

컴포넌트 기반 로봇의 결점분류 모델

김제영*, 김종영*, 윤희병*
*국방대학교 국방정보체계학과
e-mail:8585jy@naver.com

A Model of Faults Classification in Component based Robots

Jaeyoung Kim*, Jongyoung Kim*, Heebyung Yoon*

*Department of Defense Information System, Korea National Defense University

요 약

로봇은 인간의 생명과 밀접하게 관련되어 있기 때문에 높은 안전성과 신뢰성이 요구된다. 특히 최근에는 지능형 서비스 로봇의 등장과 더불어 증가되는 로봇 개발환경의 복잡성 문제와 로봇의 신속한 개발 및 유지보수 문제를 해결하기 위해 컴포넌트 기반의 로봇 플랫폼에 대한 연구가 증가하고 있으며, 특히 로봇 플랫폼을 구성하는 컴포넌트의 장애문제 해결이 중요한 이슈로 대두되고 있다. 따라서 본 논문에서는 컴포넌트 기반 로봇의 결점분류 절차를 제시하고 로봇의 결점타입과 로봇 플랫폼과의 상관관계를 분석하고 로봇 플랫폼에 결점타입을 매핑하여 최종적으로 컴포넌트 기반 로봇의 결점분류 모델을 제안한다.

1. 서론

지능형 서비스 로봇에 대한 요구사항이 증가하면서, 로봇은 인간이 생활하는 공간과 밀접하게 관계를 맺고 있다. 산업용 로봇의 경우에는 정적인 환경에서 미리 정해진 계획에 따라 작업을 수행하였지만 지능형 서비스 로봇은 사람이 생활하는 공간에서 청소, 가사 도우미 등의 힘든 일을 대신 수행해야 한다. 따라서 지능형 서비스 로봇이 운용되는 환경 역시 산업용 로봇과 비교해서 매우 동적이고 인간의 안전과 보다 밀접하게 관련되어 있다. 즉 지능형 로봇은 산업용 로봇에 비해 안전성과 신뢰성이 더 요구된다고 할 수 있다.

로봇의 안전성 및 신뢰성과 관련된 분야 중에서 대표적인 결점감내(Fault Tolerance) 분야는 오래전부터 학계를 중심으로 연구되어 왔지만 대부분의 연구가 산업용 로봇과 같은 정적인 환경과 하드웨어 중심의 연구가 대부분이었으며, 특히 결점의 예방 및 복구, 그리고 결점의 고립과 같은 연구가 대부분을 차지하고 있었다.

최근 로봇의 연구 추세는 다양한 멀티미디어를 제공하는 지능형 서비스 분야, 소프트웨어 복잡도가 증가함에 따른 재사용 및 시장 적시성 해결 분야, 그리고 로봇 컴포넌트 간의 상호운용성 해결 분야 등이며, 이러한 문제를 해결하기 위해 소프트웨어 공학에 기반을 둔 컴포넌트 기반의 로봇 플랫폼에 대한 연구가 증가하고 있으며, 특히 로봇 플랫폼을 구성하는 컴포넌트의 장애문제 해결이 중요한 이슈로 대두되고 있다.

본 논문에서는 컴포넌트 기반 로봇의 결점분류 절차를

먼저 제시하고 이 절차에 따라 로봇의 결점타입과 로봇 플랫폼과의 상관관계를 분석하며, 분석결과를 바탕으로 로봇 플랫폼에 결점타입을 매핑시켜, 최종적으로 컴포넌트 기반 로봇의 결점분류 모델을 제안한다.

2. 관련연구

2.1 용어 정의

일반적으로 에러(Error), 결점(Fault) 및 장애(Failure)는 동일한 의미로 사용되기도 하지만 결점감내 컴퓨팅 분야에서는 명확하게 구분되어 사용된다[1]. 여러 문헌[3-15]에서 세 가지 용어에 대해 살펴본 결과, 학문분야에 따라 조금씩 다른 의미로 사용되고 있으나 대부분 IEEE 표준 용어를 따르고 있음을 알 수 있다. 본 연구에서는 IEEE 표준용어[2]를 준용하여 <표 1>과 같이 정의한다.

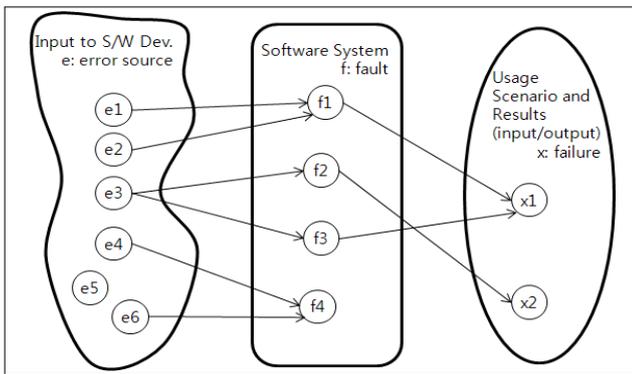
<표 1> 에러, 결점, 장애의 용어 정의

구분	용어 정의
에러	부정확한 결과를 초래하는 인간의 활동
결점	소프트웨어에서 에러의 외적표현
장애	시스템 또는 컴포넌트가 명세된 성능 요구사항 안에서 요구되는 기능을 수행하지 못하는 것

<표 1>을 세부적으로 설명하면, 에러는 넓은 의미에서 결점, 장애 그리고 실수 등을 포함한 개념으로 사용되기도 하지만 결점감내 분야에서는 소프트웨어 개발 및 운용 중에 인간에 의해 초래된 부정확한 결과로 볼 수 있다. 다시 말해, 에러는 개발단계에서 개발자에 의해 초래되는 명세

서와 실제의 차이(값 또는 조건)나 운용과정에서 사용자에게 의해 초래되는 부정확한 결과라고 볼 수 있다. 결점 역시 일반적인 의미에서 “에러” 또는 “버그”라는 용어와 유사하게 사용되지만 결점감내 분야에서는 시스템 또는 컴포넌트의 장애를 유발하는 원인으로 하드웨어 또는 소프트웨어에서 부정확한 결과를 발생하는 것을 의미한다. 장애는 시스템 또는 컴포넌트 명세서에 명시된 성능요구사항과 직접적으로 관련되어 있으며 명세서에 의해 정의되고, 사용자에게 의해 외부적으로 관찰 가능한 특성을 갖고 있다.

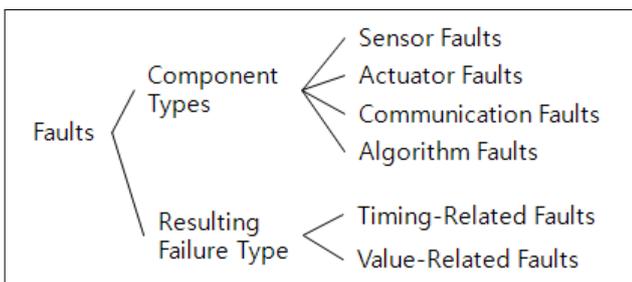
에러, 결점 및 장애 간의 관계에서 에러를 결점의 원인으로 보는 시각과 결점을 에러의 원인으로 보는 시각이 있으나, 본 논문에서는 에러를 결점의 원인으로 보며 이들 간의 관계는 (그림 1)과 같다.



(그림 1) 에러, 결점 및 장애 간의 관계

2.2 로봇의 결점분류 모델

지금까지 로봇의 결점분류 모델에 대한 많은 연구가 진행되어 왔으나 대부분 하드웨어 관점이거나 아니면 특정 애플리케이션에 국한되어 연구되고 있다[14-16]. 또한 결점분류 모델이 (그림 2)와 같이 일반적인 경우가 대부분이다[17].



(그림 2) 로봇 결점분류 모델

2.3 컴포넌트 기반 로봇 소프트웨어 플랫폼

최근 소프트웨어 공학을 기반으로 컴포넌트 기반 로봇의 소프트웨어 플랫폼에 대한 연구가 활발하게 진행되고 있다. 대표적인 예로, OROCOS[18], OMG의 RTC(Robot Technology Component)[19], ETRI의 OPRoS(Open Platform for Robot Service)[20] 등이 있다. 이러한 로봇 소프트웨어 플랫폼은 공통적으로 로봇 소프트웨어 컴포넌트 기술, 로봇 소프트웨어 프레임워크 기술, 통합개발환경

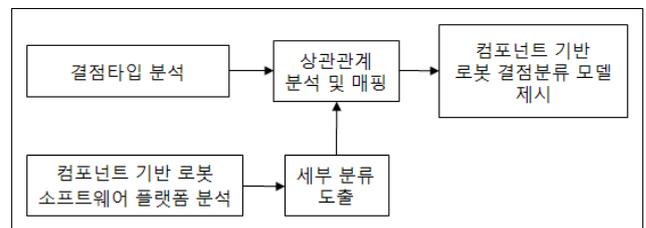
을 포함하고 있다.

ETRI의 OPRoS 플랫폼을 기준으로 분석해보면, 로봇 컴포넌트 간 서비스 호출이나 데이터/이벤트 교환은 포트를 통해 이루어진다. 컴포넌트는 서비스 포트를 통해 다른 컴포넌트가 제공하는 서비스를 호출하거나 해당 컴포넌트의 속성에 접근할 수 있으며, 데이터/이벤트 포트를 통해 해당 컴포넌트에 데이터나 이벤트를 전달할 수 있다. 로봇 프레임워크는 컴포넌트 간의 연결 및 관리를 체계적으로 지원하고, 운영체제, 하드웨어 간의 독립성을 유지하고 유지보수를 용이하게 하는 역할을 한다. 컴포넌트 관리자, 이벤트 관리자, QoS관리자, 타이머 관리자, 결함 및 보안 관리자 등으로 구성된다. 로봇 통합개발환경은 로봇에 필요한 소프트웨어 컴포넌트 및 콘텐츠를 쉽고 편리하게 개발 가능하게 하고, 개발된 컴포넌트 및 콘텐츠를 시뮬레이션 또는 에뮬레이션 할 수 있는 개발환경을 제공한다.

3. 컴포넌트 기반 로봇 결점분류 모델

3.1 컴포넌트 기반 로봇의 결점분류 접근절차

본 논문에서는 로봇의 결점타입과 컴포넌트 기반 로봇의 소프트웨어 플랫폼을 분석하고 이를 기반으로 상관관계를 파악한다. 그리고 결점타입과 플랫폼을 매핑하여 컴포넌트 기반 로봇의 결점분류 모델을 제시한다. 이와 같은 로봇의 결점분류 접근절차는 (그림 3)과 같다.



(그림 3) 컴포넌트 기반 로봇의 결점분류 접근절차

3.2 결점타입 분석

로봇의 결점타입을 분석하기 위해, 여러 참고문헌[3-15]에서 나타난 결점을 추출하고 이를 기반으로 로봇에 적용이 가능한 결점타입을 분류한다.

소프트웨어에서 가능한 결점의 유일한 타입은 소프트웨어 개발과정에서 이입된 설계결점이라 할 수 있다[21]. 하지만 개발과정에서 단계별로 많은 테스트 과정을 거쳤다 하더라도 운영과정에서 장애는 발생할 수 있다[14]. 소프트웨어 비정상 타입에 대한 분류는 IEEE 표준[22]이 대표적이며, 이는 운영체제, 데이터베이스관리시스템, 애플리케이션, 펌웨어 및 임베디드 소프트웨어 등 어떤 종류의 소프트웨어에도 상관없이 적용 가능하고 또한 제품의 전 범위의 생명주기에 적용이 가능하다[23]. IEEE 분류표준을 기준으로 여러 참고문헌에 나타난 로봇 소프트웨어 결점을 분류한 결과, 로직(Logic), 계산(Computation), 인터페이스(Interface), 시간(Time), 데이터 처리(Data Handling), 데이터(Data) 등이 결점타입이라 할 수 있다.

3.3 컴포넌트 기반 로봇 소프트웨어 플랫폼 분석

OPRoS 컴포넌트는 컴포넌트가 제공하는 인터페이스를 통해 접근되고 재사용 및 교체가 가능한 소프트웨어 모듈로서, OPRoS 플랫폼에서 사용할 수 있도록 개발되어 있다. OPRoS 컴포넌트의 대상은 로봇에서 사용되는 장치, 즉 기본 소프트웨어 컴포넌트와 로봇의 응용 동작에 필요한 응용 소프트웨어 컴포넌트로 크게 구분되며 <표 2>와 같다[24].

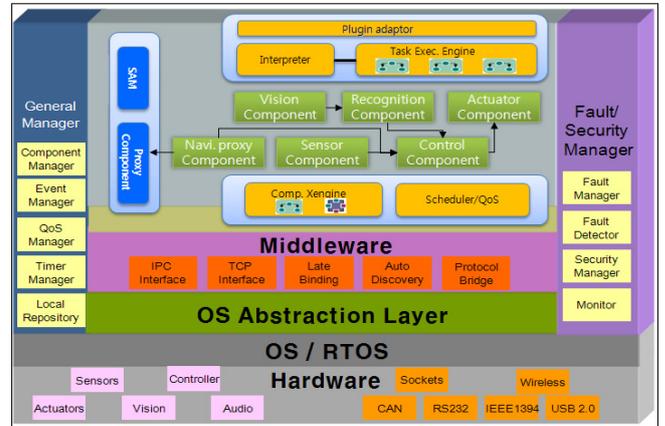
<표 2> OPRoS 컴포넌트 타입

대분류	중분류	소분류
기본SW 컴포넌트	디바이스 (Device)	액추에이터, 서보 액추에이터, 매니플레이터, 그리퍼
	버스(Bus)	범용비동기통신(UART),CAN통신
	카메라	카메라
	입출력 (IO)	아날로그 입력/출력/입출력, 디지털 입력/출력/입출력, 아날로그 디지털 입출력
	센서 (Sensor)	범퍼, 관성측정유닛(IMU), 적외선(IR), 레이저스캐너, 자이로, 위치, 터치, 초음파
	사운드	스피치
응용SW 컴포넌트	-	얼굴인식, 물체인식, 네비게이션, 충돌회피 등

OPRoS 컴포넌트는 개별 컴포넌트가 속한 그룹의 속성에 따라 공통적인 특성을 갖는 공통 부분(표준 컴포넌트)과 컴포넌트의 개별 특성(API)을 나타내는 응용부분으로 구분된다. 즉 OPRoS 컴포넌트는 표준 컴포넌트와 표준 API가 조합되어야 하나의 컴포넌트로서 완성된 기능을 수행할 수 있도록 구조화되어 있다.

OPRoS 소프트웨어 프레임워크는 (그림 4)와 같은 추상화된 계층구조를 갖고 있다. 즉 운영체제, 미들웨어, 공통 응용 소프트웨어로 구성되어 있다. 미들웨어에는 로봇

을 실행하는데 필수적인 컴포넌트 실행엔진, 통신관리기, 미들웨어/컴포넌트 어댑터 등이 있으며, 로봇의 응용에 따라 선택적으로 적용 가능한 공통 응용 소프트웨어에는 이벤트 관리자, 결함관리기, 스크립트 해석/플러그인 관리기, 보안관리기, 시간관리기 등이 있다.

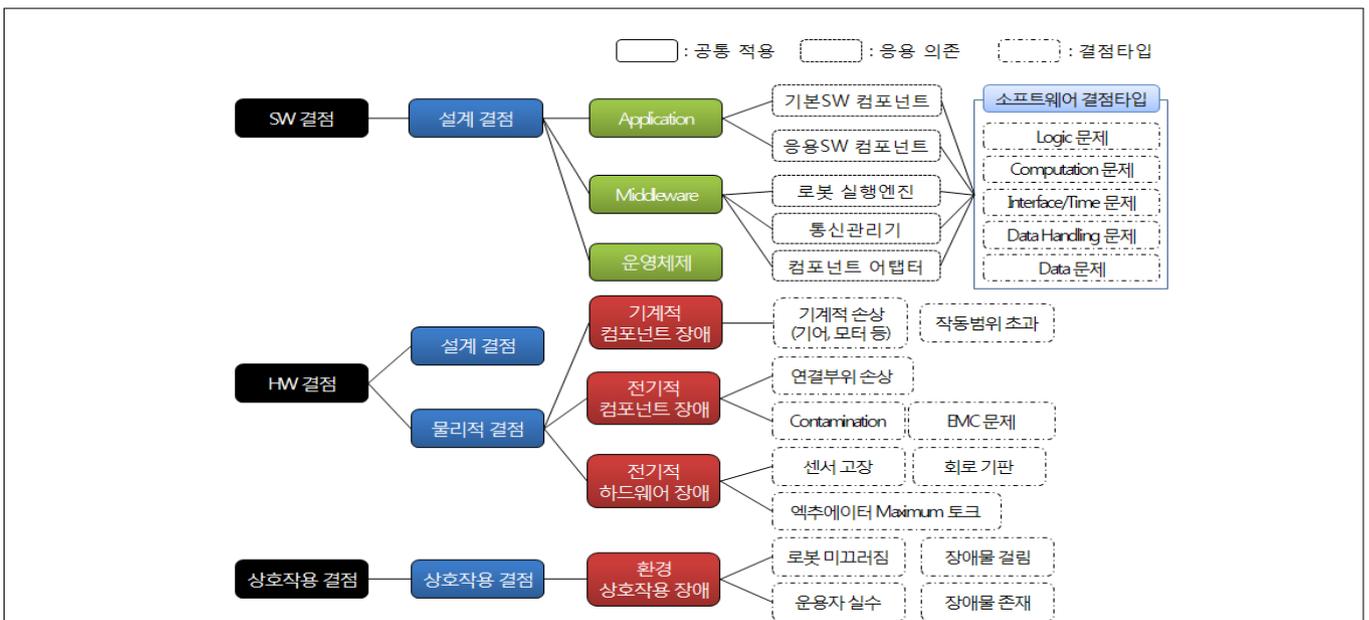


(그림 4) OPRoS 소프트웨어 프레임워크 구조

3.4 상관관계 분석을 통한 로봇의 결점분류 모델 도출

로봇의 결점은 로봇 내부의 결점과 외부의 결점으로 구분할 수 있다. 로봇의 내부결점은 로봇의 응용에 따라 구성되는 소프트웨어 결점과 하드웨어 결점으로 구분되며, 로봇의 외부결점은 로봇이 운용되고 있는 외부환경 하에서 발생하는 상호작용 결점으로 구분된다. 상관관계를 분석을 통해 본 전체적인 컴포넌트 기반의 로봇의 결점분류 모델은 (그림 5)와 같다.

요약하면 설명하면, 로봇의 소프트웨어 결점은 컴포넌트 기반 로봇 소프트웨어 플랫폼을 구성하는 로봇 소프트웨어 컴포넌트와 소프트웨어 프레임워크로 구분할 수 있으며, 이는 소프트웨어 결점 타입과 관련되어 있다. 로봇



(그림 5) 컴포넌트 기반 로봇의 결점분류 모델

의 하드웨어 결점은 로봇을 구성하는 기계적인 컴포넌트(Mechanical Component)와 전기적인 컴포넌트(Electrical Component) 결점으로 구분할 수 있으며, 이는 하드웨어 컴포넌트 결점 타입과 관련되어 있다. 로봇의 상호작용 결점은 로봇이 운용되는 외부환경 및 운용자 등과 관련되어 있다.

4. 결론

로봇의 결점은 개발하고자 하는 로봇 응용에 의존적인 특성을 갖고 있으나, 로봇의 안전성과 신뢰성을 높이기 위해서는 결점에 대한 보다 체계적인 접근이 필요하다. 따라서 본 논문에서는 로봇의 결점타입과 컴포넌트 기반 로봇의 소프트웨어 플랫폼과의 상관관계를 기반으로 컴포넌트 기반 로봇의 결점분류 모델을 제안하였다.

제안한 결점분류 모델은 최근 연구되고 있는 컴포넌트 기반 로봇에 적용할 수 있도록 설계하였으며 로봇 플랫폼에 결점타입을 매핑하여 로봇 개발자가 보다 로봇의 결점을 이해하기 쉽도록 제시하였다. 하지만 결점은 개발하고자 하는 로봇의 응용에 밀접하게 관계되어 있기 때문에 개발자는 제안한 결점분류 모델을 기반으로 보다 세부적인 결점을 분류하고, 이를 기반으로 결점감내 기법 적용을 위한 계획을 수립해야 한다.

참고문헌

- [1] Israel Koren, C. Mani Krishna, "Fault-Tolerant Systems," Elsevier, pp.2-3, 2007.
- [2] IEEE, "IEEE Standard Glossary of Software Engineering Terminology," 1990.
- [3] Jeff Tian, "Software Quality Engineering," IEEE Computer Society, p.20, 2005.
- [4] H. J. Ahn, D. S. Lee and S. C. Ahn, "OPRoS based Fault Tolerance Support for Reliability of Service Robots," *Journal of Institute of Control, Robotics and Systems*, Vol. 16, No. 6, June. 2010.
- [5] Leonardo Mariani, "A Fault Taxonomy for Component-based Software," Elsevier, 2003.
- [6] Atef Mohamed, Mohammad Zulkernine, "On Failure Propagation in Component-Based Software Systems," The 8th International Conference on Quality Software, 2008.
- [7] Grunske, L., Kaiser, B., "Automatic Generation of Analyzable Failure Propagation Models from Component-Level Failure Annotations," Proc. of the 5th International Conference on Quality Software (QSIC 2005), Australia, pp.117-123, Sep. 2005.
- [8] Goutam Kumar Saha, "Software Based Fault Tolerance-a Survey," 2006.
- [9] A.K Somani and N.H. Vaidya, "Understanding fault tolerance and reliability," IEEE Computer Society, pp. 45-50, April. 1997.
- [10] Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice 2nd," Addison-Wesley, p.101, 2003.
- [11] Ian Sommerville, "Software Engineering 8th," Addison-Wesley, pp.51-54, 2007.
- [12] OMG, "Fault Tolerant CORBA Ver1.0," p.80, 2010.
- [13] HA Forum, "Providing Open Architecture High Availability Solutions Revision1.0," p.51, 2001.
- [14] Vandi Verma, Geoff G., Reid S., Sebastian., "Real-Time Fault Diagnosis," *IEEE Robotics and Automation Magazine*, Vol.11, Issue2, pp.56-66, June. 2004.
- [15] Rolf Isermann, "Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance," Springer, pp.31-32, 2006.
- [16] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transaction On Dependable and Secure Computing*, Vol. 1, No. 1, Jan. 2004.
- [17] B.G. Shim, B.H Baek, S.Y. Park, "A Systematic Robot Fault-tolerance Approach," Proc. of 33rd IEEE COMPSAC, Vol.1, pp.624-625, 2009.
- [18] Orocos(Open Robot Control Software) Project Official Website, <http://www.orocos.org>.
- [19] OMG, Robotic Technology Component Specification Ver.1.0, April 2008.
- [20] OPRoS(Open Platform for Robotic Services) Official Website, <http://www.opros.or.kr>.
- [21] Wilfredo Torres-Pomales etc. "Software Fault Tolerance: A Tutorial," NASA, TM-2000-210616, p.6, 2000.
- [22] IEEE, "IEEE Standard Classification for SW Anomalies," IEEE Std 1044.1-1995, 1995.
- [23] IEEE, "IEEE Standard Classification for SW Anomalies," IEEE Std 1044-2009, 2009.
- [24] KITECH, "OPRoS Standard Component," pp.10-81, 2010.