

# 자가 성장 로봇 소프트웨어를 위한 컴포넌트 저장소 구조

구형민<sup>0</sup>, 고인영  
 한국정보통신대학교  
 {hyungminkoo<sup>0</sup>, iko}@icu.ac.kr

## A Repository Framework for Self-Growing Robot Software

Hyung-Min Koo<sup>0</sup>, In-Young Ko  
 Information and Communications University (ICU)

### 요 약

자가성장 소프트웨어란 동적인 상황 판단에 의거하여 스스로 자신의 기능과 컴포넌트 구성 능력을 성장시키는 소프트웨어를 말한다. 자가성장 소프트웨어는 자신의 환경을 모니터링 하여 사용자에게 적절한 행동을 제공하여 주는 지능형 로봇에 특히 필요하다. 지능형 로봇은 현재 자신이 가지고 있는 소프트웨어로는 해결할 수 없는 상황에 부딪힐 수 있고 이러한 경우 동적으로 소프트웨어 컴포넌트를 획득하여 이용하게 해주는 것이 자가성장 로봇 소프트웨어의 역할이다. 효율적인 컴포넌트의 획득과 관리를 위해 컴포넌트 저장소의 중요성이 부각되고, 자가성장 로봇 소프트웨어를 실현하기 위한 저장소 프레임워크를 ICU의 SemBots 프로젝트에서 개발 중에 있다. 본 논문에서는 저장소를 위한 요구조건과 구조를 기술하고 저장소 시스템을 위한 프로토타입을 제시한다.

### 1. 서 론

컴퓨팅 환경의 복잡성과 역동성이 강조됨에 따라 사람들은 환경과 요구의 변화에 따라 자신의 행동을 동적으로 변화시키는 자가 적응 소프트웨어를 추구하게 되었다 [1]. 자가 성장 소프트웨어는 자가 적응 소프트웨어의 한 종류이고 자가 적응이란 사람의 간섭 없이 스스로 진화하고 확장해 나가는 행위를 말한다. 유사한 개념이 " Vision of Autonomic Computing" 에서 거론 되었고 자가 구성, 자가 최적화, 자가 치유, 자가 보호가 그 필수 요소로 지적되었다[2]. 여기서 자가 최적화란 " 수행이나 비용 면에서 자신을 보다 효율적으로 만들기 위해 자신의 동작을 개선하고, 자가 최적화의 기회를 확인하고 포착하는 방법을 꾸준히 탐색하는 자동화된 시스템" 이라고 정의하고 있다. 따라서, 자가 성장 소프트웨어는 자가 최적화와 관계가 있다.

자가 성장 소프트웨어는 주변 상황을 판단하고 인간에게 알맞은 행동을 해 주는 지능형 로봇에 특히 필요한 요소이다. 이 로봇은 종종 현재의 소프트웨어로는 해결할 수 없는 문제들에 직면할 수 있다. 자가 성장 소프트웨어를 실현하기 위해서는 효율적인 저장소 시스템이 필요한데 그 이유는 저장소가 성장을 위한 중요한 정보를 효과적으로 검색하여 획득하고, 저장하는 기능을 제공하기 때문이다.

본 논문에서는 지능형 서비스 로봇의 자가 성장을 위한 저장소 구조를 기술한다. 2절에서는 자가성장과 자가성장 소프트웨어가 무엇인지, 전형적인 소프트웨어에 비해 어떤 장점이 있는지, 그리고 자가성장을 위한 단계를 설명한다. 3절에서는 현재 ICU에서 진행중인 SemBots 과제에 대해 간략히 설명하고 4절에서는 자가 성장을 위한 저장소 구조에는 어떤 요구조건들이 필요한지 설명한다. 5절과 6절에서는 저장소 구조와 그 프로토타입을, 7절에서는 관련 연구를, 마지막으로 결론과 향후 연구 방향에 대해 설명한다.

### 2. 자가 성장 소프트웨어

자가 성장 소프트웨어란 동적인 상황 판단에 의거해 스스로 자신의 기능과 컴포넌트의 구성 능력을 성장시키는 소프트웨어를 말한다. 자가 성장 소프트웨어는 자신이 현재 가진 소프트웨어의 기능으로는 해결할 수 없는 상황에 봉착 했을 때, 자신의 경험을 토대로 외부의 소프트웨어 요소를 찾는다. 이 소프트웨어 요소는

객체, 컴포넌트, 서비스 등이 될 수 있다. 자가 성장 소프트웨어를 이용하여 소프트웨어 시스템, 특히 동적으로 변하는 환경에서 작동해야 하는 소프트웨어 시스템의 효율성을 극대화시킬 수 있다. 2.1 절에서 이 효율성에 대한 설명을 한다.

#### 2.1 자가 성장 소프트웨어와 전형적인 소프트웨어

전형적인 소프트웨어 시스템에서 새로운 기능을 부가 시키기 위해서는 시스템을 정지시킨 후 새로운 기능의 컴포넌트들을 소프트웨어에 수동으로 재구성 시켜야 한다. 즉 사용자 요구의 사항의 변화나 새로운 컴포넌트들은 개발시간 중에 주로 소프트웨어 시스템에 반영되게 된다.

그러나 자가 성장 소프트웨어 시스템에서 소프트웨어는 실행시간 동안 성장 할 수 있다. 컴포넌트 개발자는 컴포넌트들을 개발한 뒤 외부 환경의 저장소들에 이를 등록시켜 놓으면 되고 자가 성장 소프트웨어가 문제가 검출되거나 사용자의 요구사항이 변화할 때 이것을 동적으로 획득하여 성장하는 것이다.

#### 2.2 자가 성장을 위한 과정

자가 성장을 위해서는 세 단계의 과정이 필요하다. 수집, 학습, 증식이 그것이다. 수집은 성장이 필요할 시, 외부의 저장소에서 컴포넌트 들을 모으는 것을 말하고 학습은 수집된 컴포넌트들 중에서 자신의 경험을 토대로 하여 상황 해결을 위해 가장 적절한 컴포넌트들을 선택하는 것을 말한다. 마지막으로 증식은 선택된 컴포넌트들을 내부 저장소에 저장하는 것을 말한다. 이러한 과정이 지속적으로 반복되면서 소프트웨어는 점차 성장하게 된다.

### 3. SemBots 프로젝트

현재 ICU의 SemBots 프로젝트에서는 지능형 서비스 로봇이 예상치 못한 상황에 처했을 때, 로봇 자신의 경험과 사용자의 피드백을 기반으로 고수준 소프트웨어의 구성을 개선하여 상황에 대처할 수 있게 해주기 위한 자가 성장 소프트웨어 기술 개발을 목표로 연구를 진행 중에 있다. 이를 위해 컴포넌트 브로커가 개발 중에 있는데, 컴포넌트 브로커는 로봇의 환경정보를 통해 로봇이 처한 상황을 인식하고 이를 바탕으로 문제 해결을 위한 가장 적절하고 효율적인 컴포넌트를 찾을 수 있도록 해준다. 이를 위해 온톨로지 기반의

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

시멘틱스 표현 모델과 시멘틱 측정 메커니즘이 개발중에 있다.

SemBots 프로젝트에서 개발중인 자가 성장 소프트웨어는 문제 해결 능력의 증진과 고 수준의 소프트웨어 구성 메커니즘의 지원을 통해 예상치 못한 상황을 다룰 수 있게 된다. 환경이 변화하거나 사용자의 요구가 변할 때 혹은 예외상황이 발생했을 때, 로봇 소프트웨어는 외부 컴포넌트 저장소에서 컴포넌트를 수집하고 증식함으로써 스스로 성장한다.

#### 4. 자가성장 소프트웨어를 위한 저장소의 요구조건

저장소에 대한 요구 조건[3]과 주 성공 요인들[4]은 관련연구들에 찾아볼 수 있다. 따라서 이 절에서는 특히 자가 성장을 제공하기 위한 저장소의 요구조건들을 기술한다.

- **정확성** : 자가 성장을 제공하기 위해서 저장소는 정확한 컴포넌트 정보를 담고 있어야 하고, 정확한 컴포넌트 정보를 찾을 수 있게 해주는 메커니즘을 제공해야 한다.
- **확장성** : 현재, 네트워크 기술의 발달로 많은 외부의 네트워크 자원들을 사용 가능하고, 이러한 자원들을 이용함으로써 시스템 내부의 용량 한계 문제와 컴포넌트 정보 관리의 부하를 해결할 수 있다. 이를 위해 외부의 저장소들은 쉽게 확장 될 수 있게 구성되어야 한다.
- **역동성** : 자가 성장 소프트웨어는 실행시간 동안에 문제를 해결해야 하기 때문에 동적인 수집과 성장, 저장을 지원해야 한다. 이러한 동적인 행위들은 로봇의 작업 수행능력과 밀접한 관계가 있으므로 빠르고 효율적인 동적 성장을 필요로 한다.
- **교환성** : 내부 시스템 저장소의 용량은 한계를 가지고 있고, 이 한계에 다다랐을 때 어떻게 새로운 컴포넌트를 추가하고 기존의 컴포넌트를 삭제할 것인가에 대한 계획이 필요하다. 특정한 기준을 두어 어떤 기존의 컴포넌트를 삭제 시켜야 하는지에 대한 계획이 필요하다.
- **공유성** : 외부 저장소들은 모든 시스템에 대해 공유가 가능해야 한다. 이는 컴포넌트의 재사용성과도 밀접한 관계가 있고, 새로운 소프트웨어 시스템이 환경에 추가 되었을 때 이 시스템은 외부 저장소의 컴포넌트를 자유롭게 이용할 수 있어야 한다.

#### 5. 저장소 구조

이 절에서는 위의 요구조건을 만족시키기 위한 우리의 접근방법과 저장소 구조를 설명한다.

##### 5.1 접근방법

- **온톨로지 저장소** : 정확한 컴포넌트를 찾기 위해서 우리는 시멘틱 정보를 이용하고 있고, 구체적으로 온톨로지 기반의 접근법을 사용하고 있다. 온톨로지 기반의 저장소는 일종의 데이터베이스이고, 온톨로지간의 관계를 그래프 형태의 정보로 저장하고 있다. 적절한 컴포넌트를 찾기 위해 외부 온톨로지를 먼저 가져와서 판단하는 방법은 실제 물리적인 컴포넌트 파일들을 가져와 판단하는 것보다 그 성능을 높일 수 있다.
- **컴포넌트 저장소** : 온톨로지 저장소의 컴포넌트 온톨로지는 물리적인 실제 컴포넌트 파일을 가리키고 있고, 컴포넌트 저장소에는 컴포넌트들의 URI (Uniform Resource Identifier)와 파일 시스템에서의 위치, 실제 물리적인 컴포넌트 파일을 저장하고 있다. 온톨로지를 통해 가장 적절한다고 판단된 컴포넌트는 그 위치 정보 쿼리를 이용해 획득할 수 있다.
- **컴포넌트 획득 엔진** : 컴포넌트 획득 엔진은 외부의 온톨로지 저장소들과 컴포넌트 저장소들로부터 온톨로지 부분 구조와 컴포넌트들을 획득해 오는 역할을 한다. 이를 위해 컴포넌트 획득 엔진은 온톨로지 수집 메커니즘과 파일 전송 메커니즘을 제공한다. 이 엔진은 외부 온톨로지 저장소에서 온톨로지들을 수집하고 성장의 한 과정인 학습을 거쳐 최종 선택된 컴포넌트를 획득하여 내부 컴포넌트 저장소에 저장한다. 본 논문은 저장소에 초점을 맞추고 있고, 학습에 관한 내용은 다루지 않는다.
- **컴포넌트 퇴거 계획** : 자가 성장은 내부 용량의 한계성 때문에 무한대로 진행될 수가 없다. 따라서 내부 컴포넌트를 퇴거하고 새로운 컴포넌트를 대체할 수 있는 특별한 계획이 필요하다. 우리의

저장소 구조에서는 사용빈도와 저장된 날짜를 내부 컴포넌트 저장소에 함께 저장해 놓음으로써, 로봇이 내부 저장소 용량의 한계에 부딪히게 되면 먼저 가장 사용되지 않은 컴포넌트를 퇴거하게 된다. 이러한 메커니즘으로 컴포넌트를 교체할 수 없다면, 로봇에 필수적인 컴포넌트를 제외한 가장 오래된 컴포넌트를 퇴거 시킨다.

- **동적 성장 메커니즘** : 모든 성장을 위한 행위는 모두 로봇의 실행 시간 동안 이루어진다. 따라서 외부 온톨로지와 외부 컴포넌트의 동적인 수집, 그리고 그것들의 동적인 업데이트가 필수적이다. 컴포넌트 획득 엔진은 이러한 일들을 동적으로 수행하고 있고 우리는 수행능력 즉 획득 시간 측면을 개선하기 위해 노력 중에 있다.

##### 5.2 저장소 구조의 소개

그림 1은 저장소의 구조를 보여준다. 예상치 못한 상황이 발생했을 때, 컴포넌트 브로커는 내부 온톨로지 저장소를 검색하여 적절한 컴포넌트 들을 찾고, 만약 내부 저장소에 없다면 컴포넌트 획득 엔진에게 외부 컴포넌트를 요청한다. 컴포넌트 획득 엔진은 외부 온톨로지 저장소에서 적절한 온톨로지들을 획득하여 내부 온톨로지 저장소를 업데이트 하고, 시멘틱스 측정과 학습을 통해 적절성을 판단하게 된다. 최종 선택된 컴포넌트를 외부 컴포넌트 저장소에서 실제 컴포넌트 파일을 획득하여 내부 컴포넌트 저장소에 저장한다.

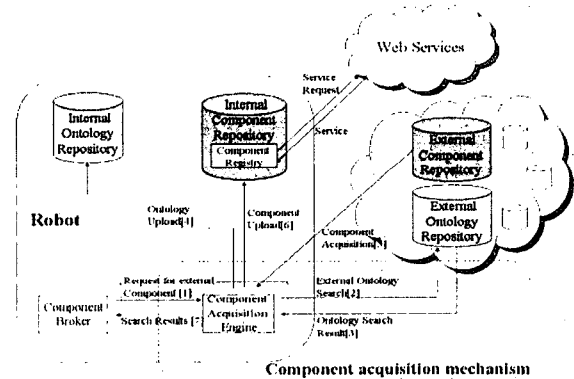


그림 1. 자가 성장 로봇 소프트웨어의 저장소 구조

##### 5.3 내부 저장소의 접근

문제상황 해결을 위해 컴포넌트 브로커는 먼저 내부 저장소를 접근하게 된다. 내부 온톨로지 저장소 검색을 통하여 컴포넌트 브로커는 문제해결 기능을 제공하는 컴포넌트를 찾기 위해 시멘틱 기반의 상호 운용성을 측정 한다[10]. 이렇게 컴포넌트 브로커는 적절한 컴포넌트들의 후보를 생성하여 학습 부분으로 전송하고 학습파트에서 경험을 바탕으로 한 적절한 컴포넌트의 선택을 통해 최종 컴포넌트가 선택된다.

##### 5.4 외부 저장소의 접근

그림 1에서 볼 수 있듯이, 내부 저장소에서 적절한 컴포넌트를 찾을 수 없다면 컴포넌트 브로커는 컴포넌트 획득 엔진에게 외부 컴포넌트의 획득을 요청한다. 컴포넌트 획득 엔진은 외부의 다양한 온톨로지 저장소들로부터 적절한 컴포넌트 온톨로지들을 먼저 획득하고 내부 온톨로지 저장소에 업로드 한다. 컴포넌트 브로커는 내부 저장소에 대한 접근에서처럼 상호 운용성을 측정하여 후보 컴포넌트들을 학습파트로 전송한다. 학습 파트에서 최종적으로 컴포넌트를 선택하면 컴포넌트 획득 엔진은 외부 컴포넌트 저장소에서 실제 컴포넌트 파일을 획득하여 내부 컴포넌트 저장소에 저장한다. 획득된 컴포넌트의 정보 또한 내부 컴포넌트 저장소에 함께 저장된다.

##### 5.5 웹서비스의 접근

로봇은 때때로 소프트웨어 컴포넌트만으로 해결할 수 없는 기능을 요구할 수도 있다. 예를 들어, 로봇의 행동이 날씨와 밀접하게 연관되어 행해져야 한다면, 그 일을 수행하기 위해 로봇은 웹을 통해

정보를 얻을 수 있을 것이다. 날씨 정보를 획득하기 위해 로봇은 레지스트리에 저장되어 있는 날씨 웹서비스에 접근 할 수 있고, 만약에 내부 레지스트리에 원하는 서비스가 없다면 로봇은 웹서비스 저장소(UDDI: Universal Description, Discovery and Integration)에 접근해서 서비스를 요청할 수 있다. 서비스를 제공받은 후 로봇은 내부 레지스트리에 서비스의 정보를 저장함으로써 사용 가능한 서비스를 성장시킬 수 있다.

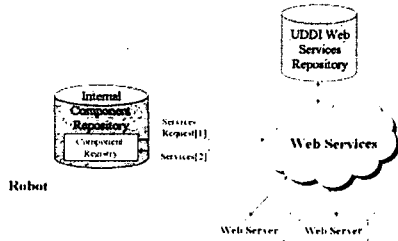


그림 2. 웹서비스의 접근

6. 프로토타입

이 절에서는 저장소 시스템의 프로토타입을 설명한다. 우리는 현재 자가 성장 소프트웨어를 위한 저장소 시스템을 윈도우 XP 운영체제 환경에서 구현하고 있고, 프로그래밍 언어는 JAVA를 사용하고 있다. 온톨로지 구축과 RDF/RDFS 파일 생성을 위해 Protégé 툴[11]을, 이 파일들로부터 온톨로지 저장소를 구축하기 위해 Jena2 라이브러리[12]를 사용하였다. Jena2는 온톨로지 쿼리 메커니즘을 제공하고, 온톨로지 구조를 데이터베이스 모델로 저장할 수 있게 해준다. 데이터베이스 생성과 관리를 위해 My-SQL을, JAVA에서 이 데이터베이스를 접근하기 위해 JDBC 드라이버를 사용하였다. 외부 저장소로부터의 온톨로지 획득을 위해 JAVA RMI를 이용하였고, 실제 컴포넌트를 획득하기 위해서는 FTP를 이용하였다. 초기 온톨로지 저장소를 구축하기 위해서는 Protégé로 온톨로지를 구축하여 RDF/RDFS 파일을 생성한 뒤 Jena2 라이브러리의 ModelRDB 클래스를 이용하여 My-SQL 서버에 데이터베이스 모델로 저장한다.

그림 3에서 구현된 저장소의 구조를 보여준다. RMI handler는 외부 저장소에서 온톨로지를 획득하기 위한 외부 메소드 호출 기능을 담당한다. 컴포넌트 획득 엔진은 찾고자 하는 컴포넌트의 스키마를 인수로 External Ontology Manager를 호출한다. External Ontology Manager는 로봇이 직접 데이터베이스에 접근하는 부하를 줄이기 위한 외부의 모델로써 외부 온톨로지 저장소에서 적절한 컴포넌트 온톨로지 후보들을 찾아서 리턴해 준다. 후보들을 찾을 때는 컴포넌트 브로커와 마찬가지로 시멘틱 측정 메커니즘을 통하여 한다. 학습파트에서 이 후보들 중 가장 적절한 컴포넌트를 선택하여 피드백을 넘겨주면 Ontology Updater는 컴포넌트의 명세를 내부 온톨로지 저장소의 구조에 업로드 한다. 최종적으로, Component Receiver는 FTP 접속을 통해 실제 컴포넌트 파일에 접근하여 내부 컴포넌트 저장소에 저장하고 컴포넌트 정보 또한 업데이트 한다. Retirement Planner는 컴포넌트 퇴거 계획을 수행하는 모델로써 내부 용량이 일정 한계점 이상에 다다랐을 때 호출 되고, 컴포넌트 저장소 데이터베이스에서 가장 사용빈도가 낮은 컴포넌트를 검색하여 삭제시킨다. 이 삭제에는 컴포넌트의 삭제 뿐 만 아니라 온톨로지, 컴포넌트 정보의 삭제 또한 포함된다.

7. 관련연구

현재 컴포넌트 저장소와 자가 성장을 위한 소프트웨어 개발을 위한 몇몇 연구들이 진행되거나 진행 중에 있다. 온톨로지 기반의 저장소를 위한 연구들이 진행 중에 있고, 정보의 메타데이터를 표현할 수 있게 해주는 RDF나 OWL과 같은 시멘틱 웹 언어는 온톨로지 와 온톨로지 저장소를 다룰 수 있게 해준다[5]. 컴포넌트 저장소에 대한 연구들도 있는데 범용 저장소[6], 상업적 저장소들[7], 지식 기반의 저장소[8], 또 재사용성을 강조한 저장소[9] 등이 있다. 이 연구들은 대부분 컴포넌트의 명세와 재사용성, 프리젠테이션,

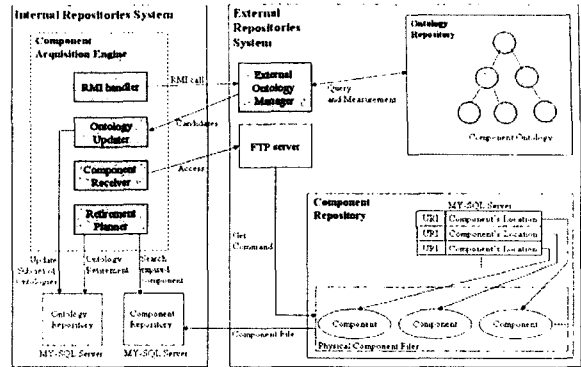


그림 3. 구현된 프로토타입

신뢰성에 초점을 맞추고 있고, 자가 성장의 개념을 다루고 있지는 않다. 자가 성장을 위한 소프트웨어는 IBM을 비롯한 몇몇 기관에서 연구를 진행 중에 있다[2].

8. 결론 및 향후 연구

지금까지 자가 성장 로봇 소프트웨어를 위한 컴포넌트 저장소 구조를 설명하였다. 자가 성장 소프트웨어는 자가 적응과 자동 컴퓨팅을 가능하게 해주고, 저장소 시스템은 자가 성장을 실현하기 위한 중요한 역할을 한다. 현재 ICU에서 개발중인 저장소 시스템은 온톨로지 저장소, 컴포넌트 저장소, 컴포넌트 획득 엔진, 컴포넌트 퇴거 계획, 동적 성장이라는 접근 방법을 이용하였다. 이러한 접근 방법은 정확성, 확장성, 역동성, 접근용이성, 교환성, 재사용성의 요구 조건을 만족시키고 있고, 현재 전체적인 저장소 시스템과 외부 컴포넌트 획득의 수행능력을 증진시키기 위해 노력하고 있다. 실제 로봇 환경에 적용하여 자가 성장 소프트웨어를 실험하고 있으며 로봇 소프트웨어의 구성과 재구성에 있어서 웹서비스를 활용하고자 하는 연구를 진행 중에 있다.

참고문헌

- [1] Peyman Oreizy and et al., "An Architecture-Based Approach to Self-Adaptive Software, IEEE Intelligent System, 1999.
- [2] Jeffrey O. Kephart and et al., "Vision of Autonomic Computing, IEEE Computer Society, 2003.
- [3] Michael Blaha, "Requirements for Repository Software, IEEE, 1998.
- [4] Kurt Schneider and Jan-Peter von Hunnius, "Effective Experience Repositories for Software Engineering", Proceedings of the 25th ICSE'03, IEEE, 2003.
- [5] Regina M. M. Braga et al., "The Use of Mediation and Ontology Technologies for Software Component Information Retrieval, ACM, 2001.
- [6] Sridhar Iyengar, "A Universal Repository Architecture using the OMG UML and MOF, IEEE, 1998.
- [7] Luqi and Jiang Guo, "A Survey of Software Reuse Repository", Research Associate US national Research Council.
- [8] Padmal Vitharana et al., "Knowledge-Based Repository Scheme for Storing and Retrieving Business Components: A Theoretical Design and an Empirical Analysis", IEEE transactions on Software Engineering, Vol. 29, NO. 7, July 2003.
- [9] Luqi and Jiang Guo, "A Survey of Software Reuse Repository, Research Associate US national Research Council.
- [10] In-Young Ko, Robert Neches, and Ke-Thia Yao, "A Semantic Model and Composition Mechanism for Active Document Collection Templates in Web-based Information Management Systems", Electronic Transactions on Artificial Intelligence (ETAI), Vol. 5, Section D, pp.55-77, 2001.
- [11] What is Protégé?, <http://protege.stanford.edu/overview/>
- [12] Jena Semantic Web Framework, <http://jena.sourceforge.net/>