

Reusing XML Objects in Context-Aware Workflow Model for Improving the Development of Service Scenario

Yeon Seung Yoo[†] · Jong Hyeok Mun^{††} · Do Hyung Kim^{†††} · Jong Sun Choi^{††††} · Jae Young Choi^{†††††}

ABSTRACT

In order to provide customized services according to a specific user or environment, various service scenarios should be developed based on context-aware workflow model. As the context-aware workflow model is constructed using limited context information and service information in the service domain, overlapping elements can occur in many service scenarios. The repetitive work process that results from these overlapping elements delays the development process of the service scenario. Therefore, the elements of the context-aware workflow model must be reused to solve the unnecessary work processes of service scenario development. In this paper, we propose a reuse method XML Object in context-aware workflow model to improve the process of service scenario development. The proposed method documents and manages the independent XML Object of the context-aware workflow model and reuses it by invoking the unit document in the service scenario development process. It can also be applied to new service scenarios by changing the attribute values of reusable elements. Experiments show example that the development process of the service scenario is simplified by reusing the elements of the context-aware workflow model.

Keywords : Service Scenario Editor, Context-Aware Workflow Model, Reusing XML Objects, Improvement of Scenario Development

서비스 시나리오 개발 프로세스를 개선시키기 위한 상황인지 워크플로우 모델에서 XML 객체의 재사용

유연승[†] · 문종혁^{††} · 김도형^{†††} · 최종선^{††††} · 최재영^{†††††}

요약

상황인지 워크플로우를 바탕으로 특정 사용자 또는 환경에 따라 맞춤형 서비스를 제공하기 위해서는 상황인지 워크플로우 모델을 기반으로 다양한 서비스 시나리오를 개발해야 한다. 상황인지 워크플로우 모델은 서비스 도메인 내의 한정된 상황 정보와 서비스 정보를 활용하여 구성되기 때문에 다수의 서비스 시나리오에는 중복되는 요소들이 발생할 수 있다. 이러한 중복되는 요소들로 인해 발생하는 반복적인 작업 프로세스는 서비스 시나리오의 개발 프로세스를 지연시킨다. 따라서 서비스 시나리오 개발의 불필요한 작업 프로세스를 해소하기 위해 상황인지 워크플로우 모델의 요소들은 재사용되어야 한다. 본 논문에서는 서비스 시나리오 개발의 프로세스 개선을 위한 상황인지 워크플로우 모델에서 XML 객체의 재사용 방법을 제안한다. 제안하는 방법은 상황인지 워크플로우 모델에서 독립적으로 사용 가능한 XML 객체를 문서화하여 재사용 요소로 관리하고 서비스 시나리오 개발 과정에서 호출을 통해 재사용한다. 또한 재사용 요소의 속성 값을 변경하여 새로운 서비스 시나리오에 적용할 수 있다. 실험에서는 상황인지 워크플로우 모델의 요소들을 재사용하여 서비스 시나리오의 개발 프로세스가 간소화되는 과정을 시나리오 예제를 통해 보인다.

키워드 : 서비스 시나리오 편집기, 상황인지 워크플로우 모델, XML 객체의 재사용, 시나리오 개발 프로세스 개선

1. 서론

상황인지는 IoT (Internet of Things) 환경에서 발생하는

대량의 센서 데이터를 수집 및 분석하여 주변 환경을 판단하는 기술이다[1]. 이와 같은 상황인지 기술은 스마트 공장, 지능형 로봇, 재난 구호, 헬스케어 등 다양한 산업 분야에 적용되어 활발히 연구가 진행되고 있다[2, 3]. 센서 데이터를 추상화하여 주변 환경을 표현한 상황 정보는 상황인지 기반의 서비스를 제공하기 위한 중요한 수단이 된다[4]. 하지만 상황 정보를 서비스의 실행 조건으로 이용한 이벤트 기반의 서비스는 변화하는 환경에 따른 유연한 서비스를 보장하지 못한다. 이를 위해 다양한 패턴으로 서비스의 흐름을 조정할 수 있는 워크플로우 기술이 활용된다.

※ 이 논문은 2019년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2019R1A2C1007861).
† 비회원: 송실대학교 지능형로봇연구소 연구원
†† 준회원: 송실대학교 컴퓨터학과 박사과정
††† 비회원: 송실대학교 컴퓨터학과 석사과정
†††† 정회원: 송실대학교 컴퓨터학부 부교수
††††† 종신회원: 송실대학교 컴퓨터학부 교수
Manuscript Received : September 20, 2019
First Revision : December 16, 2019
Accepted : January 8, 2020
* Corresponding Author : Jong Sun Choi(jongsun.choi@ssu.ac.kr)

워크플로우는 일련의 작업 절차를 따라 문서, 정보, 업무 등이 한 참가자에서 다른 참가자로 전달되는 비즈니스 프로세스의 전체 또는 일부를 자동화하는 기술이다[5]. 기존 워크플로우 기술을 응용한 상황인지 워크플로우는 다수의 서비스를 순차적으로 조합하고, 서비스 간의 전이 조건으로 상황 정보를 이용하여 자동화된 맞춤형 서비스를 제공할 수 있다[6]. 상황인지 워크플로우를 바탕으로 특정 사용자 또는 환경에 따라 맞춤형 서비스를 제공하기 위해서는 상황인지 워크플로우 모델을 기반으로 다양한 서비스 시나리오를 개발해야 한다[7]. 상황인지 워크플로우 모델은 서비스 도메인 내의 한정된 상황 정보와 서비스 정보를 활용하여 구성되기 때문에 서비스 시나리오 간에는 중복되는 요소들이 발생할 수 있다[17]. 시나리오 개발자는 다수의 상황인지 워크플로우 모델에 중복되는 요소들을 포함하기 위해 동일한 작업을 반복적으로 수행해야 한다. 이러한 반복 작업은 서비스 시나리오 개발의 작업 프로세스를 지연시키게 된다. 따라서 상황인지 워크플로우 모델을 구성하는 개별적 요소들은 재사용되어야 한다.

워크플로우 모델의 재사용은 공동 작업자와 지식 공유, 시간 절약, 표준화, 디버깅 등의 장점을 갖는다[8]. 일반적으로 워크플로우 모델은 구조화된 시나리오 문서로 생성하여 관리된다. 그렇기에 시나리오 개발자는 사전에 작업된 시나리오 문서를 호출하여 워크플로우 모델을 재사용할 수 있다[9]. 그러나 시나리오 문서에 기술된 워크플로우 모델의 일부 요소를 재사용하기 위해서는 워크플로우의 전체 흐름을 분석하고 해당 요소를 추출하는 작업이 필요하다. 이와 같은 작업은 시나리오 개발자에게 재사용에 대한 많은 노력과 작업 시간을 요구하게 된다[18].

이에 본 논문에서는 서비스 시나리오 개발의 프로세스를 개선하기 위한 상황인지 워크플로우 모델에서 XML 객체의 재사용 방법을 제안한다. 상황인지 워크플로우 모델의 재사용 단위는 상황인지 워크플로우를 구성하는 XML 객체(Object)로 Element를 재사용 요소로 정의한다. 제안하는 방법은 상황인지 워크플로우 모델의 재사용 요소(Reusable Element)를 정의하고 각 요소들을 문서화하여 관리한다. 그리고 서비스 시나리오 개발 과정에서 저장된 재사용 요소 문서의 호출을 통해 각 요소들을 재사용한다. 또한 새로운 서비스 시나리오에 재사용하는 요소들이 올바르게 적용될 수 있도록 요소의 속성 값을 변경할 수 있도록 지원한다. 실험에서는 상황인지 워크플로우 모델의 개별적인 재사용 요소들을 문서 단위로 재사용하여 서비스 시나리오의 개발 프로세스가 간소화되는 과정을 예제 시나리오를 통해 보인다.

2. 관련 연구

본 장에서는 상황인지 워크플로우 모델과 워크플로우 모델의 재사용 사례에 대해 조사하고 구조화된 문서를 효율적으로 관리하는 방법을 살펴본다.

ASTRO-CAptEvo[10]는 항구에서 IoS (Internet of Services) 기반의 자동차 물류 시나리오를 자동화하기 위해 상황인지 워크플로우 기술을 적용한 프레임워크이다. 자동차 물류 시나리오는 서비스 환경의 역동성, 복잡성, 가용성으로 인해 사전에 설계한 서비스 시나리오를 동적으로 변경할 필요가 있다. 이를 위해 상황인지 워크플로우를 워크플로우 조각으로 관리하고 런타임 환경에서 특정 상황에 적합한 워크플로우 조각을 처리하여, 변화하는 환경에 따라 유연하게 서비스 시나리오를 대처할 수 있다. 서비스 시나리오는 전통적인 워크플로우 언어인 BPEL (Business Process Execution Language)을 확장한 APFL (Adaptable Pervasive Flows Language)을 바탕으로 설계한다. APFL은 워크플로우를 구성하는 입력, 출력, 활동, 제어 흐름 외에도 상황 조건에 따른 서비스 활동을 추가하여 상황인지 기반의 워크플로우를 모델링할 수 있다.

CONFlexFlow (Clinical cONtext based Flexible workFlow) [11]는 임상 경로를 CDSS (Clinical Decision Support Systems)에 통합하기 위한 프레임워크로서, 확립된 임상 표준 지침에 따라 환자의 치료 단계를 상황인지 워크플로우로 모델링한다. 의료 시설의 자원을 바탕으로 모델링된 상황인지 워크플로우는 개별 환자의 변경된 서비스 요구 사항을 충족하기 위해 지속적으로 조정된다. 상황인지 워크플로우의 서비스 전이는 OWL-DL (Web Ontology Language-Description Logic)을 통해 정의된 온톨로지 명세를 따른다. 온톨로지는 도메인 내의 각 정보를 명시적인 형태로 정의하고 정보 간의 관계를 표현할 수 있다. 또한 BPMN (Business Process Model and Notation)을 이용하여 그래픽 기반의 상황인지 워크플로우를 손쉽게 모델링할 수 있다. ASTRO-CAptEvo와 CONFlexFlow는 동적인 서비스 환경에서 서비스 시나리오를 빠르게 재구성하기 위해 상황인지 워크플로우 모델을 재사용한다. 그러나 서비스 시나리오 개발 과정에서 상황인지 워크플로우 모델을 재사용하기에는 어려움이 있다.

iPOJO Flow[12]는 선언적 워크플로우 정의 언어인 iCDL (iPOJO Composition Description Language)을 기반으로 상황인지 워크플로우를 모델링한다. iCDL은 워크플로우의 구조 요소, 동작 요소, 제어 요소를 바탕으로 상황인지 워크플로우의 각 요소를 구성한다. 컴포넌트 기반의 워크플로우 서비스를 제공하는 iPOJO Flow는 서비스 저장소에서 제공하는 서비스 정보를 재사용하여 상황인지 워크플로우를 손쉽게 모델링할 수 있다. 하지만 상황인지 워크플로우 모델의 재사용 단위가 서비스 요소로 한정되어 추가적인 요소의 재사용이 제한된다.

DWMR (Data-centric Workflow Model Reuse framework) [13] 프레임워크는 데이터의 종속성과 데이터의 작업 관계를 고려하여 워크플로우 모델을 분류하고, 이후 시나리오 개발자의 요구 사항에 따라 재조합하여 워크플로우 모델을 재사용한다. 워크플로우 모델을 재사용하기 위해서는 워크플로우 모

델의 조직화, 저장, 검색, 구성과 같은 주요 기능이 제공되어야 한다. 이와 같은 기능을 통해 기존 워크플로우 모델은 비즈니스 프로세스의 개발 목적에 따라 효율적으로 재사용한다.

S-IOFLOW[14]는 비즈니스 프로세스 개발의 협력을 달성하기 위해 IOWF (Inter-Organizational Workflows) 워크플로우 모델을 재사용한다. 캡슐화된 서비스를 호출하는 다른 참가자는 재사용하는 워크플로우 모델을 작업 중인 워크플로우에 결합하기 위해 추가적인 제어 흐름을 표현한다. 이에 다수의 참여자는 워크플로우 모델의 재사용하기 위해 IOWF 기반 워크플로우 모델을 서비스 단위로 캡슐화한다.

XMap[15]은 XML (Extensible Markup Language)과 RDB (Relational Database)를 기반으로 구조화된 문서를 관리하는 방법을 제안한다. XMap은 XML 데이터를 RDB에 저장하고 쿼리하여 XML 문서를 재사용하는 방식을 지원한다. 하지만 XML 문서에 포함된 요소 정보들이 단일 문서 단위로 재사용되어 문서 간의 참조를 통한 다중 문서 통합을 지원하지 못 한다.

워크플로우 기술이 비즈니스 프로세스 관리의 표준 솔루션으로 점차 보급됨에 따라 전자 상거래 시대에는 수많은 워크플로우 모델이 설계되어 사용되었다. 이러한 워크플로우 모델은 워크플로우 도메인 지식을 위한 중요한 자원이 될 수 있다. 따라서 새로운 워크플로우 모델을 설계하는데 기존 워크플로우 모델이 재사용될 수 있어야 하고 이를 위한 연구가 진행되었다.

워크플로우 모델의 재사용을 위한 연구로, DWMR과 S-IOFLOW는 워크플로우 모델을 재조합하거나 서비스 단위로 캡슐화하여 재사용함으로써 시나리오 개발의 작업 프로세스를 단축시킬 수 있다. 그러나 재사용하는 워크플로우 모델에 포함되는 요소들의 속성 값들을 세부적으로 수정하는 기능이 결여되어, 동적인 서비스 환경에서 실행되는 상황인지 워크플로우 모델을 재사용하기에는 한계가 있다.

따라서 서비스 시나리오의 개발 프로세스를 개선하기 위해 상황인지 워크플로우 모델의 요소를 재사용되어야 한다. 또한 상황인지 워크플로우 모델의 독립적인 요소들을 재사용 여부에 따라 재사용 요소 문서로 생성하고, XML 기반의 재사용 요소 문서를 RDB에서 효율적으로 관리해야 한다.

3. 상황인지 워크플로우 모델의 XML객체 재사용 방법

본 장에서는 상황인지 워크플로우를 모델링하기 위해 설계된 CAWL (Context-Aware Workflow Language)[16]의 개념과 재사용 단위를 살펴본 후, 재사용 요소 생성을 위한 서비스 시나리오 문서 편집기 개요와 동작과정을 설명한다.

3.1 배경지식

상황인지 워크플로우를 바탕으로 다수의 서비스 시나리오를 개발하기 위해서는 상황인지 워크플로우를 일관된 방법으

로 표현할 수 있는 구조화된 모델이 필요하다. CAWL은 개별적인 서비스를 조합하여 워크플로우를 구성하고 서비스 간의 전이 조건으로 상황 정보를 이용하여 상황인지 기반의 워크플로우를 모델링할 수 있다.

1) CAWL의 개념

CAWL은 Fig. 1과 같이 최상위 요소인 <CAWL> 요소를 따라 다양한 하위 요소들을 구성하여 상황인지 워크플로우를 모델링할 수 있는 워크플로우 언어이다.

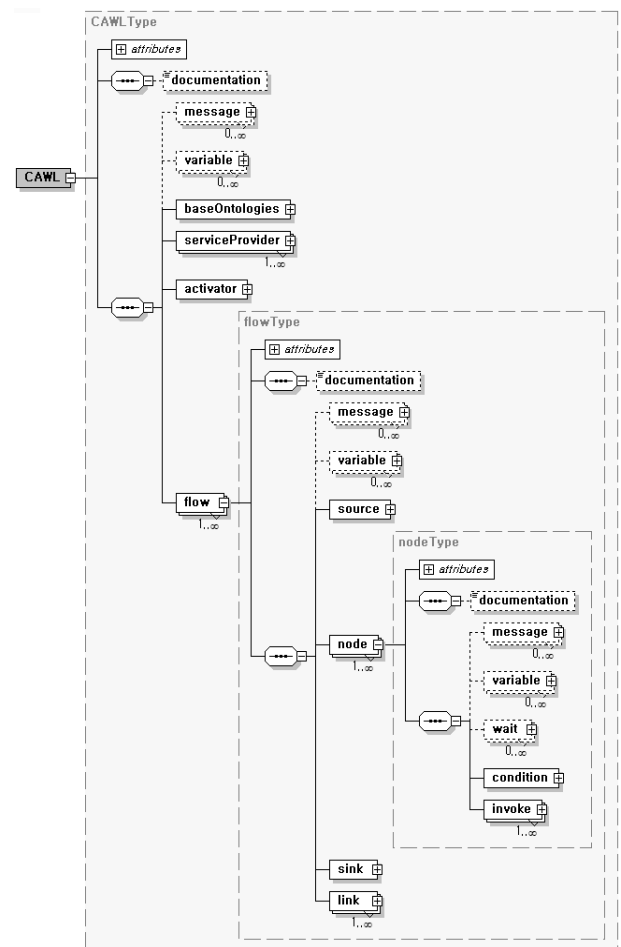


Fig. 1. Overview of the CAWL Schema

<message>와 <variable> 요소는 워크플로우를 순차적으로 처리하기 위한 데이터를 정의하고 유지하는 요소로 사용한다. <baseOntologies>와 <serviceProvider> 요소는 유효한 상황 정보를 제공하는 상황인지 엔진의 접근 정보와 실행 가능한 서비스 정보를 제공하는 서비스 에이전트의 접근 정보를 정의하기 위해 사용한다. 또한 워크플로우의 시작점을 나타내는 <activator> 요소와 서비스 노드들의 집합으로 구성되는 <flow> 요소가 있다.

워크플로우를 정의하는 <flow> 요소에는 워크플로우의 시작과 종료를 나타내는 <source>, <sink> 요소와 서비스 노드

를 정의하는 <node> 요소가 있다. <node> 요소를 통해 정의된 다수의 서비스 노드는 <link> 요소를 통해 연결한다. <node> 요소는 상황인지 기반의 서비스를 실행하기 위한 하위 요소들을 포함한다. <condition> 요소는 서비스 노드의 전이 조건인 상황 정보를 정의하고, <invoke> 요소는 해당 상황에 따라 실행되어야 하는 서비스 정보를 정의한다. 다중 워크플로우 서비스를 제공하는 CAWL은 워크플로우 서비스 간의 흐름을 제어하기 위해 <wait> 요소가 사용된다.

2) CAWL의 재사용 단위

상황인지 워크플로우 모델의 재사용을 위해서는 CAWL의 재사용 단위가 정의되어야 한다. 본 논문에서는 CAWL을 구성하는 XML 객체(object)인 Element를 서비스 시나리오 편집의 재사용 단위로 정의하며, 재사용 요소(reusable element, 이하 RE)로 지칭한다. RE는 CAWL에서 독립적으로 사용될 수 있는 요소로써 Fig. 2와 같이 분류되고, 문서화하여 상황인지 워크플로우 모델 설계 단계에서 재사용된다.

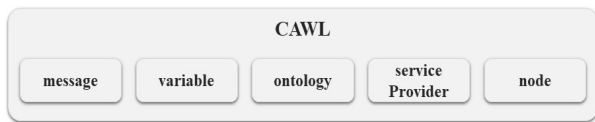


Fig. 2. Reusable Elements of the CAWL

CAWL RE는 message, variable, ontology, serviceProvider, node로 분류되며 각 CAWL RE는 계층적인 구조를 따라 하위 태그 정보를 조합하여 구성된다. 서비스 노드 간의 데이터를 전달하기 위한 <message> 요소와 데이터의 값을 변경하기 위한 <variable> 요소는 워크플로우를 순차적으로 처리하는데 필요한 파라미터를 기술하는 요소로 서비스 시나리오의 유형에 따라 재사용된다. 외부 모듈의 접근 정보를 정의하는 <ontology>는 워크플로우 노드의 조건을 판단하기 위해 필요한 상황 정보를 참조하기 위한 정보를 포함하고 <serviceProvider> 요소는 서비스 에이전트의 참조 정보를 포함한다. 마지막으로 <node>는 워크플로우의 단일 서비스 노드 구성 정보를 포함한 재사용 단위로 사용된다.

재사용 가능한 CAWL의 요소들은 각 요소의 스키마를 기반으로 CAWL RE로 생성되어 관계형 데이터베이스에 트리 구조로 저장된다. 이는 XMap에서 XML 문서를 관리하기 위한 방법으로 본 논문에서는 요소 단위로 데이터베이스에 관리한다. RE는 트리 구조로 저장되며 쿼리를 통해 참조되어 서비스 시나리오 문서에 삽입된다. Fig. 3은 <message>의 트리형태 저장 구조를 나타낸다. 저장된 CAWL RE를 호출할 경우, 각 테이블에 저장된 ID를 참조하여 하위 테이블의 정보에 접근한다.

서비스 시나리오 개발자는 서비스 시나리오를 개발하는 단계에서 서비스 시나리오 문서 편집기를 통해 생성된 CAWL RE 문서를 활용하여 효율적으로 서비스 시나리오 문서를 작성할 수 있다.

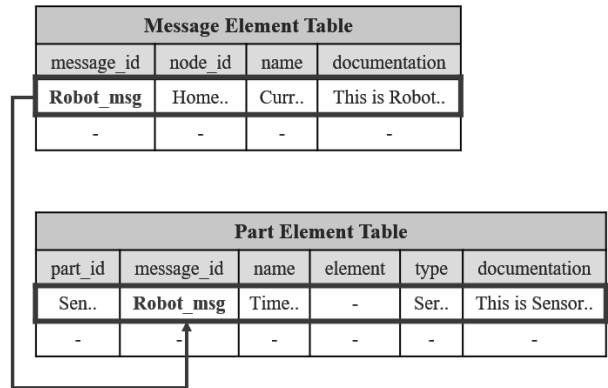


Fig. 3. Tree Structure for CAWL RE Reference in Relational Database

3.2 서비스 시나리오 문서 편집기 개요

상황인지 워크플로우 모델의 독립적인 요소들을 CAWL RE 문서로 생성 및 관리하고, 재사용을 위한 CAWL RE 문서를 활용할 수 있는 서비스 시나리오 문서 편집기의 구조는 Fig. 4와 같다.

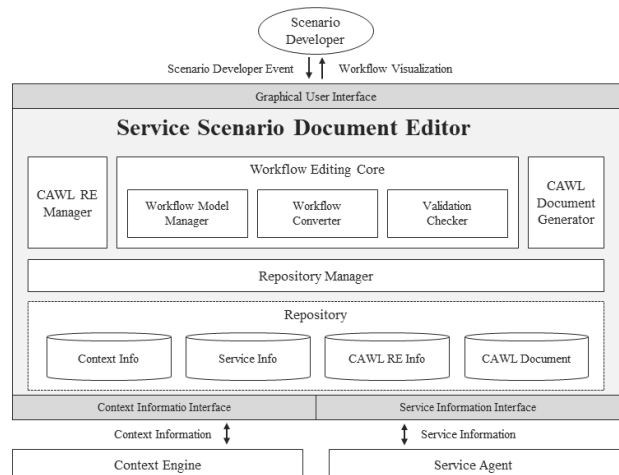


Fig. 4. Architecture of the Service Scenario Document Editor

제안하는 서비스 시나리오 문서 편집기는 시나리오 개발자의 이벤트 동작에 따라 실행되고, 실행 결과를 GUI (Graphical User Interface) 기반의 편집 화면을 통해 시각화한다. 또한 상황인지 워크플로우 모델에 표현되는 서비스 도메인의 상황 정보와 서비스 정보를 제공받기 위해 외부 모듈인 Context Engine과 Service Agent와 연동된다.

CAWL RE Manager는 상황인지 워크플로우 모델의 재사용 가능한 요소들을 CAWL RE 문서로 편집하는 기능을 제공한다. 상황인지 워크플로우를 모델링하고 문서화하는 Workflow Editing Core는 Workflow Model Manager, Workflow Converter, Validation Checker로 구성된다. Workflow Model Manager는 상황인지 워크플로우 모델을 다이어그램 형태로 편집하는 기능을 제공한다. Workflow Converter는

다이아그램 형태로 표현된 상황인지 워크플로우 모델을 텍스트 데이터로 변환하는 작업을 수행한다. Validation Checker는 CAWL 스키마를 바탕으로 변환된 텍스트 데이터의 유효성 검증 작업을 수행한다. CAWL Document Generator는 상황인지 워크플로우 모델이 표현된 텍스트 데이터를 XML 기반의 CAWL 시나리오 문서로 생성한다. Repository Manager는 Repository에 저장된 정보를 각 모듈의 요청에 따라 제공하는 역할을 수행한다. Repository는 상황 정보를 관리하는 Context Info 저장소, 서비스 정보를 관리하는 Service Info 저장소, CAWL RE 정보를 관리하는 CAWL RE 저장소, CAWL 시나리오 문서를 관리하는 CAWL Document 저장소로 구성된다.

3.3 서비스 시나리오 문서 편집기 동작 과정

서비스 시나리오 문서 편집기는 상황인지 워크플로우 모델의 독립적인 요소들을 CAWL RE로 관리하고, 서비스 시나리오 개발 과정에서 저장된 CAWL RE 문서를 재사용하여 CAWL 시나리오 문서를 생성한다.

1) CAWL RE 생성 과정

CAWL RE 문서는 Fig. 5와 같이 CAWL RE Manager가 동작하여 생성된다.

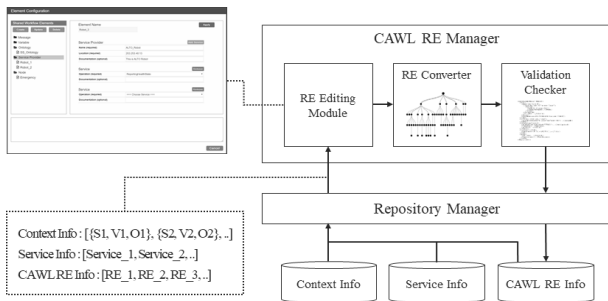


Fig. 5. Generation Flow of the CAWL RE Document

시나리오 개발자는 RE Editing Module에서 제공하는 CAWL RE 설정 화면을 통해 CAWL RE를 작성한다. 이를 위해 Repository Manager는 CAWL 요소를 구성하는데 필요한 상황 정보, 서비스 정보 그리고 사전에 작업이 완료된 CAWL RE 정보를 제공한다. RE Converter는 시나리오 개발자에 의해 입력된 CAWL 요소 정보를 트리 자료 구조로 관리하고, 트리의 하위 노드부터 언파싱 (Unparsing) 작업을 수행하여 텍스트 데이터로 변환한다. Validation Checker는 변환된 CAWL 요소 정보를 CAWL의 해당 하위 스키마를 바탕으로 유효성 검증 작업을 수행한다. Repository Manager는 생성된 CAWL RE 문서를 CAWL RE 저장소에 저장한다.

2) CAWL 시나리오 문서의 생성 과정

CAWL 시나리오 문서는 Fig. 6에서 보이는 것과 같이 Workflow Model Manager, Workflow Converter,

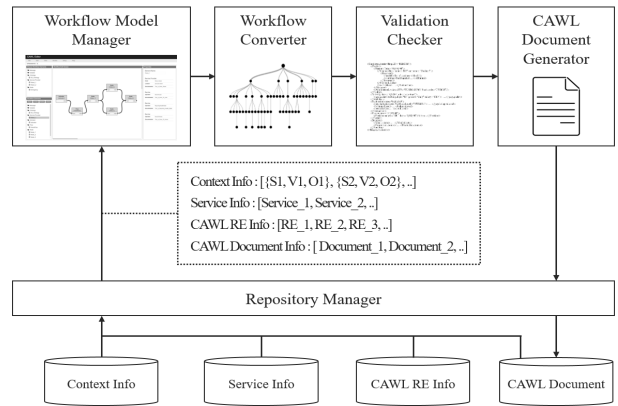


Fig. 6. Generation Flow of the CAWL Scenario Document

Validation Checker, CAWL Document Generator가 동작하여 생성된다.

Workflow Model Manager는 CAWL 시나리오 문서에 대한 작업을 요청받게 되면 다이어그램 기반의 상황인지 워크플로우 편집 화면을 제공한다. 이때 Repository Manager는 저장소에서 보유하고 있는 상황 정보, 서비스 정보, CAWL RE 정보, CAWL 시나리오 문서 정보를 Workflow Model Manager에 전달하여 시나리오 개발자가 상황인지 워크플로우를 손쉽게 모델링할 수 있도록 지원한다. 추가적으로 Workflow Model Manager는 시나리오 개발자가 재사용하는 CAWL RE 문서의 세부 속성 정보들을 변경할 수 있도록 수정 기능을 제공한다. Workflow Converter는 CAWL 기반의 트리 자료 구조를 이용하여 시나리오 개발자의 입력 정보를 텍스트 데이터로 변환한다. Validation Checker는 CAWL 스키마의 구조를 따라 변환된 텍스트 데이터에 대한 유효성 검증 작업을 수행한다. CAWL Document Generator는 텍스트 데이터를 바탕으로 XML 기반의 CAWL 시나리오 문서를 생성한다. Repository Manager는 생성된 CAWL 시나리오 문서를 CAWL Document 저장소에 저장한다.

4. 실험

본 장에서는 상황인지 워크플로우 모델에서 재사용 요소의 재사용성을 입증하기 위해 스마트 홈 환경의 서비스 시나리오를 구상하고, 이를 기반으로 재사용 요소의 적용 및 시나리오 문서 작성 과정을 보인다. 서비스 시나리오 예제는 두 개의 시나리오의 재사용이 가능한 여러 중복 요소들중 하나의 요소를 중점으로 전반적인 과정을 기술한다. 마지막으로 두 시나리오의 재사용 요소를 생성하고 활용함으로써 서비스 시나리오 개발자의 작업 비용을 단축시킬 수 있음을 서술한다.

4.1 스마트 홈 서비스 시나리오

본 절에서는 사용자 A에게 홈 케어 로봇을 통해 아침/저녁 케어 서비스를 제공하는 시나리오를 소개한다. 구체적인 시나리오는 다음과 같다.

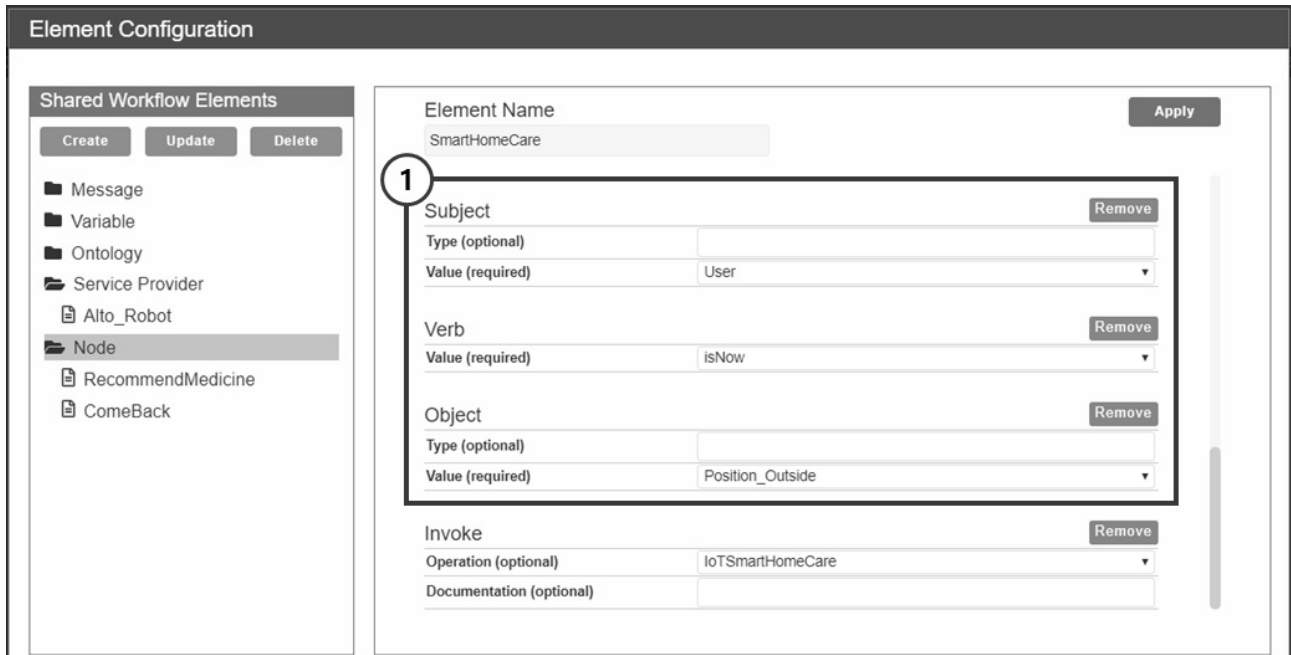


Fig. 7. CAWL RE Document Editing Screen

· 시나리오 1 - 모닝 케어 서비스

- ① 사용자 A는 매일 오전 7시에 기상하여 출근 준비를 시작한다. 로봇은 시간에 맞춰 사용자에게 이동한다.
- ② 사용자 A가 주방으로 이동하면 로봇은 오늘 섭취가 필요한 약 또는 식단을 추천한다.
- ③ 로봇은 오늘 날씨 정보를 사용자에게 레포팅한다.
- ④ 사용자 A가 출근을 하면 소등 서비스를 실행한다.

· 시나리오 2 - 굿나잇 케어 서비스

- ① 사용자 A가 퇴근하여 집에 도착하면 에어 케어 서비스와 점등 서비스를 실행한다.
- ② 로봇은 사용자 A의 건강 상태를 분석하고 현재 건강 상태를 레포팅한다.
- ③ 건강 상태를 레포팅한 후, 로봇은 사용자의 상태에 따라 헬스케어 정보를 제공한다.
- ④ 사용자가 취침 준비를 마치면 소등 서비스와 날씨 레포팅 서비스를 제공한다.

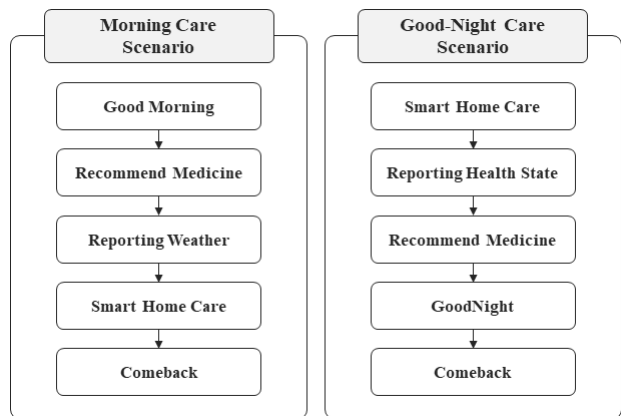


Fig. 8. Smart Home Service Scenario

Fig. 8은 시나리오 1, 2를 바탕으로 서비스 플로우 형태로 표현한 것이다. 두 시나리오는 동일한 서비스 에이전트를 대상으로 하여 서비스를 제공한다. 또한 구성 노드중 두 개의 노드(RecommendMedicine, Comeback)가 동일하고 또 하나의 노드는 같은 형태(SmartHomeCare)를 가진다. 이에 4.2에서는 재사용될 수 있는 요소중 Smart Home Care를 생성하고 편집하여 활용되는 과정을 보인다.

서비스 시나리오 개발자가 사용하는 시나리오 문서 편집기는 Node.js Express Web Framework를 기반으로 개발한 웹 애플리케이션이다. Node.js는 확장성 있는 네트워크 애플

리케이션 개발에 사용되며, Non-blocking I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능을 특징으로 갖는다. 재사용을 위한 CAWL RE 문서는 MySQL을 통해 효율적으로 관리된다. MySQL은 계층적인 요소 정보들을 테이블 단위로 관리할 수 있도록 지원하고, 표준 데이터베이스 질의 언어인 SQL (Structured Query Language)을 사용하여 작업 요청에 대한 처리를 빠르고 유연하게 수행할 수 있다.

4.2 재사용 요소의 적용 및 시나리오 문서 작성

본 논문에서 제안하는 재사용 요소는 XML 객체 단위이며, 본 절에서는 재사용 요소를 적용하여 서비스 시나리오 문서를 작성하는 과정과 더불어, 이 과정을 통해 기존의 방법보다 개발 비용이 개선되었음을 보인다. 재사용 요소를 사용하는 주체인 서비스 시나리오 개발자는 서비스 시나리오 문서 편

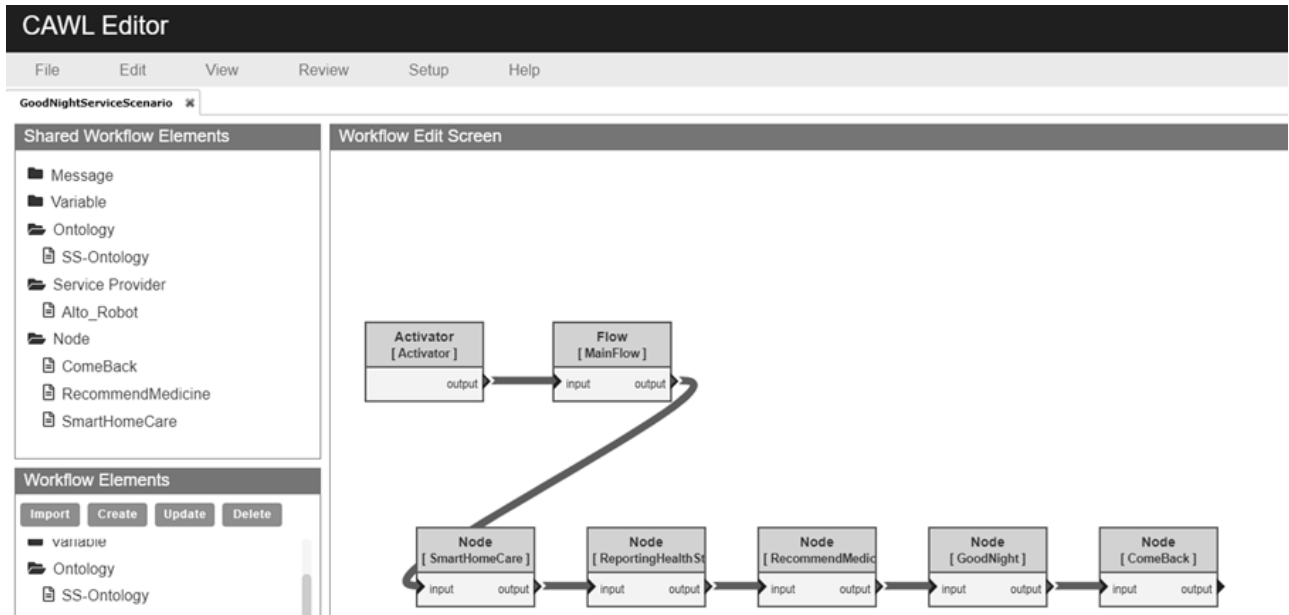


Fig. 9. Workflow Diagram Screen in CAWL Document Editor

Node Element Table			
node_id	Name	state	Documentation
SmartHomeCare	SmartHomeCare	deactivate	Smart Home ...

Condition Element Table			
condition_id	node_id	expression	documentation
SmartHomeCare_0	SmartHomeCare	C4	

Context Element Table				
context_id	condition_id	name	priority	documentation
SmartHomeCare_0_0	SmartHomeCare_0			

Rule Element Table			
rule_id	context_id	name	expression
SmartHomeCare_0_0_0	SmartHomeCare_0_0		

Fig. 10. Example of Node Element Storing

집기에서 제공하는 GUI 기반의 편집 화면을 통해 상황인지 워크플로우를 모델링 한다. 이 때, 동일한 서비스 도메인에서 다수의 서비스 시나리오를 개발할 경우 상황인지 워크플로우 모델의 요소 정보들은 중복될 수 있다.

그렇기에 서비스 시나리오 개발자는 중복되는 요소들을 상황인지 워크플로우 모델에 추가하기 위해 반복적인 작업을 수행하게 된다. 이와 같은 불편한 작업을 해소하기 위해 시나리오 개발자는 Fig. 7과 같이 재사용 요소들을 제공되는 편집 화면을 이용하여 CAWL RE 문서를 생성하거나 삭제할 수 있고, 기존 CAWL RE 문서를 호출하여 수정할 수 있다.

생성된 CAWL RE는 관계형 데이터베이스에 트리 구조로 저장된다. CAWL 스키마에 따라 <node>-<condition>-<context>-<rule>-<constraint>순으로 계층적 구조로 관리되며 시나리오1의 SmartHomeCare node는 Fig. 10과 같이 값을 갖는다.

서비스 시나리오 문서 편집기가 관리하는 CAWL RE 문서는 서비스 시나리오 개발 과정에서 단위 문서의 호출을 통해

Fig. 11. Screen to Change Property Information of Reusing CAWL RE Document

재사용할 수 있다. 상황인지 워크플로우 서비스가 실행되는 동적인 서비스 환경에서는 변화하는 환경에 따라 각 요소의 속성 정보들을 적절히 수정할 수 있어야 한다. 이에 시나리오 개발자는 재사용 과정에서 RE의 상세 파라미터 값을 수정하여 구성에 맞도록 변경할 수 있다.

Fig. 11은 시나리오2에서 SmartHomeCare node의 상황 정보가 변경되어 재사용 과정에서 속성 값을 수정하는 화면으로, Fig. 7과 Fig. 11의 1번 영역의 값이 다른 것을 볼 수 있다. 서비스 시나리오 문서 편집기를 RE를 수정하여 적용된 워크플로우의 시각화된 모습은 Fig. 9와 같고 이는 최종적으로 문서로 작성되어 사용된다. 서비스 시나리오 문서 편

```

<?xml version="1.0" encoding="UTF-8"?>
<CAWL name="GoodNightServiceScenario" ...>
  <documentation>
    IoT Good Night Service Scenario Document
  </documentation>
  <baseOntologies>
    ...
  </baseOntologies>
  <serviceProvider name="Alto_Robot"
    location="...">
    <service operation="IoTGoodMorning">
      </service>
    ...
  </serviceProvider>
  <activator name="Activator">
    <condition expression="C0">
      <context>
        <rule>
          <constraint name="C0">
            <subject>Alto_Robot</subject>
            <verb>isNow</verb>
            <object>Position_HOME</object>
          </constraint>
        </rule>
      </context>
    </condition>
    <activate flow="MainFlow">
      </activate>
    </activator>
    <flow name="MainFlow">
      <source name="MainSource">
        </source>
      <node name="SmartHomeCare"
        state="deActivate">
        <documentation>
          Smart Home Care Service
        </documentation>
        <condition expression="C1">
          <context>
            <rule>
              <constraint name="C1">
                <subject>User</subject>
                <verb>isNow</verb>
                <object>Position_Inside</object>
              </constraint>
            ...
          </node>
          ...
        </flow>

```

Fig. 12. Service Scenario Document based on CAWL Document Editor

집기를 통해 생성된 굿나잇 케어 서비스 시나리오 문서는 CAWL 스키마 기반으로 Fig. 12와 같이 표현된다.

실험에서 제시된 두 개의 시나리오는 본 논문에서 정의한 CAWL 재사용 단위를 기준으로 4개의 중복되는 요소를 갖는다. 이러한 시나리오는 논문에서 정의한 RE 구성요소를 7개

포함하고 있으며 이중 4개가 재사용됨으로써 57%의 재사용율을 보인다. 그리고 시나리오는 XML 기반으로 작성되어 전체 188개의 라인으로 작성되었으며, 141개의 라인은 7개의 RE의 정보를 나타낸다. 이중 69개의 라인이 활용되어 49%가 재사용된다. 결과적으로 본 실험에서 구상한 Smart Home 서비스 시나리오는 평균적으로 절반 이상의 재사용율을 보인다. 실험환경에서 적용되는 시나리오의 구성에 따라 재사용성은 편차가 있겠으나, RE를 적용하고 시나리오 편집기를 활용한다면 서비스 시나리오 개발자는 기존보다 서비스 시나리오 문서 작성에 소요되는 시간을 반복되는 요소만큼 줄일 수 있을 것이다. 또한 다른 사용자에게 동일한 서비스를 제공해야 할 경우 더욱 많은 반복 요소가 발생하여 작업 비용을 단축시킬 수 있을 것이다.

5. 결 론

본 논문에서는 서비스 시나리오 개발의 프로세스를 개선하기 위한 상황인지 워크플로우 모델에서 XML 객체를 재사용하는 방법을 제안하였다. 제안하는 방법은 상황인지 워크플로우 모델을 구성하는 독립적인 요소들인 XML 객체의 요소들을 RE라 정의하고 RE 단위의 문서로 생성 및 관리하고, 서비스 시나리오 개발 과정에서 저장된 RE 문서를 호출하여 재사용하는 것이다. 또한 재사용하는 요소의 속성 값을 적절히 수정하여 변경된 서비스 환경에 적용될 수 있도록 지원한다. 본 논문에서 제안하는 방법을 통해 서비스 시나리오 작성 시간이 단축됨을 보이기 위해 실험에서는 특정 사용자에게 홈케어 서비스를 제공하는 시나리오 예제를 구성하였으며 이를 통해 작업 시간이 단축될 수 있음을 보였다.

따라서 재사용하여 시나리오 개발자는 반복적으로 사용될 수 있는 요소들을 사전에 CAWL RE 문서로 생성한 후, 서비스 시나리오의 유형에 따라 중복되는 요소들에 대한 불필요한 반복 작업을 최소화하여 서비스 시나리오 개발 시간을 단축시킬 수 있다.

추가적으로, 지속적인 연구를 진행한다면 서비스 시나리오가 사용되고 다수의 상황인지 서비스가 제공되고 사용자부터 피드백 정보가 대량으로 적재가 될 수 있을 것이다. 이러한 적재된 정보는 인공지능 기술에 접목하여 사용자의 개인 맞춤형 서비스를 제공하기 위한 방안으로 활용될 수 있을 것이다. 또한 워크플로우 모델의 중복 요소들을 서비스 시나리오 개발 과정에서 추천한다면 시나리오 개발자는 편리하게 상황인지 워크플로우 모델을 재사용할 수 있을 것으로 기대된다.

References

- [1] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications*

- Surveys & Tutorials*, Vol.16, No.1, pp.414-454, 2014.
- [2] Forkan, Abdur, Ibrahim Khalil and Zahir Tari, "CoCaMAAL: A cloud-oriented context-aware middleware in ambient assisted living," *Future Generation Computer Systems*, Vol.35, pp.114-127, 2014.
- [3] Jiafu Wan, Shenglong Tang, Qingsong Hua, Di Li, Chengliang Liu, and Jaime Lloret, "Context-aware cloud robotics for material handling in cognitive industrial internet of things," *IEEE Internet of Things Journal*, Vol.5, No.4, pp.2272-2281, 2018.
- [4] Adnan Akbar, Francois Carrez, Klaus Moessner, Juan Sancho, and Juan Rico, "Context-aware stream processing for distributed iot applications," *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, IEEE, 2015.
- [5] Barker, Adam, and Jano Van Hemert, "Scientific workflow: A survey and research directions," *International Conference on Parallel Processing and Applied Mathematics*, Springer, Berlin, Heidelberg, pp.746-753, 2007.
- [6] Pascal Hirmer, Matthias Wieland, Holger Schwarz, Bernhard Mitschang, Uwe Breitenbucher, Santiago Gomez Saez, and Frank Leymann, "Situation recognition and handling based on executing situation templates and situation-aware workflows," *Computing*, Vol.99, No.2, pp.163-181, 2017.
- [7] Wang, Pengfei, Huifang Li, and Baihai Zhang, "A context-aware workflow framework and modeling language," *Sensors & Transducers*, Vol.175, No.7, pp.198-206, 2014.
- [8] Daniel Garijo, Oscar Corcho, Yolanda Gil, Meredith N. Braskie, Derrek Hibar, Xue Hua, Neda Jahanshad, Paul Thompson, and Arthur W. Toga, "Workflow reuse in practice: A study of neuroimaging pipeline users," *2014 IEEE 10th International Conference on e-Science*, IEEE, Vol.1, pp.239-246, 2014.
- [9] Cuadrado, Jesús Sánchez, Esther Guerra, and Juan de Lara, "A component model for model transformations," *IEEE Transactions on Software Engineering*, Vol.40, No.11, pp. 1042-1060, 2014.
- [10] Antonio Bucchiarone, Annapaola Marconi, Marco Pistore, and Heorhi Raik, "A context-aware framework for dynamic composition of process fragments in the internet of services," *Journal of Internet Services and Applications*, Vol.8, No.1, p.6, 2017.
- [11] Yao, Wen and Akhil Kumar, "CONFlesFlow: Integrating flexible clinical pathways into clinical decision support systems using context and rules," *Decision Support Systems*, Vol.55, No.2, pp.499-515, 2013.
- [12] Zhang, Xipu, Choonhwa Lee, and Sumi Helal, "iPOJO Flow: A declarative service workflow architecture for ubiquitous cloud applications," *Journal of Ambient Intelligence and Humanized Computing*, Vol.10, No.4, pp.1483-1494, 2019.
- [13] Zhiyong Liu, Shaokun Fan, Harry Jiannan Wang, and J. LeonZhaod, "Enabling effective workflow model reuse: A data-centric approach," *Decision Support Systems*, Vol.93, pp.11-25, 2017.
- [14] Saida Boukhedouma, Mourad Oussalah, Zaia Alimazighi and Dalila Tamzalit, "Service based cooperation patterns to support flexible inter-organizational workflows," *International Journal Information Technology and Computer Science*, Vol.6, No.4, pp.1-18, 2014.
- [15] Bousalem, Zakaria and Ilias Cherti, "XMap: A novel approach to store and retrieve xml document in relational databases," *Journal of Software*, Vol.10, No.12, pp.1389-1401, 2015.
- [16] Choi, Jong-Sun, Yong-Yun Cho, and Jae-Young Choi, "The design of a context-aware workflow language for supporting multiple workflows," *Journal of Internet Computing and Services*, Vol.10, No.6, pp.145-157, 2009.
- [17] Marcello La Rosa, Petia Wohed, Jan Mendling, Arthur H. M. ter Hofstede, Hajo A. Reijers, and Wil M. P. van der Aalst, "Managing process model complexity via abstract syntax modifications," *IEEE Transactions on Industrial Informatics*, Vol.7, No.4, pp. 614-629, 2011.
- [18] Saida Boukhedouma, Mourad Oussalah, Zaia Alimazighi, and Dalila Tamzalit, "Service based cooperation patterns to support flexible inter-organizational workflows," *Information Technology and Computer Science*, Vol.6, No.4, pp.1-18, 2014.

유 연 승



<https://orcid.org/0000-0001-7094-710X>
 e-mail : tmahtus@gmail.com
 2017년 백석대학교 소프트웨어학과(학사)
 2019년 숭실대학교 컴퓨터학과(석사)
 2019년~현 재 숭실대학교
 지능형로봇연구소 연구원

관심분야 : 로봇 미들웨어, 사물인터넷, 상황인지 워크플로우

문 종 혁



<https://orcid.org/0000-0002-4919-0058>
 e-mail : jonghyeokmun@soongsil.ac.kr
 2017년 한경대학교 컴퓨터공학과(학사)
 2019년 숭실대학교 컴퓨터학과(석사)
 2019년~현 재 숭실대학교 컴퓨터학과
 박사과정

관심분야 : 로봇 미들웨어, 빅데이터 처리, 인공지능 모델링



김도형

<https://orcid.org/0000-0002-8951-8679>
e-mail : dohyungkim@soongsil.ac.kr
2019년 한경대학교 컴퓨터공학과(학사)
2019년~현 재 송실대학교 컴퓨터학과
석사과정
관심분야: 클라우드 컴퓨팅, 빅데이터
처리, 인공지능 모델링



최종선

<https://orcid.org/0000-0001-9648-0667>
e-mail : jongsun.choi@ssu.ac.kr
2000년 송실대학교 컴퓨터학부(학사)
2002년 송실대학교 컴퓨터학부(석사)
2008년~2010년 유한대학교 e-비즈니스과
전임교원

2010년 송실대학교 컴퓨터학부(박사)
2011년~2012년 송실대학교 지능형로봇연구소 연구원
2012년~2013년 서일대학교 인터넷정보과 전임교원
2013년~현 재 송실대학교 컴퓨터학부 부교수
관심분야: 로봇 소프트웨어 플랫폼, 클라우드 컴퓨팅



최재영

<https://orcid.org/0000-0002-7321-9682>
e-mail : choi@ssu.ac.kr
1984년 서울대학교 제어계측공학과(학사)
1986년 미국 남가주대학교 전기공학과
(컴퓨터공학)(석사)
1991년 미국 코넬대학교 전기공학부
(컴퓨터공학)(박사)

1992년~1994년 미국 국립오크리지연구소 연구원
1994년~1995년 미국 테네시 주립대학교 연구교수
1995년~현 재 송실대학교 컴퓨터학부 교수
관심분야: 로봇 미들웨어, 시스템소프트웨어, 병렬/분산처리,
고성능컴퓨팅