
유아교육을 위한 콘텐츠 저작 도구

Contents Authoring Tool for Early Childhood Education

한선아
목원대학교

Sun-Ah Han(sahan@mokwon.ac.kr)

요약

본 논문은 시맨틱 웹 서비스 환경에서 유아교육을 위한 각각의 서비스를 표현하는 메타 정보를 기술한 서비스 템플릿 객체 모델을 기반으로 그래픽한 서비스 템플릿 저작도구 시스템을 구현하였다. 제안한 시스템은 자동으로 웹 서비스를 선택하고 구성하여 적절한 서비스 플랜을 생성하여 로봇 서비스를 제공한다. 또한 STDL을 기반으로 URC용 서비스 템플릿 생성, 추가, 삭제, 수정 등의 기능뿐만 아니라 서비스 템플릿 리소스들에 대한 그래픽한 처리 기능을 제공한다. 서비스 템플릿 과정에서 사용자 편의적인 환경 제공을 위해 플로우 뷰 스타일, 그리드 뷰 스타일, 텍스트 뷰 스타일 등의 편집 환경을 구현했다. 또한 실제 로봇 구동에 필요한 로봇API를 참조하고 이를 기반으로 추상 서비스 블록을 구체화하여 서비스 템플릿을 쉽게 작성할 수 있는 기능을 제공하게 된다. 따라서 서비스의 시맨틱 정보를 기반으로 서비스 에이전트의 지능적이고 자율적인 서비스를 제공할 수 있다.

■ 중심어 : | 유아교육 | 저작도구 | 콘텐츠저작 |

Abstract

This paper implements the graphic service template authoring tool based on the service template object model of describing the meta information of the services for early childhood education in the semantic web service environment. Our proposed system provides the robot services by constructing web services automatically and making the appropriate service plans. Moreover, it can create, append, delete, and update the service templates of URC based on STDL, and provide the graphic function on service template resources.

In order to provide the user friendly environment in the service template phase, we implement the various editing environment : flow view style, grid view style, and text view style. We also provide the easy editing function by realizing abstract service block based on the robot API. Finally we can offer the intelligent and autonomous service of service agent based on semantic information.

■ keyword : | Early Childhood Education | Authoring Tool | Contents Authoring |

I. 서론

현재, 멀지 않은 미래에 우리의 생활 속에 하나의 독

립적인 주체로 존재하게 될 로봇에 대한 수많은 연구가 이루어지고 있고, 이미 상용화 되어 다양한 분야에서 사용되고 있는 것이 현실이다. 기존의 IT 기술들의 집

합체로 불릴 수 있을 만큼, 다양한 기술들이 필요로 하는 로봇 기술은 기존의 산업용 로봇에서 점차 인간과 함께 하는 인간 친화적인 형태의 로봇에 대한 관심과 연구가 이루어지고 있다.

기존의 로봇에서 하나의 작업을 수행하기 위해 필요한 정보를 로봇에 내장된 서비스만을 이용하여 수행할 수 있는 형태인 수동적이고 고립된 형태의 제한적인 영역의 서비스 개념이 유비쿼터스 공간상에 존재하는 모든 활용 가능한 서비스를 포함하는 개념으로 확대되어야 할 필요가 있다. 이와 같이 이질적인 공간상에 존재하는 수많은 서비스들을 하나의 서비스로 구성하고 실행하기 위해서는 이종 플랫폼 상에 탑재되어 있는 서비스 객체들을 기능적으로 통합할 수 있어야 한다. 웹 서비스 기술은 단일 웹 서비스에 대한 정의와 비즈니스 레지스트리에 등록된 웹 서비스만을 이용할 수 있는 제한적인 구조이기 때문에 구문적인 확장은 가능하지만 의미적인 확장이 불가능한 한계를 내포하고 있다.

IT 신성장동력 기획보고서에 따르면 지능형 서비스 로봇을 Ubiquitous Robotic Companion(이하 URC)[8]으로 명명하고 비즈니스 모델에 입각한 산업 활성화와 기술개발을 추진하도록 되어 있다. 여기서, URC란 “언제 어디서나 나와 함께 하며 나에게 필요한 서비스를 제공하는 로봇”으로 정의되는 바, 기존의 로봇 개념에 네트워크를 부가한 URC 개념을 도입함으로써 다양한 고도의 기능이나 서비스 제공이 가능하고 Mobility와 Human Interface가 획기적으로 향상되어, 사용자 측면에서는 보다 저렴한 가격으로 다양한 서비스와 즐거움을 제공할 수 있는 가능성이 크게 확대될 것으로 기대된다. URC의 범위에는, 네트워크 인프라에 연결되어 있고 Intelligence를 갖추고 있어야 하되 Mobility 측면에서 기구적 이동(Hardware mobility)뿐 아니라 Software 전이(Software mobility)까지도 포함하는 것을 고려하고 있다.

본 논문은 시멘틱 웹 서비스 환경에서 각각의 서비스를 표현하는 메타 정보를 기술한 서비스 템플릿 객체 모델을 기반으로 자동으로 웹 서비스를 선택하고 구성하여 적절한 서비스 플랜을 생성하여 로봇 서비스를 제공하는 시스템을 구현하였다. STDL을 기반으로

URC용 서비스 템플릿 생성, 추가, 삭제, 수정 등의 기능뿐만 아니라 서비스 템플릿 리소스들에 대한 그래픽한 처리 기능을 제공한다. 서비스 템플릿 과정에서 사용자 편의적인 환경 제공을 위해 플로우 뷰 스타일, 그리드 뷰 스타일, 텍스트 뷰 스타일 등의 편집 환경을 구현했으며, 또한 실제 로봇 구동에 필요한 로봇 API를 참조하고 이를 기반으로 추상 서비스 블록을 구체화하여 서비스 템플릿을 쉽게 작성할 수 있는 기능을 제공하게 된다.

II. 관련연구

1. 기존의 시멘틱 웹 문서에 대한 국외의 그래픽한 저작도구는 OntoTrack[7]과 DUET[2]가 있다. OntoTrack[7]은 매우 큰 온톨로지를 마우스로 편집, 항해, 처리가 가능한 그래픽한 온톨로지 편집기 및 저작도구이다. 이는 대화적인 그래픽 기반 시각화 기능과 온톨로지 변화에 대한 즉시적인 추론 피드백 기능을 갖고 있다. 각 편집 단계는 클래스의 포섭(subsumption) 관계나 불충분 관계를 즉시 피드백하는 추론기와 동기화 된다. 이러한 피드백은 사용자가 의도하지 않은 모델링 에러를 쉽게 찾도록 도와준다. 최근 포섭관계에 대해 글로써 설명하는 기능을 OntoTrack에 확장했다.

DUET[2]은 AT&T에서 개발되었고, DAML을 위해 UML(Unified Modeling Language) 시각화와 저작 환경을 제공한다. 핵심 DAML 개념은 DAML을 위해 UML 프로파일을 통하여 UML에 매핑시킨다. 유효성 검증을 통해 유효한 UML 다이어그램을 DAML+OIL로 생성시켜준다. 현재 Rational Rose Addin과 ArgoUML 버전을 포함하고 있다. 특징은 클래스 다이어그램을 기반으로 UML의 이용가능한 정적 구성으로 표현한다. Rose 버전과 Argo 버전으로 분리하여 RDF/RDFS 엔티티와 DAML 엔티티 사이의 차별화를 시도했다.

그래픽한 저작도구가 아닌 국외의 저작도구에 대해서는 많은 연구가 이루어져왔다. 대표적인 저작도구는 Protégé-2000[3], OilEd[1], OntoEdit[6], MobiOWLS,

SMORE[4], 그리고 OntoMat이다. Protégé-2000[3]은 국립의료도서관, NSF, DARPA의 후원을 받아 스탠포드 의과대학의 의료정보학과에서 지식 기반의 구조를 작성하기 위한 도구로 15년간의 연구 기간을 거쳐 개발되었다. 맨체스터 대학에서 개발된 OiEd[1]은 DAML+OIL 기반으로 그래프식으로 온톨로지를 작성하는 저작도구이다. OiEd는 OIL을 기반으로 하며 프리웨어 편집기로 지원하는 목표로 개발되었기 때문에 전체적인 온톨로지 개발환경을 제공하지 않는다. OntoEdit[6]은 독일의 ontoprise® GmbH 사에서 개발하였으며, 그래픽을 이용하여 온톨로지의 개발과 유지보수를 제공하는 온톨로지 공학 환경(Ontology Engineering Environment)이다. OntoEdit은 강력한 내부 온톨로지 모델을 가지고 있다.

국내의 경우, '이지아울'(ezOWL)[5]은 복잡한 온톨로지를 그래픽 다이어그램으로 생성 및 편집할 수 있어, 온톨로지 형식에 대한 지식 없이도 그 내용을 쉽게 파악하고 저작할 수 있다. '이지아울'은 독자적인 S/W이지만 현재 가장 널리 보급되어 있는 Protégé-2000과 플러그인(Plug-in)하여 사용할 수도 있게 함으로써 기존의 온톨로지 개발자들이 쉽게 접할 수 있게 배려하였다. ezOWL은 W3C(국제 웹 표준화기구)에서 표준화 작업중인 차세대 웹 온톨로지 언어 OWL(Web Ontology Language)을 사용하고 있으며 기존 언어인 DAML+OIL, RDF/RDFS도 지원하고 있어 호환성도 뛰어나다.

III. 서비스 템플릿 기술언어(STDL)

1. STDL의 기본 개념

STDL[8-12]은 서비스 컴포지션을 위한 서비스 템플릿을 기술하기 위한 언어로서 OWL을 기반으로 한 온톨로지로서 정의된다. 온톨로지는 "공유된 개념화(shared conceptualization)에 대한 정형화되고 명시적인 명세(formal and explicit specification)"라고 정의할 수 있다. 온톨로지는 간단히 표현하면 단어와 관계들로 구성된 사전으로서 어느 특정 도메인에 관련된 단어들

을 계층적 구조로 표현하고 추가적으로 이를 확장할 수 있는 추론규칙을 포함한다. 온톨로지의 역할 중 하나는 서로 다른 데이터베이스가 같은 개념에 대해서 서로 다른 단어나 식별자를 사용할 경우에 이를 해결해 주는데 있다. 온톨로지는 웹 기반의 지식 처리나 응용 프로그램 사이의 지식 공유, 재사용들을 가능하게 하는 아주 중요한 요소로 자리잡고 있다. 온톨로지로서 정의되는 STDL을 구성하는 핵심 개념들은 다음과 같다.

1) 서비스 템플릿 레지스트리 및 리파지토리

서비스 템플릿 레지스트리는 서비스 템플릿에 대한 의미를 기반으로 하는 색인 서비스를 제공한다. 이러한 서비스 템플릿의 의미에는 Input, Output, Precondition, Effect 및 Category 정보가 있으며 각각의 정보를 표현하는 값들은 개념 온톨로지 및 서비스 온톨로지로부터 참조하여 사용한다. 서비스 템플릿 리파지토리는 서비스 템플릿을 위한 저장 공간을 제공한다. 실제로 템플릿 레지스트리와 리파지토리는 하나의 서버 시스템 혹은 별도의 서버 시스템에 탑재 가능하다.

2) 서비스 템플릿

서비스 템플릿은 서비스 컴포지션을 위한 재활용 가능한 Building Block으로서 각각의 서비스를 표현하는 Meta 의미를 기술한 객체이다. 서비스 템플릿은 Basic 타입과 Aggregate 타입으로 구분된다. Basic 템플릿은 최소 단위의 서비스 의미를 기술하며 Aggregate 템플릿은 Basic 템플릿 또는 또 다른 Aggregate 템플릿의 집합으로 표현된다. [그림 1]은 서비스 템플릿의 구성도를 보여주고 있다.

3) 서비스 템플릿 인스턴스

서비스 템플릿 인스턴스는 메타 의미를 기술하는 서비스 템플릿에 대해 실존하는 구체적인 서비스 객체를 바인딩하기 위한 정보를 기술한다.

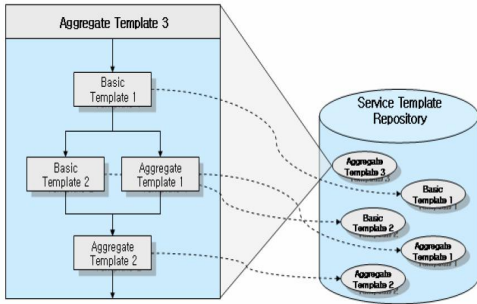


그림 1. 서비스 템플릿 구성

2. STDL의 기반의 서비스 컴포지션

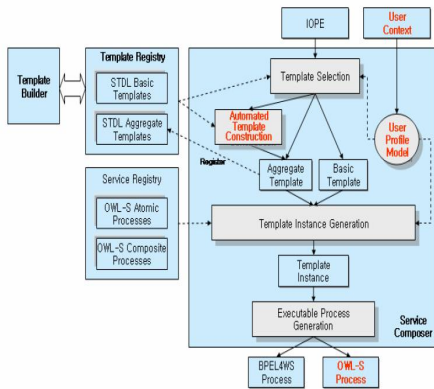


그림 2. STDL 기반 URC 서비스 컴포지션

[그림 2]은 STDL을 기반으로 하는 URC 서비스 컴포지션의 처리 절차를 보여주며, 각각의 구성 요소에 대해 설명하기로 한다.

URC용 지능형 웹 기반 서비스 템플릿 서비스 컴포지션 기술은 사용자의 요구를 만족하는 최적의 해를 웹 서비스를 통해 검색하고 서비스 프로세스를 구성하는 기술로, 기존의 웹 서비스들과 다른 컴포넌트들을 조합하고 연결하기 위해 필요한 액티비티들로 구성된 일련의 프로세스이다. 서비스 플랜을 실행 가능한 구체적인 플랜으로 생성하는 과정에서 사용자의 명령 정보를 정형화한 IOPE(Input, Output, Precondition, Effect)정보를 획득하고, 적절한 서비스 템플릿을 기반으로 최적의 URC 서비스들을 탐색, 선택 및 조합하여 서비스 플랜 인스턴스를 생성하게 된다. 이 과정에서 필요한 서비스

템플릿을 사용자 편의적인 환경에서 저작 가능하도록 하는 도구가 요구되며, 이것은 다양한 서비스 도메인 환경을 지원할 수 있어야 한다.

1) 서비스 템플릿 저작

우선 서비스 컴포지션을 위해서는 적절한 템플릿이 필요하다. 서비스 개발자는 이러한 템플릿을 저작하기 위한 WYSIWYG 기반의 템플릿 저작 도구(Template Builder)를 기반으로 다양한 Basic 및 Aggregate 서비스 템플릿을 저작하여 템플릿 레지스트리에 등록하여야 한다. 템플릿 저작 도구는 기본적으로 STDL 직렬화(Serialization) 기능을 제공하여야 한다.

2) 서비스 템플릿 선택

서비스 컴포지션은 컨텍스트 처리 모듈로부터 입력 받은 IOPE 정보의 의미를 기반으로 템플릿 레지스트리로부터 서비스 템플릿을 검색한다. 검색된 서비스 템플릿을 Deterministic하게 결정할 수 없는 경우 사용자 컨텍스트(User Context)로부터 사용자의 프로파일 정보(신상정보, 선호도 등)를 획득하여 적절한 서비스 템플릿 선택에 활용한다.

반대로 레지스트리로부터 적합한 템플릿을 검색하지 못한 경우에는 자동 템플릿 생성(Automated Template Construction)을 통해서 새로운 템플릿을 생성하고 이를 기반으로 서비스를 구성한다. 자동 생성된 템플릿은 재활용을 위해 템플릿 레지스트리에 등록될 수 있다.

3) 서비스 템플릿 인스턴스 생성

서비스를 위한 템플릿이 결정되고 나면 실세계에 존재하는 서비스와 바인딩하기 위한 템플릿 인스턴스 생성을 수행한다. 템플릿 인스턴스를 생성하기 위해서는 템플릿의 의미를 기반으로 시멘틱 서비스 레지스트리(Service Registry)를 검색하여 적절한 서비스 인스턴스에 대한 OWL-S[1] Description을 획득해야 한다. 획득한 OWL-S 정보 및 서비스 템플릿을 기반으로 템플릿 인스턴스를 생성하고 다음 단계로 진행한다.

4) 실행 가능한 서비스 프로세스 생성

서비스 컴포지션의 마지막 단계로서 템플릿 인스턴스의 정보를 기반으로 실행가능한 형태의 프로세스를 생성한다. 실행가능한 프로세스는 기본적으로 BPEL4WS를 기반으로 생성되는데 선택에 따라 다양한 프로세스 포맷(OWL-S Process 등)을 지원할 수 있는 유연한 생성 메커니즘을 제공하여야 한다. [그림 3]은 하나의 온톨로지 예시를 보여주고 있다.

```
<owl:Class rdf:ID="RobotService">
  <rdfs:label>RobotService</rdfs:label>
  <rdfs:comment>
    Top level of an ontology of robot services
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="&#x26;profile;#Profile" />
</owl:Class>

<owl:Class rdf:ID="RobotTestService">
  <rdfs:label>RobotTestService</rdfs:label>
  <rdfs:comment>
    Definition of robot test services
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="&#x26;RobotService" />
</owl:Class>
```

그림 3. 온톨로지의 예제

IV. STDL 기반 서비스 템플릿 저작 도구 구현

1. 시스템 설계

구현된 시스템은 OWL 문서의 정확한 Parsing 처리 및 Validation 체크를 위해서 XML Parser와 Jena API를 통하여 OWL 문서의 로딩 및 저장시 사용되며, OWL 문서의 Validation Check를 위해서 사용된다. 또한 JGraph와 JGraphPad는 구현된 시스템의 그래픽적인 사용자 인터페이스 기능을 제공한다. 다음은 각 컴포넌트의 간략한 설명이다.

먼저 XML Parser는 Jena API 모듈과 연동되며, 기본적으로 OWL 문서를 읽는 과정에서 문서를 파싱하여 메모리에 로딩하는 역할을 수행한다. Jena API는 OWL 문서를 로딩하면서 관련된 템플릿 정보들과 관련 링크된 파일들을 모두 읽어와 문서를 분석할 수 있도록 하며, 문서에 대한 Validation Check 기능을 제공한다.

JGraph & JGraphPad는 Jena를 통해서 로딩된 OWL 문서를 그래픽컬하게 화면에 보여주기 위한 모듈로 URBuilder 모듈과 유기적으로 관계하며 OWL문서의 템플릿 및 템플릿간의 관계 정보를 WIZWIG 형태로 보여준다. URBuilder(구현된 시스템)은 URBuilder의 핵심 모듈로 외부 모듈들과 관계하며, OWL 문서를 화면에 보이고, 업데이트 및 관리하는 기능을 제공한다.

구현된 시스템은 JGraph & JGraphPad 와 Jena를 기반으로 하여 상위 5가지 모듈로 구성되어 있다. 각각의 모듈에 대한 상세한 기능 설명은 다음과 같다. UI Module은 UI 모듈은 URBuilder의 전체 UI에 나타나는 컨트롤 컴포넌트들을 배치하고 배치된 컴포넌트를 통해서 데이터를 출력하거나 입력하는 역할을 수행하는 모듈로 전반적인 화면 처리를 위한 모듈이다. 본 모듈을 JGraph 및 JGraphPad 모듈과 밀접하게 관계되면서 상호 데이터를 주고받는 관계이다. Action Module은 UI 모듈의 각 컨트롤 컴포넌트 중에서 Event 처리를 수행해야 하는 일부 컴포넌트들에 대해서 각종 Event 처리를 처리하는 모듈이다. 본 모듈은 UI 모듈과 밀접하게 관계되어 상호 이벤트를 주고받는 역할을 수행한다.

Cell View Module은 Cell View 모델은 UI 모델 및 JGraph & JGraphPad 모듈과 밀접하게 연관되어있으며 템플릿을 이용하여 OWL 문서를 생성하는 작업공간에서 그려지는 일부 컴포넌트들에 대한 형태를 정의한 모듈로 Graphic적으로 뷰 모델을 그려주는 역할을 수행한다. Config Module은 URBuilder에서 기본적으로 설정이 필요한 것들에 대한 정보를 관리하는 모듈로, 신규 생성 혹은 변경된 정보에 대한 저장 및 관리기능을 수행하는 모듈이다. OWL Module은 OWL모듈은 JGraph & JGraphPad 모듈 및 Jena 모듈과 관계를 가지며, 사용자가 작업중인 문서에 대해서 화면상에서 나타나는 문서의 정보를 내부적으로 필요한 자료구조 형태로 유지하는 역할을 수행하는 모듈로, OWL 문서와 그래픽 문서인 JGX 문서에 대해서 상호 전환이 가능한 형태로 데이터를 유지하고 관리하는 기능을 수행한다.

2. URBuilder 구현

1) 메인 화면

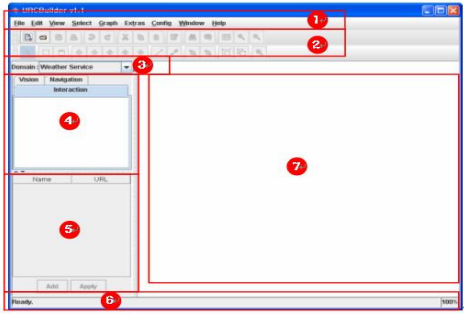


그림 4. URCBuilder 메인 화면

[그림 4]에서 볼 수 있듯이 URCBuilder는 총 7가지 영역으로 나뉘며, 각 영역별로 서로 다른 기능을 제공함으로써 URCBuilder를 통해 서비스 템플릿을 쉽게 제작할 수 있다. 다음은 각 영역별 기능에 대한 설명이다. 1번 영역 [메뉴]은 URCBuilder 사용을 위한 메뉴 영역으로 사용자의 선택을 돕도록 상황에 따라 적절하게 활성화/비활성화 되어 보여준다. 2번 영역 [툴바]은 사용자가 메뉴를 통하지 않고, 빠른 접근이 가능하며 필요한 기능을 툴바 형태로 제공하며, 해당 툴바를 이용하여 빨리 기능을 수행할 수 있게 한다. 3번 영역 [도메인]은 템플릿 라이브러리들을 도메인으로 묶어서 사용자가 필요한 영역의 팔레트 도메인을 접근할 수 있도록 한다. 서비스 도메인은 날씨, 교통, 로봇 서비스 등으로 구성된다.

4번 영역 [템플릿 팔레트]은 3번 영역을 통해서 선택한 도메인에 해당하는 템플릿 팔레트들을 표현하는 영역으로 각각의 팔레트별 해당 템플릿 라이브러리를 가지고 있는 형태로 구성된다. 또한 각 서비스 도메인별로 서비스 블록(Basic Template & Aggregate Template) 들을 관리하기 위한 파일이 존재한다. 5번 영역 [속성]은 4번 영역에서 선택된 템플릿 라이브러리에 대한 상세 정보를 보이고, 해당 정보를 수정 및 변경할 수 있도록 기능을 제공한다. 또한 7번 영역의 작업공간에서 선택된 템플릿의 정보를 이곳을 통해서 보여주며, 정보를 수정, 변경 할 수 있도록 기능을 제공한다. 6번 영역 [상태]은 URCBuilder를 사용하는 과정에서 필요한 상태정보를 이곳을 통해서 간단하게 보여주는 역할을 수행한다. 7번 영역 [작업]은 실제 사용자가 템

플릿을 만드는 작업공간을 제공하는 영역으로 MDI(Multi Document Interface) 형태로 구성된다.

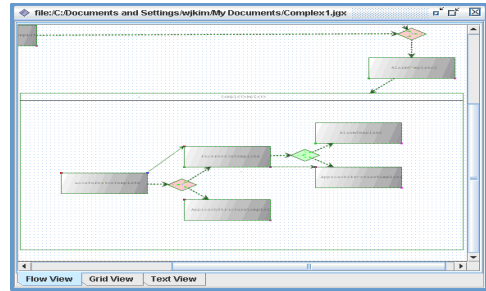


그림 5. 작업 영역의 Flow View

[그림 5]은 7번 영역인 템플릿 작업 영역을 보여준다. Flow View는 다이어그램 방식을 이용하여 서비스 템플릿을 저작할 수 있는 기능을 제공하며, 또한 서비스 템플릿 저작을 위한 모델링 아이콘은 상황에 맞게 활성 및 비활성 기능을 제공한다. 서비스 블록 팔레트 패널에서 서비스 블록(Basic Template 혹은 Aggregate Template)을 선택하고 이를 Drag-and-Drop 기능을 이용하여 편집할 수 있는 기능을 제공한다.

그리드 뷰는 사용자가 그래픽 기반으로 작성한 것을 OWL 문서의 형태에 입각하여 트리(tree) 형태로 보여줌과 동시에 Aggregate Template과 Basic Template을 구분하여 보여준다. 루트는 템플릿이 되며, 하위에 Aggregate Template 클래스 및 Basic Template 클래스 등이 위치하게 된다. Aggregate Template은 하위에 Basic Template들을 가진다. 각 Basic Template은 하위에 hasInput, hasOutput, hasPrecondition, hasEffect의 Object property들을 가진다.

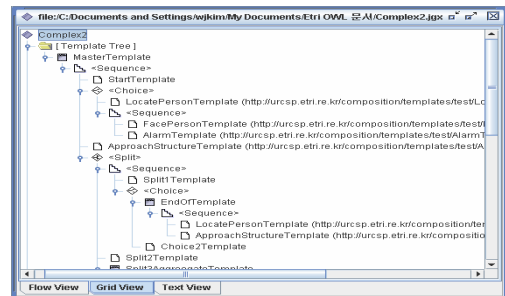


그림 6. 작업 영역의 그리드 뷰

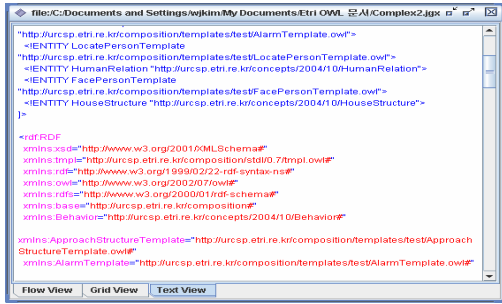


그림 7. 작업 영역의 Text View

[그림 7]에서 Text View는 소스에 대한 텍스트 방식의 뷰 기능 및 저작 환경을 제공하며, Flow View 저작 환경에서 저작한 내용을 STDL(OWL) 문서 타입에 맞게 Serialize하여 텍스트 방식으로 화면에 출력하여 보여준다.

V. 결론

유비쿼터스 공간 상에 존재하는 수많은 서비스들을 하나의 서비스로 구성하고 실행하기 위해서는 이종 플랫폼 상에 탑재되어 있는 서비스 객체들을 지능적으로 통합할 수 있어야 한다. 또한 유비쿼터스 환경의 지능형 로봇도 유비쿼터스 에이전트로 URC용 서비스를 제공하기 위해서는 사용자의 상황 정보를 처리하는 능력이 필요하다.

본 논문은 유아교육 서비스의 시맨틱 정보를 기반으로 서비스 에이전트의 지능적이고 자율적인 서비스 제공을 위해 제시된 시맨틱 웹 서비스 환경을 기반으로 URC 서비스 컴포지션 시스템을 구현하였다. 구현된 시스템은 URC 서비스를 표현하는 메타 정보를 기술한 서비스 템플릿 객체 모델을 기반으로 자동으로 웹 서비스를 선택하고 구성하여 적절한 로봇 서비스 플랜을 생성해준다. 또한 구현한 시스템은 기존의 웹 서비스들과 다른 컴포넌트를 조합하고 연결하기 위해 필요한 액티비티들로 구성된 일련의 프로세스인 서비스 플랜을 생성하며, 생성된 서비스 플랜을 기반으로 서비스 실행 엔진을 통해 로봇 서비스를 제공하는 특징을 가진다.

마지막으로 본 논문은 STDL을 기반으로 URC용 서

비스 템플릿 생성, 추가, 삭제, 수정 등의 기능뿐만 아니라 서비스 템플릿 리소스들에 대한 그래픽한 처리 기능을 제공한다. 서비스 템플릿 과정에서 사용자 편의적인 환경 제공을 위해 플로우 뷰 스타일, 그리드 뷰 스타일, 텍스트 뷰 스타일 등의 편집 환경을 구현했으며, 또한 실제 로봇 구동에 필요한 로봇API를 참조하고 이를 기반으로 추상 서비스 블록을 구체화하여 서비스 템플릿을 쉽게 작성할 수 있는 기능을 제공하게 된다.

향후에는 본 시스템을 확장하여 강력한 표현력을 지원하는 온톨로지 언어인 OWL-S를 그래픽환경에서 편집할 수 있는 저작도구를 개발하고자 한다. 아울러 좀 더 복잡한 템플릿의 표현과 중첩된 템플릿을 편집할 수 있도록 기능을 보완할 것이다. 마지막으로 본 연구가 유아교육의 학문적 분야는 물론 유아교육 현장에서도 효율적인 유아교육 관련 콘텐츠 제작과 활용에 기여할 수 있기를 기대한다.

참고 문헌

- [1] S. Bechhofer, I. Horrocks, C. Goble, and Robert Stevens, "OilEd: a Reasonable Ontology Editor for the Semantic Web," Proc. of the German conference on Artificial Intelligence, KI2001, pp.396-408. Springer Verlag, LNAI Vol.2174, 2001.
- [2] DAML UML Enhanced Tool, available <http://grcnet.grci.com/maria/www/CodipSite/Tools/Tools.html> last accessed on June 24th 2003.
- [3] J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crubezy, H. Eriksson, N. Fridman Noy, and Samson Tu, "The Evolution of Protege: An Environment for Knowledge-Based Systems Development," International Journal of Human Computer Studies, Vol.58, No.6, pp.737-758, 2003.
- [4] A. Kalyanpur, "SMORE-Semantic Markup,

Ontology, and RDF Editor," <http://www.mindswap.org/papers/smored.pdf>.

[5] S. Y. Oh, M. Y. Chung. ezOWL plugin for Protege. <http://iweb.etri.re.kr/ezowl/>, 2003.

[6] Y. Sure, S. Stab, and J. Angele, "OntoEdit: Guiding Ontology Development by Methodology and Inferencing," Proc. of the Confederated International Conferences CoopIS, DOA and ODBASE 2002, pp.1205-1222. Springer Verlag, LNCS Vol.2519, 2001.

[7] L. Thorsten, N. Olaf, "OntoTrack: A Semantic Approach for Ontology Authoring," Vol.3, No.2, pp.116-131, Journal of Web Semantics, 2005.

[8] URC(Ubiquitous Robotic Companion), "IT 차세대 성장동력 기획보고서(지능형서비스로봇)", 2003.

[9] 김재홍, 하영국, 박천수, 장민수, 이미경, 윤여훈, 손주찬, "URC를 위한 명령 분석 및 서비스 계획 기술", 전자통신동향분석, 제20권, 제2호, 2005.

[10] 강상승, 하영국, 손주찬, "템플릿 객체 모델을 이용한 URC 서비스 컴포지션 시스템", 한국정보과학회 2005년 추계학술대회, 제32권, 제2호, pp.565-567, 2005.

[11] 손주찬, 김재홍, 하영국, 장민수, 박천수, 이미경, "URC 지능형 서비스 기술 개발 동향", 주간기술동향 제1190호, 2005.

[12] 이미경, 장민수, 고영철, 손주찬, URC용 서비스 분석을 위한 컨텍스트 데이터 처리 및 전송 구조, 한국정보처리학회 2004년 추계학술대회, 제11권, 제2호, pp.1579-1582, 2004.

저 자 소 개

한 선 아(Sun-Ah Han)

정회원



- 1994년 : 서울여자대학교 아동학과 유아교육전공(문학사)
- 1996년 : 서울여자대학교 대학원 유아교육전공(문학석사)
- 2003년 : 서울여자대학교 대학원 유아교육전공(문학박사)

▪ 2008년 ~ 현재 : 목원대학교 유아교육과 교수

<관심분야> : 유아컴퓨터교육, 유아문학교육, 질적연구