류가 발생한 원인을 추론할 수 있고 발생할 수 있는 오류의 결과를 예측 또는 진단할 수 있다.

4단계- 오류 상태 해석(Faulty State Interpretation)

로봇이 오류상태를 해석할 수 있는 조건을 정의하여 <표3>과 같이 지식 테이블(Knowledge Table)로 작성한다.

<표3> Knowledge Table

계약 조건	오류 상태
changed angle > limit	compass sensor = abnormal
L-motor rotation rate ≠ R-motor rotation rate	step motor = abnormal

5단계- 치유전략 결정(Strategy Decision)

진단된 증상에 대한 치유행위자 및 전략을 정의 하여 <표3>와 같이 전략 테이블(Strategy Table)을 작성한다.

<표4> Strategy Table

오류 상태	치유 전략	행위자
compass sensor = abnormal	초음파 센서로 벽과의 평행 보정	로봇(센서노드)
step motor = abnormal	회는 방향의 모터 회전수를 높임	로봇(센서노드)

6단계- 시스템 모니터링(System Monitoring)

1-5단계의 산출물인 테이블 및 오류 트리를 기반 하여 시스템의 오류를 감지한다.

7단계. 진단(Diagnosis)

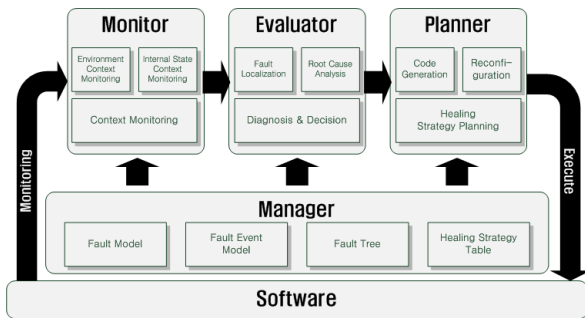
모니터링 단계에서 감지한 오류들에 대해 지식테이블을 참조하여 증상을 진단하고, 치유 실행 여부에 대해 평가한다.

8단계. 치유(Healing)

진단결과를 통해 치유가 필요하다고 판단된 경우, 증상에 따른 치유 행위자를 결정하고 치유 전략을 계획하여 실행한다.

3.2 자가 치유 시스템 구조

제안 방법론을 수행하기 위한 로봇의 자가 치유 시스템은 (그림 3)과 같이 구성한다.



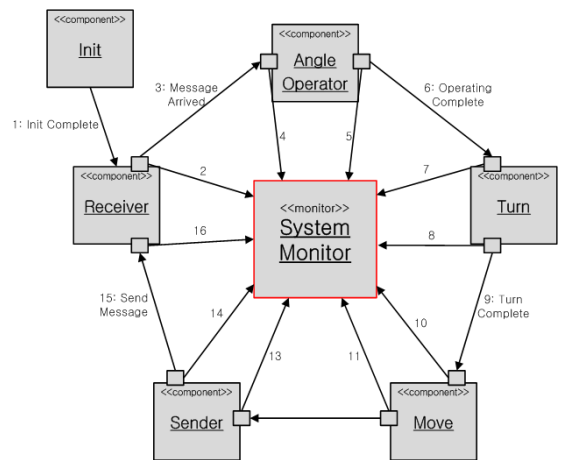
(그림 3) 자가 치유 시스템 구조

- 관리기(Manager)는 소프트웨어와 자가 치유 시스템 사이에 존재하고, 3.1절의 1-5단계에서 로봇으로부터 도출한 정상동작, 오류 테이블(Fault Table) 등을 저장하고 있다.
- 감지기(Monitor)는 소프트웨어에서 발생하는 오류를 관리기가 가지고 있는 오류 발생조건 테이블(Fault Event Table)을 통해 감지하고, 평가기로 감지된 오류정보를 전송함으로써 6단계를 수행한다.
- 평가기(Evaluator)는 감지기로부터 수신한 오류정보를 오류 테이블 및 오류 트리에 정의된 오류정보와 비교하여 오류의 종류를 분석, 진단하고 치유 실행 여부를 결정한다. 치유가 필요하다고 판단된 경우 계획자로 오류정보를 전송함으로써 7단계를 수행한다.

- 계획자(Planner)는 평가기로부터 수신한 오류정보를 치유전략 테이블(Healing Strategy Table)에 기반을 두어 치유전략을 결정하고, 코드를 생성하여 소프트웨어를 재구성 또는 사용자 애플리케이션으로 알린다. 이는 8단계에 해당한다.

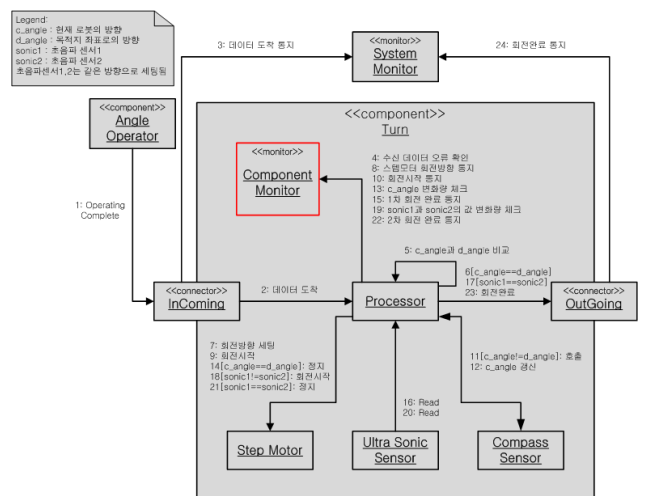
4. 구현 및 테스트

본 절에서는 제안된 자가 치유 방법에 따라 목표시스템인 지능형 서비스 로봇에 적용한다. 목표 시스템은 로봇과 사용자 애플리케이션으로 구성되어 있다. 로봇을 위한 자가 치유 시스템은 컴포넌트 기반으로 설계 하고, 송신과 수신 두 개의 커넥터를 통해 컴포넌트간의 통신이 이루어진다. 각 컴포넌트는 커넥터를 통해 메시지를 송수신하며, 컴포넌트 간의 메시지는 시스템 모니터에 의해 감시된다. (그림 4)는 컴포넌트와 커넥터를 적용한 로봇 시스템의 구조이며, 시스템 모니터가 컴포넌트 간에 통신하는 메시지를 감시하고 있다.



(그림 4) 컴포넌트와 커넥터를 적용한 시스템 모니터

컴포넌트 내부에는 태스크와 수동적 객체들로 구성된다. 컴포넌트 모니터는 각각의 컴포넌트 내부에 존재하여 태스크 및 수동적 객체들의 통신을 감시한다. (그림 5)는 컴포넌트 내부 상태 감지를 위한 컴포넌트 모니터의 행동을 나타내고 있다.



(그림 5) 컴포넌트 내부상태 감지를 위한 컴포넌트 모니터

모니터들은 로봇의 행동을 기술한 상태차트를 캡슐화 하여 보관하고 있으며, 제한시간 내에 상태의 전이가 정상적으로 이루어지지 않은 경우 비정상동작으로 인식하게 된다.



(그림 6) 사용자 애플리케이션

(그림6)은 로봇과 상호작용하는 사용자 애플리케이션을 나타내고 있으며, Visual Basic 6.0을 이용해 구현했다. 구체적인 사용자 애플리케이션의 기능은 다음과 같다.

- 1) *Map* : 마우스의 드래그나 클릭으로 로봇의 이동경로 설정할 수 있으며, 로봇의 현재위치를 이미지로 표시한다. 또한, 장애물이나 자기장 영역 등 이동 제한구역을 표시하여 사용자에게 알려준다.
- 2) *Monitor* : 인터넷으로 웹 카메라에 접속하여 영상을 보여주고, 카메라의 줌 또는 방향을 제어한다.
- 3) *Command* : 십자버튼으로 로봇을 수동 조작할 수 있고, 로봇의 코드를 재인스톨 하는 등 각종 명령 버튼이 있다.
- 4) *Log* : 로봇의 이동경로, 시작/종료 시간, 오류의 종류 및 발생 시각, 치유 결과 등을 기록하고, 저장하여 관리한다.
- 5) *State Monitor* : 로봇 센서들의 상태를 오류의 심각한 정도에 따라 색상으로 표시한다. 센서를 클릭하면 해당 센서의 오류 발생 정보를 확인할 수 있다.

#### 4.2 테스트

본 논문에서는 제안 시스템을 평가하기 위해 오류 주입 기법을 사용했다. 자가 치유 로봇의 강건성을 테스트하기 위하여 다음과 같은 오류를 발생시키고, 앞서 정의한 치유 전략을 정확히 수행하는지 확인했다.

**오류주입1-** 한쪽 스텝모터의 회전속도를 고의적으로 감소시켜 경로에서 이탈시킨다.

**해결방안-** 이동 중 컴퍼스 센서로 각도의 변화량을 체크하고 있는 로봇은 모터의 회전수가 달라 경로를 이탈할 경우 '경로가 휘어짐'으로 오류를 인식하고 휘어진 쪽 모터의 회전수를 증가시킨다.

**오류주입2-** 로봇이 회전할 때 자기장을 발생시켜 정확한 각도로의 회전을 방해한다.

**해결방안-** 회전 중 컴퍼스 센서로 각도의 평균 변화량을 체크하고 있는 로봇은 자기장으로 인해 급격한 각도의 변화가 있을 경우 '자기장으로 인한 측정값 오류' 오류를 인식하고 초음파 센서로 벽과의 평행을 맞추어 각도를 보정한다.

#### 4.3 테스트 평가

오류 주입을 통해 다음과 같은 결과를 얻었다(<표5>참조). 테스트 결과 예측 가능한 오류에 대해 치유전략을 세

워 사용자의 불편함을 최소화 할 수 있었다.

&lt;표5&gt; 테스트 결과

오류	치유행위자	치유전략	수행여부
오류주입1	로봇	모터 회전수 변경	O
오류주입2	로봇	초음파 센서로 보정	O

평가를 위해 사용한 오류 주입 방법은 간단한 문제에 대한 전략을 적용한 것이다. 그러나 보다 진보된 자가 치유를 위해서는 보다 복잡한 상황에서 일어나는 문제를 해결해야 할 필요가 있다. 이는 향후연구에서 다룰 예정이다.

#### 5. 결론

본 논문은 지능형 서비스 로봇의 강건성을 높이기 위한 자가 치유 방법론과 이를 적용하는 자가 치유 시스템의 구조를 제안하였다. 제안 방법들을 통해 1) 사용자는 로봇의 고장으로 인한 불편을 최소화 할 수 있고, 2) 유지보수 시 발생하는 비용 절감 효과를 기대할 수 있다. 그러나 예측하지 못한 오류를 감지해내지 못하는 문제점이 있었다. 이에 대한 모니터링 기법이나 치유방안에 대해서는 향후 연구로 나타내며, 오류정보 및 테스트 결과 등 기록된 오류정보 데이터들을 통해 진단 가능한 자가 치유 방법으로 확장할 계획이다.

#### 참고문헌

- [1] 김두식, "美, 최근 산업용 로봇 시장동향", <http://www.globalwindow.org/>, 2007. 06.
- [2] Leslie Cheung, "Early Prediction of Software Component Reliability" ICSE 08, May 10 - 18, 2008. 05.
- [3] 고재현, "Ez-Robomaster를 이용한 네트워크 기반의 서비스 로봇 시스템의 설계와 구현" 한국정보과학회 제 35회 춘계학술대회 2008. 10.
- [4] EASYTECH, <http://www.ezlab.com>
- [5] 김 현, 조영조, 오상록, "URC(Ubiquitous Robotic Companion) 네트워크 기반 서비스 로봇" 한국정보과학회 학회지 VOL. 24 NO. 03 pp. 0005 ~ 0011, 2006. 03.
- [6] Michel E. Shin, "Detection of anomalies in software architecture with connectors" Science of Computer Programming 61 2006.
- [7] 김재선, "자기적응형 소프트웨어를 위한 목표 기반의 외부상황 평가기법", 정보과학회논문지 소프트웨어 및 응용 제 33 권 제 3 호 2006. 03.
- [8] Bill N. Schilit, Norman Adams and Roy Want, "Context-Aware Computing Application" IEEE Workshop on Mobile Computing System and Application, 1994. 12.