

아키텍처 기반의 자가 성장 로봇을 위한 컴포넌트 선택 메커니즘

박유식⁰ 고인영
한국정보통신대학교
{yusikpark⁰, iko}@icu.ac.kr

A Component Selection Mechanism for Architecture Based Self-Growing Robot Software

Yu-Sik Park⁰, In-Young Ko
Information and Communications University

요 약

로봇 소프트웨어와 같이 사용자의 요구사항과 주변 환경의 잦은 변화에 직면하는 소프트웨어는 스스로 문제 상황을 판단하고, 동적으로 문제 상황을 극복할 수 있는 기능을 검색하고 이용할 수 있어야 한다. 이를 위해 본 논문에서는 아키텍처 기반의 로봇 소프트웨어 환경하에서 온톨로지를 이용하여 로봇의 문제 상황 극복을 위한 소프트웨어의 아키텍처를 재구성 전략과 컴포넌트를 모델링하고 이를 이용하여 문제 상황을 해결 할 수 있도록 적합한 컴포넌트를 선택하는 메커니즘을 기술한다.

1. 서 론

지능형 서비스 로봇이란 사용자 및 환경적인 상황들을 스스로 파악하고 문제 상황이 발생할 경우 이를 해결 함으로써 그에 맞는 유용한 서비스를 사용자에게 제공 로봇을 말한다. 지능형 서비스 로봇은 예기치 못한 환경이나 문제에 자주 직면할 수 있는 특성을 가지고 있는데 반면, 이러한 상황을 미리 예견하고 그에 필요한 기능을 로봇이 모두 갖추도록 하는 것은 불가능 할 것이다. 따라서 로봇이 내부의 기능만으로 이러한 문제를 해결할 수 없을 때 외부로부터 필요한 기능을 스스로 탐색, 획득하고 이용하여 문제를 해결할 수 있는 방법이 필요하다. 이러한 필요성에 의해 본 연구는 고수준의 소프트웨어 재구성 능력을 제공하는 자가성장 소프트웨어를 위한 프레임워크를 개발하는 것을 목적으로 실행되었다.¹ 이 프레임워크에서 중요한 요소는 문제상황을 해결하기 위해 적절한 기능을 검색하는 수단과 이러한 기능들을 획득, 저장, 관리하는 수단이다. 기능들의 획득, 저장 관리를 위한 저장소는 개발되었으며[1,3] 본 논문에서는 문제 해결을 위한 기능 검색 및 추론 방법에 대해 중점을 두고 설명한다.

2 절에서는 자가 성장 소프트웨어 프레임워크를 위해 필요한 요소들을 간략히 설명하고, 3 절에서는 이 프레임워크의 중요 요소인 온톨로지 기반의 컴포넌트 명세 모델과 아키텍처 브로커 및 컴포넌트 브로커의 역할에 대해 기술한다. 4절에서는 결론과 향후 연구에 대해 설명 한다.

2. 자가 성장 소프트웨어 프레임워크

사용자의 요구나 시스템의 환경이 계속적으로 변하는 상황에서 로봇 소프트웨어는 최적의 질을 가지는 서비스를 지속적으로 제공해야 한다. 컴포넌트 개념이 이와 같은 소프트웨어의 복잡성을 다루는 데 적합한 해결책으로 주로 고려되고 있다[2]. 컴포넌트 기반의 소프트웨어 환경에서는 필요에 따라 컴포넌트를 추가하거나 빼면서 소프트웨어를 재구성 하는 것이 용이하다. 그러나 동적인 상황에서 이와 같은 개념을 사용하기 위해서는 컴포넌트들의 관계를 명시하고 다루기 위한 소프트웨어 아키텍처가 필요하고 컴포넌트의 추가나 제거를 명시하는 재구성 절차를 설명하는 방법이 필요하다. 또한 문제 해결을 위한 적합한 컴포넌트를 찾는 컴포넌트 선택 메커니즘 역시 중요한 요소이다. 그림 1은 전반적인 자가 성장 소프트웨어의 구조[3]를 보여주며 본 논문에서는 온톨로지를 이용한 컴포넌트 명세 모델과 이를 이용한 아키텍처/컴포넌트 브로커에 초점을 둔다.

2.1 온톨로지 기반의 컴포넌트 명세 모델

자가성장 성장 소프트웨어를 구현하는데 중요한 문제 중 하나는 예기치 못한 상황을 해결하기 위하여 로봇이 적절한 컴포넌트를 찾아내는 것이다.

* 이 연구(논문)는 산업자원부 지원으로 수행하는 21세기 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술개발사업)의 일환으로 수행되었습니다.

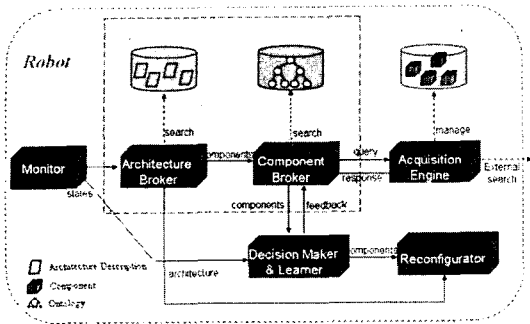


그림 1. 자가 성장 소프트웨어의 구조 [3]

다양한 연구들이 컴포넌트 선택 기법들을 제공하기 위하여 실행되었다. 이러한 연구들은 주로 세 가지의 타입으로 나뉘어진다: keyword-based, faceted based, semantic-net based[4]. 우리는 좀더 풍부한 시맨틱스(semantic)를 반영할 수 있는 semantic-net 방법을 사용하였다. 이 방법을 이용해 로봇이 처할 수 있는 상황과 아키텍처 재구성 그리고 컴포넌트의 시맨틱스를 묘사하기 위한 온톨로지를 개발하였다. 온톨로지는 관심 영역의 지식을 형식적인 방법으로 명확하게 표현하고 공유할 수 있도록 해준다[5]. 따라서 온톨로지는 적합한 아키텍처와 컴포넌트를 획득할 수 있는 쿼리를 생성하는데 필요한 용어를 제공한다. 온톨로지 기반의 컴포넌트 명세 모델은 그림 2와 같이 다양한 속성(property)을 가지며 이를 기반으로 로봇이 처한 문제 상황을 이해하고 극복하기 위한 아키텍처와 컴포넌트를 찾기 위해 적합성을 평가하고 비교한다.

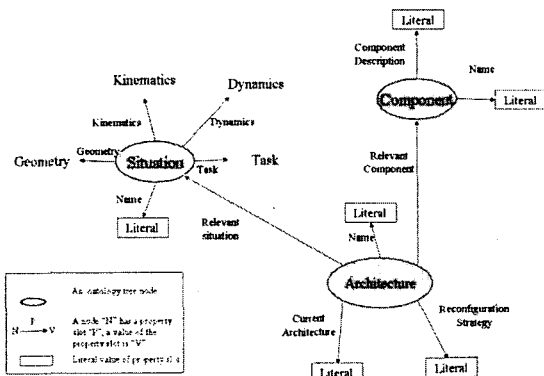


그림 2. 온톨로지 기반의 명세 모델

온톨로지 기반의 모델은 로봇 센서로부터 받은 상태 정보들을 로봇이 처한 상황의 시맨틱스와 연계 시킴으로 로봇이 처한 상황을 묘사한다[6]. 상황 정보들의 집합이 'Situation' 온톨로지와 비교되면 아키텍처 브로커는 상황 정보의 집합과 'Situation' 온톨로지의 속성들과 유사성을 평가한다. 'Architecture'라 불리는 또 다른 온톨로지는 이전 단계에서 탐지된 상황을 극복하기 위한

아키텍처 재구성 전략을 정의한다. 아키텍처 재구성 전략은 특정 기능을 제공하는 아키텍처 스타일을 명시한다. 즉, 로봇이 어떤 상황을 해결하기 위해 요구되는 특정 컴포넌트들의 조합을 명시하는 것이 아니라 로봇이 처한 문제 상황을 해결하기 위해 요구되는 기능들의 조합을 명시한다. 마지막 온톨로지는 'Component' 온톨로지이다. 이 온톨로지는 컴포넌트의 기능을 기반으로 구성되어 있으며 좀더 자세한 컴포넌트의 특성(비 기능적 특성 등)을 명시하기 위해 속성으로 컴포넌트 명세를 가진다. 아키텍처 재구성 전략을 실행하기 위해 컴포넌트 브로커는 'Component' 온톨로지에 접근하여 아키텍처 재구성 명세에 기술된 필요한 기능을 제공하는 컴포넌트를 찾게 된다. 이 온톨로지는 로봇이 사용 가능한 소프트웨어 컴포넌트의 기능적 시맨틱스를 정의한다.

2.2 아키텍처 브로커

위에서 언급한 예기치 못한 상황과 사용자의 요구사항의 변경으로 인한 문제를 해결하기 위해 필요한 기능을 제공하는 컴포넌트를 추가 또는 교체를 할 필요가 있다. 소프트웨어 아키텍처를 변경함에 있어 새로운 아키텍처에 의해 추가되는 컴포넌트가 요구되는 기능을 제공하는지를 알아야 한다. 아키텍처 브로커의 역할은 요구되는 새로운 기능을 제공하기 위하여 현재의 소프트웨어 구성을 변경하는 방법을 명시하는 아키텍처 재구성 명세를 찾는 것이다.

이러한 로봇의 재구성은 두 단계로 이루어져 있다. 아키텍처브로커는 추상적 레벨의 재구성을 담당하는 반면 재구성기는 선택된 컴포넌트를 사용하여 실제 로봇 소프트웨어의 재구성을 담당한다. 앞에서 언급한 바와 같이 재구성 전략은 요구되는 컴포넌트의 기능을 명시한다. 특정 컴포넌트를 명시하는 것이 아니라 요구되는 컴포넌트의 기능을 명시함으로써 요구되는 기능을 제공하는 다양한 컴포넌트를 모두 후보로 평가할 수 있다. 이와 같이 아키텍처 재구성을 두 단계로 나누는 이유는 가능한 모든 컴포넌트를 평가하고 사용하기 위함이다.

아키텍처 재구성을 위해서는 일련의 아키텍처 재구성 명령과 아키텍처의 구성을 묘사하기 위한 언어가 필요하다. 재구성 전략은 그림 3과 같이 XML 기반의 언어로 기술되었다. 아키텍처 재구성 명세는 변경되어야 할 추상적 컴포넌트의 이름과 추가되거나 교체되어야 할 새로운 컴포넌트의 기능을 묘사한다. 이 재구성 명세는 세 개의 기본 동작을 사용하여 묘사된다.

- ADD: 현재 아키텍처에 추상적 컴포넌트를 추가하는 동작으로 추가되는 컴포넌트의 기능을 표현하는 온톨로지가 명시되어야 한다.
- Remove: 아키텍처로부터 추상적 컴포넌트를 제거한다.
- Replace: 아키텍처 내의 기존의 추상적 컴포넌트를 유지하고 실제 컴포넌트를 교체한다. 이 작동은 교체되는 컴포넌트에 의해 제공되는 기능을 명시하는 온톨로지가 필요하다.

```
<?xml version="1.0" ?>
<reconfiguration description name="http://semantics.icu.ac.kr/reconf#ToVisionbasedLocalization"
  <description>
    Change the current robot architecture into vision-based Localizer
  </description>
  </description>
  <profile>
    <required slotName="http://semantics.icu.ac.kr/service#Localizer"
      <action="http://semantics.icu.ac.kr/action#Replace"/>
    </required slotName="http://semantics.icu.ac.kr/service#MapBuilder"
      <action="http://semantics.icu.ac.kr/action#Remove"/>
    </profile>
  <configuration>
    <script>
      <replace slotName="http://semantics.icu.ac.kr/service#Localizer"
        <services>
          <service name="http://semantics.icu.ac.kr/service#VisionbasedLocalization"/>
          <service name="http://semantics.icu.ac.kr/service#VisionbasedMapBuilding"/>
        </services>
      </replace>
      <remove slotName="http://semantics.icu.ac.kr/service#MapBuilder"/>
    </script>
  </configuration>
</reconfigurationdescription>
```

그림 3. XML 기반의 재구성 명세의 예

2.3 컴포넌트 브로커

컴포넌트 브로커는 문제 상황을 해결하기 위한 적합한 소프트웨어 컴포넌트를 검색하고 추론한다. 아키텍처 재구성명세를 기반으로 선택된 소프트웨어 아키텍처를 실행하기 위한 적절한 컴포넌트를 선택하는 것이 컴포넌트 브로커의 역할이다. 컴포넌트 브로커는 component 온톨로지를 접합하여 아키텍처 재구성명세에 기술된 기능을 제공하는 컴포넌트를 검색하게 된다. 온톨로지는 계층적 구조를 가지고 있으며 상위 개념은 하위 개념을 포함한다. 따라서 컴포넌트 중 문제 해결을 위한 가장 적합한 컴포넌트를 찾기 위한 가능성을 높이기 위해 relaxation 방법이 적용되었다[6]. 온톨로지는 계층적 구조를 가지기 때문에 더 많은 결과를 얻기 위해 쿼리 트리의 edge와 node를 완화하는 structure relaxation 방법이 사용되었다. 이러한 relaxation 방법에는 edge relaxation, node relaxation, order relaxation이 있다[7]. 이러한 방법 중 쿼리에 대한 제약성을 완화하기 위해 edge relaxation 방법이 사용되었다.

2.4 아키텍처/컴포넌트 선택 브로커링 과정

전반적인 브로커링 과정은 세 단계로 구성된다: 1) 상황 매칭 단계, 2) 아키텍처 재구성 선택 단계, 3) 컴포넌트 선택 단계. 모니터가 탐지한 상황정보를 바탕으로 적합한 situation 온톨로지를 식별하면 아키텍처 브로커는 situation 온톨로지와 연계된 아키텍처 재구성 전략을 선택한다. 선택된 재구성 전략의 relaxation 후에 아키텍처 브로커는 현재 소프트웨어 아키텍처에 적용할 수 있는 적합한 재구성 전략을 선택하게 된다. 아키텍처 브로커는 소프트웨어 아키텍처를 재구성하기 위해 필요한 컴포넌트들을 컴포넌트 브로커에 요청하게 되고 이때 컴포넌트 브로커는 아키텍처 재구성 명세에 기술된 기능을 제공하기 위한 후보 컴포넌트를 검색한다. 이때에도 역시 relaxation 방법이 적용된다. 아키텍처 브로커 및 컴포넌트 브로커의 브로커링 과정은 그림 3과 같이 이루어지며 다음과 같이 요약된다.

- i. 모니터로부터 받은 상황 정보를 바탕으로 situation 온톨로지서 상황 인스턴스를 추론한다.
- ii. 아키텍처 브로커는 상황 인스턴스와 연계된 아키텍처

- iii. 아키텍처 브로커는 선택된 아키텍처 재구성 명세로부터 재구성 절차를 추출한다.
- iv. 아키텍처 브로커는 아키텍처 재구성 전략에 포함된 컴포넌트의 기능을 이해하고 컴포넌트 브로커에 컴포넌트 검색을 요청한다.
- v. 컴포넌트 브로커는 component 온톨로지를 사용하여 표현된 기능들을 기반으로 적합한 컴포넌트를 추론한다.

3. 결론 및 향후 연구

본 논문에서는 온톨로지를 이용하여 로봇이 처한 문제 상황과 이를 극복하기 위한 아키텍처 재구성 전략과 아키텍처 재구성에 필요한 컴포넌트를 모델링하고 이를 선택하기 위한 방법을 설명하였다.

앞으로 로봇이 더 많은 기능을 지원하게 됨에 따라 이를 위한 소프트웨어 컴포넌트가 증가하게 될 것이다. 이때 발생할 수 있는 컴포넌트 선택의 복잡성을 고려하여 효율적인 컴포넌트 선택을 위해 현재 feature model을 사용하여 컴포넌트 사이의 관계를 명시하여 ontology를 확장하기 위한 연구가 진행 중이다.

참고문헌

- [1] Hyung-Min Koo and In-Young Ko, " A Repository Framework for Self-Growing Robot Software " , Proceedings of 12th Asia-Pacific Software Engineering Conference (APSEC' 2005), Taiwan, December 2005.
- [2] Narsis, Y. Amirat, Y. Djouani, K, " Dynamic reconfiguration service for component based software architectures " , The 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS), Singapore.
- [3] Hyung-Min Koo and et al. " A Repository Framework for Architecture Based Self-Grwoing Robot Software " , Proceedings of 2006 Korea Conference on Software Engineering
- [4] Eduardo Ostertag and James Hendler and Ruben Prieto Diaz and Christine Braun, " Computing similarity in a reuse library system: An AI-based approach " , in ACM transaction on Software Engineering and Methodology, 1992
- [5] N. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05 and SMI-2001-0880, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, March 2001.
- [6] Hwayoun Lee, Ho-Jin Choi, and In-Young Ko, " A Semantically-based Component Selection Mechanism for Intelligent Service Robot " , MICAL 2005, pp 1042-1051
- [7] S. Amer-Yahia, S. Cho, and D. Srivastava, " Tree pattern relaxation " , in Conference on Extending Database Technology, 2002.