

온톨로지를 이용한 이산 사건 시뮬레이션의 개념적 모델 구축 지원에 관한 연구

박지성 · 정성환 · 손미애[†]

성균관대학교 산업공학과

Ontology-based Conceptual Model Building Framework for Discrete Event Simulation

Jisung Park, Sunghwan Jeong, and Mye Sohn[†]

Department of Industrial Engineering, Sungkyunkwan University

Received 12 July, 2013; received in revised form 26 November, 2013; accepted 20 December, 2013

ABSTRACT

Conceptual Modeling is the process of abstracting a model from a real or proposed system. It is probably the most important aspect of a simulation study. Related works show that the elementary developers devoted little time to understanding how the systems actually worked, namely they didn't build appropriate conceptual model. Thus, the result of simulation is inconsistent because it depends on developer's competence. Although many researchers suggested various techniques enabling developer to build conceptual model, there were several limitations. In this study, to overcome the limitations of existing techniques, we proposed COMBINE-DES (Conceptual Model BuildINg framEwork using ontology for Discrete Event Simulation). The COMBINE-DES supports expediting the conceptual modeling with Solution ontology generated by Domain ontology and Simulation ontology. Moreover, it provides consistent simulation result regardless of repeated modeling.

Key Words: Computer model, Conceptual model, Discrete event simulation, Ontology

1. 서 론

시뮬레이션 프로젝트의 라이프 사이클은 개념적 모델링, 코딩을 통한 컴퓨터 모델 구현, 해당 모델의 실험을 통한 솔루션 획득 및 실제 문제 상황 개선 등의 과정으로 구성된다^[2]. 이 중 개념적 모델링은 시뮬레이션을 통해 해결하고자 하는 문제로부터 시뮬레이션 모델을 추출하는 과정으로서,

문제 상황의 이해, 모델링 목표의 설정, 모델의 입출력(input/output) 확인 및 모델 범위와 수준을 결정하는 단계를 거쳐 수행된다. 개념적 모델링은 시뮬레이션 프로젝트의 초기 단계에서 수행되나, 일회성 작업이 아니라 시뮬레이션 프로젝트의 전체 라이프사이클을 통틀어 지속적으로 반복되어야 하는 중요한 단계이다. 시뮬레이션 프로젝트의 라이프 사이클과 개념적 모델링간의 관계를 도식화하면 Fig. 1과 같다.

개념적 모델은 문제 상황과 모델링 목표에 적합한 시뮬레이션 모델의 개발과 시뮬레이션 실행을

[†]Corresponding Author, myesoehn@skku.edu
©2014 Society of CAD/CAM Engineers

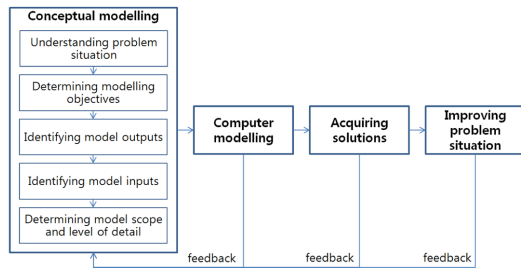


Fig. 1 Conceptual modeling in the simulation project life-cycle (revised from Robinson, 2004)

위한 로드맵으로서의 역할을 수행하기 위해 개발하며^[15], 동시에 클라이언트, 도메인 전문가 및 개발자 간의 중요한 커뮤니케이션 수단으로 활용하기 위해 구축한다^[14]. 또한 명확한 개념적 모델은 불명확하거나 일관성 없는 시뮬레이션 모델링 요소를 줄이는 데 기여하며, 이를 통해 시뮬레이션의 신뢰성 및 성공 가능성을 제고시키기 때문에 시뮬레이션 프로젝트에서 가장 중요한 측면으로 여겨진다^[3]. 그러나 시뮬레이션 개발자가 전문가가 아닌 초보자일 경우, 개념적 모델링의 중요성을 인식하지 못한 채, 개념적 모델링에 상대적으로 적은 시간과 노력을 할애하기도 한다^[32]. 즉, 충분한 사전 검토 없이 곧바로 컴퓨터 모델을 위한 자료 수집과 코딩 단계에 돌입함으로써, 모델의 범위를 제대로 설정하지 못해 적절한 데이터 분포를 찾지 못하거나 코딩의 어려움을 야기하기도 하며, 시뮬레이션 프로젝트 전체의 가이드라인 역할을 하는 개념적 모델의 반복적 수정으로 개발자와 클라이언트 간의 커뮤니케이션에도 어려움이 발생할 수 있다.

이러한 문제를 해결하기 위해, 개념적 모델링을 효과적으로 수행할 수 있는 방법과 도구들에 관한 연구가 수행되었으나, 기존 연구들은 다음과 같은 한계를 가지고 있다. 먼저, 개념적 모델링을 위한 이론적 프레임워크를 제안하는 연구의 경우^[22], 실제 개념적 모델링 과정의 자동화가 연구의 초점이 아니기 때문에 개념적 모델링에 소요되는 시뮬레이션 개발자의 모델링 부담은 여전히 존재한다. Generic template 기반 기법의 경우^[5,7,9-11,15,26,27], 자주 활용되는 모델링 대상과 문제 상황 등의 조합을 template으로 사전에 구축한 후, 이의 사용을 통해 모델링 부담을 경감시키고자 한다. 그러나 발생 가능한 모든 조합을 식별해 이를 template화 하

는 과정 자체가 매우 어려운 작업이다. Diagrammatic 기법의 경우^[8,12,19,20,24,25,29,31], 개념적 모델을 명료하게 ‘표현’하기 위한 방법으로서의 역할은 명확하나, 모델링 ‘과정’을 지원하기 위한 가이드라인으로서의 역할은 다소 제한적이다. 따라서 시뮬레이션 대상 프로세스와 관련된 데이터 수집의 번거로움이나 반복 작업으로 인한 개발자의 부담 등의 문제는 상존한다. 이로 인해 개발자의 역량에 따라 향후 시뮬레이션에 투입되는 정보의 양과 질에 차이가 발생하여 일관성이 떨어지는 시뮬레이션 결과를 도출하게 되는 문제로 이어질 수 있다.

이에 본 연구에서는 이러한 한계들을 보완할 수 있는 온톨로지 기반의 개념적 모델링 환경인 COMBINE-DES(Conceptual Model Building framework using ontology for Discrete Event Simulation)를 제안한다. COMBINE-DES가 개념적 모델링과 관련해 기여하는 바는 다음과 같다.

- 개념적 모델링 과정에서 발생하는 수작업의 부담 경감. 이론적 프레임워크에 COMBINE-DES가 활용한 온톨로지 개념을 접목함으로써 이론적 프레임워크의 완성도 제고
- 시뮬레이션 대상 및 문제 상황에 적합한 개념적 모델을 위해 온톨로지에 정의된 관계와 규칙(rule)을 활용해 추론. 시뮬레이션 대상 및 문제 상황들간의 다양한 조합을 모두 고려해 template을 사전에 정의해야 하는 template 기반 방법 보완
- 개념적 모델의 구축 과정에 대한 가이드라인과 소요 개념 및 어휘 간의 일관성 등을 온톨로지를 활용해 보장함으로써 diagrammatic 기법 보완

이상과 같은 장점을 기반으로, COMBINE-DES는 시뮬레이션 개발자의 부담을 경감시켜 줄 수 있는 자동화된 개념적 모델링 환경의 구축 가능성을 확인하고, 이러한 환경의 구축을 통해 시뮬레이션 개발자들에게 개발하고자 하는 시뮬레이션 모델링을 위한 대략적 가이드라인을 제공해 줌으로써, 효율적이고 효과적인 시뮬레이션 수행 기반을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 시뮬레이션에서의 개념적 모델링 및 온톨로지와 관련된 기존 연구를 살펴보고 3장에서는 COMBINE-

DES의 전체 프레임워크를 제안하고 세 가지 온톨로지의 구축 과정, 세부 구성 및 활용방법에 대해 설명한다. 4장에서는 자동차 엔진 조립공정을 예시로 사례연구를 수행하며 제안한 프레임워크를 대상으로 평가를 5장에서 수행한 후, 6장에서 본 논문의 결론과 향후 연구방향을 요약하였다.

2. 관련 연구

2.1 개념적 모델링기법

개념적 모델은 문제 상황 및 모델링 목표로부터 시뮬레이션 모델을 설계하고 소프트웨어 실행에 이르기까지의 로드맵 역할을 하기 위해 설계하며^[15], 컴퓨터모델 개발을 위한 가이드라인이자, 클라이언트, 도메인 전문가 및 개발자 간의 중요한 커뮤니케이션 수단이다^[14]. 또한, 불완전 또는 불명확하거나 일관성 없는 시뮬레이션 모델링 요소들을 최소화시킴으로써 시뮬레이션의 신뢰성 및 성공 가능성을 향상시키기 때문에 시뮬레이션 프로젝트에서 가장 중요한 측면으로 여겨진다^[3]. 이러한 중요성에 입각하여, 개념적 모델링의 효과적 구축을 지원하기 위한 다양한 연구들이 수행되었다. 해당 연구들을 세 가지 유형으로 분류하고, 각각의 정의 및 특징, 장점 및 단점을 요약한 결과가 Table 1과 같다.

이와 같은 한계들을 극복하고자 본 논문에서는

온톨로지 기반의 개념적 모델링 방법을 제안하고자 한다. 실제로, 시뮬레이션 분야에 온톨로지를 활용하는 연구들은 이미 여러 연구에서 진행된 바 있다. 2.2절에서는 온톨로지에 관련된 기존 연구들을 살펴본다.

2.2 시뮬레이션에서의 온톨로지 활용

온톨로지는 분산되어 있는 이질적인 정보들을 하나의 구체화된 도메인으로 구성하여 개념과 개념 사이의 관계를 기술하는 정형화된 어휘의 집합으로써, 공유된 개념화의 정형화되고 명시적인 명세를 표현한다^[4,8,29]. 온톨로지는 지식을 공유하고 재사용하기 위해 구축되어 활용되며, 이미 전자상거래(E-Commerce), 의료(Medical), 공학(Engineering), 화학(Chemistry) 및 지식경영(Knowledge Management) 등과 같은 다양한 분야에서 활용되고 있다^[1]. 시뮬레이션 분야도 마찬가지로 온톨로지를 접목시킨 다양한 연구가 진행된 바 있다. 하지만 첫째, 분산형 시뮬레이션 모델링 프로세스 전반에 걸친 온톨로지의 역할을 이론적으로 잘 설명하고 있지만, 실제로 구현하는 방법에 대한 설명이 없다는 한계가 있다^[24]. 둘째, 서로 다른 출처와 형식의 데이터를 변환하여 제조분야 시뮬레이션에 활용하는데 온톨로지 개념을 이용했으나^[6], 각 사례별 XML 문서 등을 온톨로지로 변환하고 해당 온톨로지의 추론에 필요한 규칙(rule)을 개별

Table 1 Comparison of conceptual modeling techniques

Method	Definition & Features	Pros vs. Cons	Related research
Theoretical framework	- Defining the conceptual modelling framework and the BP of each steps	<p><Pros></p> <ul style="list-style-type: none"> - Easy understanding of the definition and sequences of conceptual model <p><Cons></p> <ul style="list-style-type: none"> - Time consuming 	[22]
Generic & Template based	- Pre-constructing the module of model for the common simulation situation	<p><Pros></p> <ul style="list-style-type: none"> - Rapid construction of model parameter <p><Cons></p> <ul style="list-style-type: none"> - Need pre-definition of various combinations of model objects 	[5],[7],[9],[10],[11],[15],[26],[27]
Diagrammatic Technique	- Visualizing conceptual model using diagram	<p><Pros></p> <ul style="list-style-type: none"> - Useful communication tool for developer and client <p><Cons></p> <ul style="list-style-type: none"> - only focused on the expression of modeling result, not procedure 	[8],[12],[19],[20],[24],[25],[29],[31]

문제별로 정의해야 하는 부담이 있다^[18]. 즉, 기존 연구들에서 온톨로지를 사용한 주 목적이 모델링에 필요한 도메인 지식의 재사용이나 공유를 위한 것임을 알 수 있다. 그러나 시뮬레이션 모델의 구축을 위해서는 도메인 지식의 재사용 및 공유를 위한 역할뿐만 아니라, 개발자의 모델링 부담을 줄이기 위한 역할로 온톨로지를 활용할 필요가 있다. 따라서 시뮬레이션 모델을 구축하는데 필요한 모든 정보를 통합적으로 관리하는 시뮬레이션 온톨로지가 이러한 점을 해결해 줄 수 있다. 실제로, 이산 사건 시뮬레이션 관련 요소들을 정의한 DES 온톨로지를 개발하고, 이를 특정 온톨로지와 매핑하여 실제 시뮬레이션 소프트웨어에서 실행 가능한 모델로 변환하는 방법을 제안한 연구가 있다^[9]. 그러나 이 연구는 JSim이나 Arena와 같은 실제 소프트웨어에서 실행 가능한 모델의 생성이 최종 목표이기 때문에, 상세한 개념적 모델링의 구축을 전제로 한다. 이처럼, 시뮬레이션 분야에서 온톨로지를 활용하는 여러 가지 방법들을 제안하고는 있으나 개념적 모델링에 초점을 맞추는 경우가 적었고, 혹은 이론적 제안에 그치는 경우가 많아 실제 개념적 모델링 과정에서 온톨로지를 이용하는 실증적인 연구가 필요한 상황이다.

이에 본 연구에서는 특정 도메인 지식의 재사용 및 공유를 위한 도메인 온톨로지를 활용하는 한편, 이산 사건 시뮬레이션과 관련된 정보를 통합적으로 정의한 시뮬레이션 온톨로지를 별도로 구축 및 활용한다. 또한 이 두 가지 온톨로지를 활용하여, 해결하고자 하는 특정 시뮬레이션 문제를 위한 솔루션 온톨로지를 구축하여 이를 개념적 모델로 활용함으로써 개발자의 모델링 부담을 경감시키고, 향후 일관성 있는 시뮬레이션 결과 도출에 기여한다.

3. COMBINE-DES의 전체 아키텍처

COMBINE-DES는 획득한 도메인 지식을 활용해 시뮬레이션을 위한 개념적 모델링을 지원하기 위해 제안되었다. COMBINE-DES는 요구사항 확인 모듈(Requirement Identification Module, RIM), 부분 온톨로지 생성 모듈(Partial Ontology Extraction Module, POEM) 및 솔루션 온톨로지 구축 모듈(Solution Ontology Generation Module, SOGM)로 구성되어 있고, 시뮬레이션 개발자는 COMBINE-

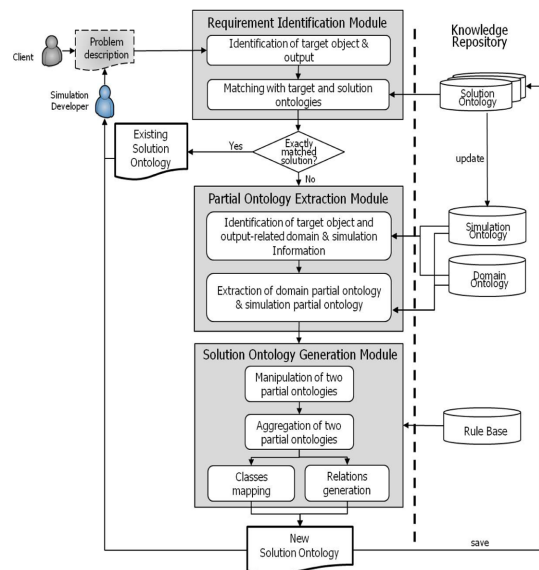


Fig. 2 Overall architecture of COMBINE-DES

DES가 제공하는 GUI를 이용해 개념적 모델링을 수행한다. COMBINE-DES의 구조를 도식화하면 Fig. 2와 같다.

3.1 요구사항 확인 모듈(RIM)

RIM은 시뮬레이션 개발자로부터 개념적 모델 구축을 위한 요구사항을 확인하는 모듈이다. 시뮬레이션의 개념적 모델링은 문제 상황의 인식에서 출발한다. 개발자는 클라이언트와의 커뮤니케이션을 통해 개선이 필요한 문제 상황을 이해하고 시뮬레이션 모델링의 목표를 세운다. 이후 해당 문제를 위한 시뮬레이션 모델링의 대상(object)과 산출물(output)을 결정한다. 개발자는 COMBINE-DES는 GUI를 이용해 결정된 대상과 산출물을 입력함으로써, 이를 활용해 COMBINE-DES는 개발자의 요구사항을 확인한다. RIM은 해당 대상과 산출물 조합에 따라, 기존 솔루션 온톨로지의 활용 여부를 결정하게 된다. 솔루션 온톨로지는 특정 대상과 산출물의 조합을 위한 개념적 모델로써, 유사한 조합의 시뮬레이션 소요가 발생할 경우 재활용을 하기 위해 지식저장소에 저장되어 있다. 만약 요구사항에 부합하는 솔루션 온톨로지가 존재할 경우 기존 솔루션 온톨로지를 활용하여 곧바로 개념적 모델을 개발자에게 제공하고, 존재하지 않을 경우 해당 요구사항에 맞는 새로운 솔루션 온톨로지 개발을 위해 다음 모듈인 부분 온톨로지

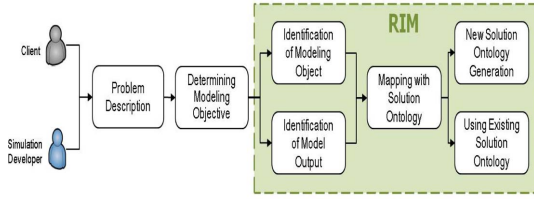


Fig. 3 Process of RIM

추출 모듈(POEM)로 이동하게 된다. RIM의 진행 과정을 절차적으로 나타내면 아래 Fig. 3과 같다.

3.2 부분 온톨로지 추출 모듈(POEM)

POEM은 RIM으로부터 확인된 개발자의 요구사항에 부합하는 정보로 구성된 부분 온톨로지들을 추출하는 모듈이다. 개발자가 선택한 두 가지 요구사항 중, 모델링 ‘대상’을 기준으로 해당 도메인 개체와 관련된 정보를 확인하고, ‘산출물’을 바탕으로 시뮬레이션 문제에 대한 표준 구성요소와 기타 세부사항을 확인한다. 이러한 정보는 COMBINE-DES의 지식 저장소(KR)에 있는 두 가지 온톨로지, 즉 도메인 온톨로지(domain ontology, DO)와 시뮬레이션 온톨로지(simulation ontology, SO)로부터 확인할 수 있다. DO는 대상 도메인과 관련된 정보를 정의하고 있는 온톨로지이고, SO는 시뮬레이션 모델링에 필요한 개념들을 정의하고 있는 온톨로지이다. 확인된 요소들에 부합하는 도메

```

 $I_s$ : Selected instances by developer
 $C_s$ : Set of classes of  $I_s$ 
 $P_s$ : Set of properties of  $C_s$ 
 $c_i$ : i-th class of  $C_s$ 
 $p_i$ : i-th property of  $P_s$ 
 $DO-ONT^{PART}$ : Domain partial ontology
 $SI-ONT^{PART}$ : Simulation partial ontology

Begin process
  for each class  $c_i$ 
    if  $c_i = domain(p_x)$ 
       $C_s \leftarrow C_s \cup range(p_x)$ 
    end if
  select  $P_s, I_s$  assigned to  $C_s$ 
  return  $P_s, I_s$ 
end for
Create PO( $C_s, P_s, I_s$ )
PO  $\leftarrow$  T
if  $C_s$  is classes of Domain Ontology
   $DO-ONT^{PART} \leftarrow$  PO
else
   $SI-ONT^{PART} \leftarrow$  PO
End process
    
```

Fig. 4 Partial ontology extraction algorithm of POEM

인 온톨로지와 시뮬레이션 온톨로지의 클래스들이 매핑되고, 이 클래스들을 토대로 부분 온톨로지가 추출된다. POEM의 부분 온톨로지 추출 과정을 알고리즘으로 표현하면 Fig. 4와 같다.

3.3 솔루션 온톨로지 구축 모듈(SOGM)

SOGM은 앞서 생성된 두 개의 부분 온톨로지를 통합함으로써 최종 솔루션 온톨로지를 구축하는 모듈이다. 통합에 앞서 개발자의 추가적인 요구사항을 확인하여 두 개의 부분 온톨로지를 수정한다. 수정을 통해 시뮬레이션 모델의 범위가 확대 혹은 축소되거나 기타 세부사항을 추가 혹은 제거할 수 있다. 수정사항이 반영된 후 두 개의 부분 온톨로지가 통합되고, 통합된 온톨로지 내 도메인 관련 클래스와 시뮬레이션 관련 클래스 간 매핑을 통해 관계가 생성된다. 도메인 관련 클래스와 시뮬레이션 관련 클래스 간 관계 설정을 위해 ‘Object Property’를 이용한다. 예를 들어, ‘ObjectProperty’인 ‘is_Entity’는 도메인 관련 클래스 중 시뮬레이

```

SO-ONTNEW: Solution ontology
DO-ONTPART: Domain partial ontology
SI-ONTPART: Simulation partial ontology
 $C_{DO}$ : Class of  $DO-ONT^{PART}$ 
 $C_{SI}$ : Class of  $SI-ONT^{PART}$ 
 $I_X$ : Instance Value of  $C_X$ 
mv: manipulated value by developer

Import  $DO-ONT^{PART}$ 
Import  $SI-ONT^{PART}$ 

Begin process
  for each class  $C_{DO}$ 
     $DO-ONT^{PART} \leftarrow DO-ONT^{PART} \cup Update\_INSTANCE(C_{DO})$ 
  end for
  for each class  $C_{SI}$ 
     $SI-ONT^{PART} \leftarrow SI-ONT^{PART} \cup Update\_INSTANCE(C_{SI})$ 
  end for
   $SO-ONT^{NEW} \leftarrow DO-ONT^{PART} \cup SI-ONT^{PART}$ 
End process

Function Update_INSTANCE( $C_X$ )
  for each  $C_X$ 
    if ( $I_X.NE.mv$ )
      Replace( $I_X \leftarrow mv$ )
    end if
  CreatePO(s)
  PO  $\leftarrow$  T
  end for
End function
    
```

Fig. 5 Solution ontology generation algorithm of SOGM

Table 2 SWRL description logic for class mapping

Rule Name	SWRL Description Logic
Class Mapping	$Part(?x) \wedge Process(?y) \wedge Human(?h) \wedge$ $Machine(?m) \wedge Entity(?e) \wedge Activity(?a)$ $\wedge Queue(?q) \wedge Resource(?r) \wedge$ $Has_Process(?x, ?y) \wedge Need_Machine(?y,$ $?m) \wedge Need_Human(?y, ?h) \wedge$ $is_Entity(?x, ?e) is_Activity(?y, ?a) \wedge$ $is_Queue(?m, ?q) \wedge is_Resource(?h, ?r)$

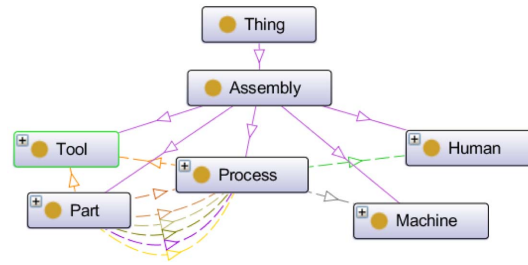
션 모델의 Entity로 할당될 대상을 정의역(*rdfs:domain*)으로, 시물레이션 모델의 구성요소인 Entity 클래스를 치역(*rdfs:range*)으로 가지게 된다. 이러한 매핑을 위해 SWRL(Semantic Web Rule Language) 기반의 추론이 수행된다. 이러한 과정을 통해 구축된 솔루션 온톨로지는 특정 시물레이션 문제에 대한 개념적 모델을 위한 맞춤형 온톨로지로서, 시물레이션 개발자에게 컴퓨터 모델 개발을 위한 가이드라인을 제공하는 역할을 하게 된다. 또한, 구축된 솔루션 온톨로지는 지식저장소(KR)에 저장되어 향후 유사한 시물레이션 문제를 모델링할 경우 재사용한다. SOGM에서의 솔루션 온톨로지 구축 과정을 알고리즘으로 표현하면 Fig. 5와 같고, 클래스 매핑 및 관계 생성을 위한 SWRL 규칙은 Table 2와 같다.

3.4 지식 저장소(KR)

KR은 본 프레임워크 진행에 필요한 지식인 도메인 온톨로지, 시물레이션 온톨로지, 솔루션 온톨로지 및 규칙베이스 등을 저장하고 있다. 개별 온톨로지에 대한 설명은 아래와 같다.

3.4.1 도메인 온톨로지

도메인 온톨로지는 특정 영역과 관련되어 있는 개념들과 이들 간의 관계를 정의한 온톨로지로서, 다양한 분야에서 구축되어 활용되고 있는 온톨로지임을 가정한다. 즉, 본 연구의 초점은 도메인 온톨로지를 구축하는 것이 아니라, 기존 도메인 온톨로지로부터 시물레이션 수행에 필요한 정보만으로 구성된 부분 온톨로지를 추출하는 방법을 제안하는 것이 목표이다. 본 연구에서는 ‘자동차 조립’에 대한 도메인 온톨로지를 대상으로 사례 연구를 수행했으며, 이외에도 다양한 온톨로지들이 COMBINE-DES에 적용될 수 있다. 본 연구의 사례 연구에서 활용한 도메인 온톨로지의 클래스

**Fig. 6** Main Classes of Domain Ontology

스 구성은 Fig. 6과 같다.

3.4.1.1 부품(Part)

General Assembly Ontology는 자동차 조립에 사용되는 여러 가지 부품들을 정의한다. 이 부품들은 자동차 조립 시물레이션 모델링의 대상으로 선택되는 주요 부품과, 주요 부품의 조립에 필요한 하위 부품 등이 모두 포함된다. 이들은 추후 시물레이션 모델에서 ‘Entity’로 매핑이 된다. COMBINE-DES가 제공하는 GUI를 통해 개발자가 선택한 시물레이션 대상 부품에 따라 해당 부품의 조립을 위한 공정의 유형과 순서가 추론되고, 해당 공정에 필요한 도구와 사람, 기계 정보도 함께 추론된다. 이러한 정보들이 POEM으로 전달되어 관련된 정보들로 이루어진 부분 온톨로지를 추출하게 된다.

3.4.1.2 공정(Process)

General Assembly Ontology는 특정 부품의 조립에 필요한 공정들을 정의하며, 이들은 향후 시물레이션 모델의 Activity로 매핑이 된다. 자동차 조립에 필요한 부품, 사람, 기계 및 도구 등은 모두 할당될 공정을 가지며, 이 공정에 따라 시물레이션 모델의 윤곽이 형성된다. 즉, 개발자가 시물레이션의 대상인 조립 부품을 선택하면 해당 부품 조립에 필요한 공정이 추론되고, 이 공정과 관련된 클래스인 기계, 도구 및 사람 등의 정보도 함께 추론된다.

3.4.1.3 기계(Machine), 도구(Tool), 사람(Human)

기계, 도구 및 사람 클래스는 각각 작업 공정의 운영에 필요한 기계, 도구 및 사람에 대한 정보를 담고 있다. General Assembly Ontology에서 개발자가 선택한 부품에 따라 해당 공정이 추론되고,

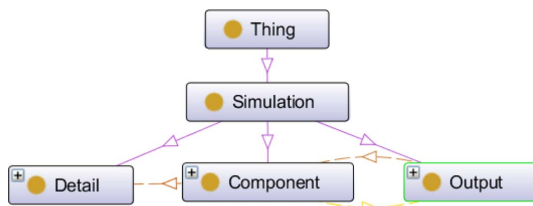


Fig. 7 Main Classes of Simulation Ontology

이 두 가지 사항에 필요한 기계, 도구 및 사람 정보도 함께 추론되어 개발자에게 추천된다.

3.4.2 시뮬레이션 온톨로지

시뮬레이션 온톨로지는 시뮬레이션 문제와 관련된 모든 요소들을 통합적으로 관리하는 온톨로지이다. 따라서 개발자가 요청하는 시뮬레이션 문제를 인식하고 해당 문제에 필요한 시뮬레이션 요소들을 부분 온톨로지로부터 추출하여 활용하게 된다. 본 연구에서는 이산 사건 시뮬레이션에서 일반적으로 활용되는 요소들을 시뮬레이션 온톨로지에 정의하였고, 해당 온톨로지의 메인 클래스 구성이 Fig. 7과 같다.

3.4.2.1 구성요소(Component)

일반적으로 시뮬레이션 모델은 entities, active states, dead states 및 resources 등 네 가지 유형의 요소로 구성된다^[17]. 이는 각각 entities, activities, queues 및 resource로도 나타낼 수 있으며^[23], COMBINE-DES의 시뮬레이션 온톨로지에서는 이 네 가지 구성요소를 Component 클래스의 하위 클래스 및 인스턴스들로 정의하였다. 이들은 새롭게 구축될 솔루션 온톨로지의 구성요소인 특정 도메인 개념들과 매핑되어 시뮬레이션 모델링에 활용된다.

3.4.2.2 산출물(Output)

시뮬레이션을 통해 해결할 문제를 인식하고 시뮬레이션의 목표를 설정하면, 개발자는 모델링 대상과 산출물을 결정하게 된다. COMBINE-DES는 이산 사건 시뮬레이션에서 주로 사용되는 산출물의 리스트를 시뮬레이션 온톨로지에 정의한 후, RIM에서 개발자가 GUI를 통해 원하는 산출물을 선택할 수 있도록 하였다. 또한, 개발자는 시뮬레이션 온톨로지에 정의되어 있지 않은 산출물도 추가하여 솔루션 온톨로지의 인스턴스로 정의할 수

있으며, 이와 같은 인스턴스들은 향후 시뮬레이션 온톨로지의 산출물 목록의 갱신에 활용된다.

3.4.3 솔루션 온톨로지

솔루션 온톨로지는 개발자가 원하는 특정 시뮬레이션 문제에 맞춰 최종적으로 구축될 목표 기반의 온톨로지이다. 도메인 온톨로지로부터 필요한 정보가 부분 온톨로지 형태로 추출되고, 해당 시뮬레이션 문제의 해결을 위해 필요한 요소들이 시뮬레이션 온톨로지로부터 부분 온톨로지 형태로 추출되면 두 온톨로지가 통합되어 최종적인 솔루션 온톨로지가 구축된다. 즉, 개발자는 기존 도메인 온톨로지, 시뮬레이션 온톨로지로부터 필요한 정보를 추출하여 통합한 솔루션 온톨로지를 활용함으로써 특정 시뮬레이션 문제를 위한 개념적 모델 가이드라인을 손쉽게 제공받을 수 있게 된다. 이렇게 구축된 솔루션 온톨로지는 지식저장소(KR)에 저장되어 향후 유사한 시뮬레이션 문제를 모델링할 경우 그대로 재사용할 수 있다. 특정 시뮬레이션 문제에 맞게 최종적으로 구축된 솔루션 온톨로지의 클래스 구성은 다음 4장에서 특정 시나리오에 적용시킨 결과로써 보일 것이다.

4. COMBINE-DES의 구현 및 적용

4.1 COMBINE-DES의 구현

자동차 조립 시뮬레이션의 개념적 모델 구축을 지원하는 COMBINE-DES의 구현을 위해, 온톨로지 에디터로 Protégé 4.1을 사용하였으며, 구축된 온톨로지의 추론을 위해 FaCT++ 엔진을, 일관성 평가를 위해 RacerPro1.9.0을 사용했다. 구축된 도메인 온톨로지와 솔루션 온톨로지로부터 부분 온톨로지를 추출하여 최종 솔루션 온톨로지를 구축하기 위해 Eclipse 소프트웨어에서 제공하는 자바 가상 머신(JVM)을 활용했으며, 개발자 GUI 설계를 위해 마찬가지로 JAVA와 Eclipse를 활용했다. COMBINE-DES의 GUI는 개발자가 모델링할 시뮬레이션의 대상과 산출물을 선택할 수 있게 하고 그에 맞는 솔루션 온톨로지를 검색 및 제공하는 기능, 제안된 시뮬레이션 구성요소와 세부사항을 개발자의 의도에 따라 커스터마이징 할 수 있는 기능, 개발자가 선택한 대상과 산출물의 조합과 수정사항을 고려하여 새로운 솔루션 온톨로지를 구축 및 제공하는 기능이 있다. 첫 번째 단계로, 개

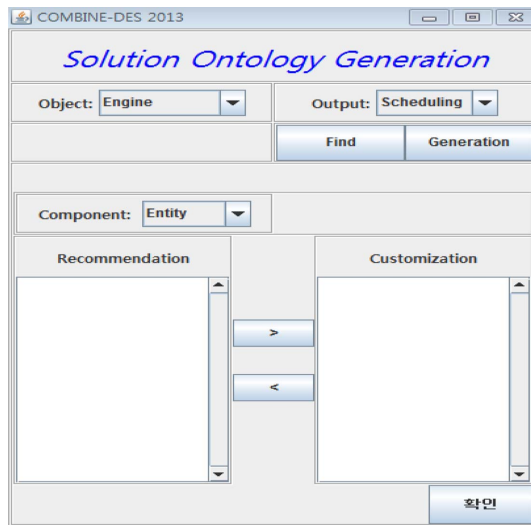


Fig. 8 GUI of COMBINE-DES

발자가 선택한 대상과 산출물 정보에 맞는 기존 솔루션 온톨로지 목록을 확인하고, 기존 솔루션 온톨로지가 있을 경우 기존 솔루션 온톨로지를 활용하여 개념적 모델을 제공한다. 만약 솔루션 온톨로지가 없을 경우, Generation 버튼을 통해 개발자가 선택한 대상과 산출물에 필요한 도메인 및 시뮬레이션 정보를 추출한다. 개발자는 추출된 정보를 본인이 계획하는 모델에 맞게 각 시뮬레이션 구성요소에 매핑시킨다. 이와 같은 과정을 지원하는 COMBINE-DES의 메인 GUI는 Fig. 8와 같다.

개발자의 계획에 따라 수정사항이 반영된 다음, COMBINE-DES는 개발자가 선택한 사항에 맞게 시뮬레이션 모델에서 고려해야 할 세부사항, 즉 Detail 항목들을 다시 추론하여 제공함으로써 최종적인 개념적 모델이 완성된다. 다음 절에서는 이러한 과정을 통해 구축된 솔루션 온톨로지와 개념적 모델 추출 결과를 Ford사의 엔진조립공장 건설 사례에 적용시켜 설명한다.

4.2 COMBINE-DES의 적용 시나리오

본 연구에서는 COMBINE-DES의 실효성 검증을 위해 Ford 자동차의 엔진조립공장 건설 사례를 이용했다^[21,22]. Ford 사는 South Wales에 새로운 엔진 조립공장 건설을 계획하고 있다. 본 사례에서 엔진의 조립공정은 ‘Head Line’, ‘Line A’, ‘Line B’, ‘Hot Test’, ‘Final Dress’와 같은 5개의 주요 공

정으로 구성되어 있다.

공장의 건설과 관련하여, 시뮬레이션 개발자는 엔진을 구성하는 핵심 구성부품들의 생산 및 보관과 관련된 Scheduling 문제, 엔진의 전체 조립 공정에 대한 Throughput 문제를 모델링하고자 한다.

- 모델링 1 (Scheduling): 먼저 Scheduling 문제의 경우, 조립공정을 통과하는 엔진의 ‘rate’보다 ‘sequence’가 핵심 이슈이며, 생산 스케줄, 테스트 및 재작업 구역, 병렬 머신과 같은 ‘sequence’ 결정요소를 시뮬레이션 모델에서 반드시 다뤄야 한다. 따라서 GUI를 통해 개발자가 엔진의 산출물로 ‘scheduling’을 선택할 경우 COMBINE-DES는 앞서 언급된 세 가지 결정요소를 고려하여 해당하는 클래스를 추론한 뒤 솔루션 온톨로지 클래스에 포함시킨다. 또한, scheduling 문제에서는 엔진을 구성하는 핵심 부품들의 생산 및 보관을 고려함으로써 이들을 적시적소에 제공이 가능하게 해야 하므로, 개발자가 선택한 시뮬레이션의 대상인 엔진뿐만 아니라 엔진을 구성하는 핵심 부품들도 함께 추론하여 솔루션 온톨로지 클래스에 포함시킨다.
- 모델링 2 (Throughput): Throughput 문제의 경우 설계된 생산 시설로부터 원하는 throughput을 달성할 수 있는지 결정하는 문제로, 세부 ‘operation’보다는 전체 ‘assembly line’의 throughput을 고려한다. 또한, 엔진의 구성부품들은 품질 없이 언제나 이용이 가능하다는 것을 전제로 하기 때문에 모델에서 따로 고려하지 않는다. 이처럼, 개발자가 얻고자 하는 시뮬레이션 모델의 대상과 산출물의 조합에 따라 그에 맞는 솔루션 온톨로지가 생성되어 정보가 추론되기 때문에 같은 문제 상황에서도 전혀 다른 개념적 모델의 구축이 가능하다.

또한, COMBINE-DES는 구축된 솔루션 온톨로지로부터 Component와 Detail(세부사항)을 추론하여 개발자에게 제공한 후, 추론된 정보를 개발자로 하여금 수정하도록 한다. 개발자는 자신이 구축하고자 하는 시뮬레이션 모델의 범위에 맞게 각 Component에 추천된 도메인 정보를 적절히 선택해 수정한다. 다음 단계에서 그 결과에 따라 추천된 각 세부사항에 대한 데이터 값이나 부가설명을 기록함으로써 최종적인 개념적 모델이 완성된다.

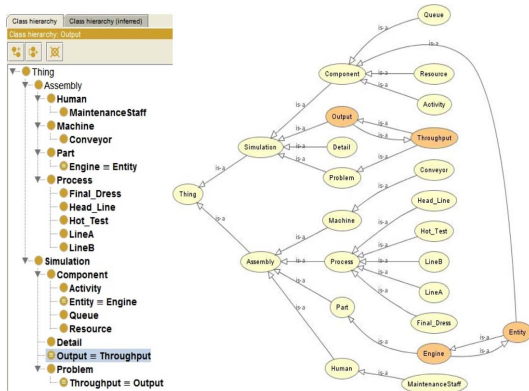


Fig. 9 Solution ontology for throughput problem

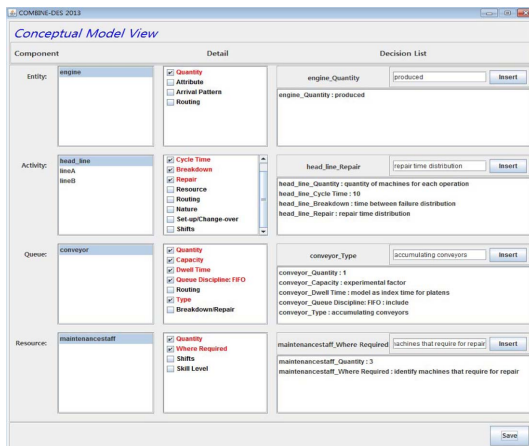


Fig. 10 Conceptual Model View for throughput problem

완성된 개념적 모델 정보 및 이를 위한 솔루션 온톨로지는 지식저장소에 저장함으로써 추후 모델링에 재사용될 수 있다. 구축된 솔루션 온톨로지는 Fig. 9와 같고, 이를 바탕으로 추출된 throughput 문제에 대한 개념적 모델 결과 예시 화면이 Fig. 10과 같다.

이상의 과정을 통해 구축된 개념적 모델을 바탕으로 개발자는 컴퓨터 모델 개발 및 시뮬레이션 SW를 이용한 모델링을 위한 가이드라인으로 활용할 수 있다.

5. 실험 및 평가

5.1 솔루션 온톨로지의 적합성 평가

본 연구에서는 솔루션 온톨로지를 구축하여, 이를 개념적 모델로 활용하였다. 따라서 구축된 솔

루션 온톨로지가 해당 시뮬레이션 문제를 위한 개념적 모델로서 적합한지의 여부를 평가하는 과정이 필요하다. [23]의 경우, Validity, Credibility, Utility, Feasibility 등의 4가지 조건에 부합하는지 평가함으로써 구축한 개념적 모델을 평가한 바 있다. 본 연구에서는 구축한 솔루션 온톨로지의 적합성 평가를 위해, RacerPro를 이용한 온톨로지의 일관성 검증을 수행했다. RacerPro는 추론 과정에서 생기는 온톨로지의 일관성, 개념의 만족성, 개념의 포함관계, 인스턴스 확인 및 결합 질의응답 등의 문제를 확인해주는 엔진으로서^[33], 수행결과 온톨로지의 적합성과 일관성을 위배하는 어떠한 증거도 나타나지 않았다.

5.2 COMBINE-DES의 성능 평가

COMBINE-DES의 우수성을 입증하기 위해, 수작업으로 개념적 모델링을 수행한 경우와, COMBINE-DES를 이용해 모델링한 경우를 비교해 개념적 모델링의 정확도를 분석하였다. 정확도란 구축된 개념적 모델이 실제 시뮬레이션 및 적용 대상을 얼마나 잘 반영하고 있는지를 평가하는 지표이다. 정확한 실험을 위해, 각 실험군에 속한 피험자들을 시뮬레이션 분야에 대한 지식이 충분한 전문가 그룹과 시뮬레이션 분야에 대한 지식이 충분하지 못한 초보자 그룹으로 구분하여 배정했다. 따라서 수작업으로 모델링하는 전문가(Expert-Manual) 3명, 수작업으로 모델링하는 초보자(Beginner-Manual) 3명, 그리고 COMBINE-DES를 이용해 모델링하는 전문가(Expert-COMBINE) 3명 및 COMBINE-DES를 이용해 모델링하는 초보자(Beginner-COMBINE) 3명 등 총 12명의 피험자가 실험에 참여했다. 모든 피험자들은 4.2절의 시나리오에 해당하는 문제 상황에 대한 자세한 설명을 듣고, ‘scheduling’ 문제와 ‘throughput’ 문제 각각에 대한 개념적 모델을 구축했다.

모델링 과정의 정확도 평가를 위해, ‘scheduling’ 문제와 ‘throughput’ 문제 각각에 대해 개념적 모델에서 반드시 고려되어야 할 사항이 제대로 포함되었는지 여부를 측정하였다. 즉, 각 문제에 반드시 포함되어야 할 구성요소 및 세부사항 리스트를 결정하고, 각 피험자들이 구축한 개념적 모델에 포함되지 않은 사항의 개수를 도출하였다. 포함되지 않은 사항의 개수가 적을수록 정확도 높은 모델링이 이루어졌음을 알 수 있다. 분석 결과, 수작업으

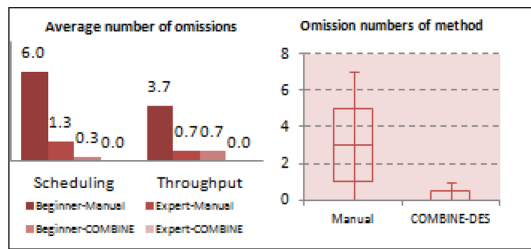


Fig. 11 Performance test results for object omission

로 모델링을 수행한 그룹의 경우 피험자의 역량에 따라 누락개수의 편차가 컸던 반면에, COMBINE-DES를 이용할 경우 피험자의 역량과 상관없이 모두 매우 정확도 높은 모델링을 수행함을 보여주었다. 각 피험자군의 문제별 누락사항 개수의 평균을 비교한 결과 및 모델링 방법별 누락사항 개수를 비교하여 Boxplot으로 나타낸 결과가 Fig. 11과 같다.

6. 결론 및 추후 연구 과제

본 연구에서는 기존 이산사건 시뮬레이션에서의 개념적 모델링의 부담을 경감시키는 데 기여할 수 있는 COMBINE-DES를 제안하였다. COMBINE-DES는 시뮬레이션 문제와 관련된 정보를 통합적으로 관리하는 세 가지 온톨로지를 개념적 모델링에 활용하였다. 시뮬레이션을 위한 개념적 모델링에 온톨로지를 접목함으로써 개념적 모델링과 관련되어 수행된 바 있는 기존 연구들이 가진 한계나 문제점을 보완할 수 있는 가능성을 확인하였다. 또한 COMBINE-DES는 시뮬레이션에 전문성이 떨어지는 초보 개발자들을 위한 개념적 모델링 지원 도구로서의 가능성을 실험을 통해 확인하였다. COMBINE-DES는 체계적인 개념 모델링 과정의 지원을 통해 시뮬레이션 모델을 위한 가이드라인을 제공함으로써 일관성 있는 시뮬레이션 결과를 도출하도록 해주고, 시뮬레이션 프로젝트의 신뢰성 및 성공 가능성을 높이는 데 기여할 수 있다.

그러나 본 연구는 몇 가지 한계와 그에 따른 추후 연구의 필요성을 지니고 있다. 첫째, 본 연구에서는 자동차 조립에 관련된 도메인 온톨로지 한 가지만 사용해 COMBINE-DES를 적용 및 구현했다. 따라서 그 외 다양한 도메인 온톨로지에도 추가 적용시켜볼 필요가 있다. 둘째, 본 연구에서는 구축된 솔루션 온톨로지를 바탕으로 GUI를 통

해 개발자에게 개념적 모델 정보를 제공한다. 그러나 UML, Flow chart, IDEF와 같은 기존의 Diagrammatic techniques와 결합하여 좀 더 시각화된 개념적 모델을 추출한다면 개발자가 효과적으로 이해하는데 도움이 될 것이다. 셋째, COMBINE-DES를 통해 구축된 개념적 모델은 단지 컴퓨터 모델 구축을 위한 가이드라인으로써, 시뮬레이션 소프트웨어에서의 모델링 과정을 지원해주지는 않는다. 따라서 Arena나 Simio와 같은 시뮬레이션 소프트웨어에서 바로 적용이 가능하도록 개념적 모델을 컴퓨터 모델로 변환시키는 기능까지 확장한다면 개발자의 작업 부담을 경감시킬 수 있을 것으로 기대된다.

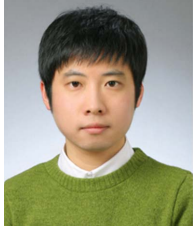
감사의 글

본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발사업의 일환으로 수행하였음(No. 10041788, 향상된 상황인지 기반 스마트 홈 서비스 기술 개발).

References

1. Asuncion Gomez-Perez, M. Fernandez-Lopez, and O. Corcho, 2004, *Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*, Springer.
2. Benjamin Perakath, M. Patki, and R. Mayer, 2006, Using Ontologies for Simulation Modeling, *Proceedings of the 38th Conference on Winter Simulation. Winter Simulation Conference*, pp.1151-1159.
3. Borah, J.J., 2002, Conceptual Modeling - The Missing Link of Simulation Development, *Proceedings of the 2002 Spring Simulation Interoperability Workshop*, Available online.
4. Borst, W.N., 1997, *Construction of Engineering Ontologies*, Centre for Telematica and Information Technology.
5. Brown Nancy and S. Powers, 2000, Simulation in a Box (a generic reusable maintenance model), *Proceedings of the 2000 Winter Simulation Conference*, pp.1050-1056.
6. Kim, D.-S. et al., 2011, Building and Applying Shipbuilding Ontology for BOM Data Interoperability in Heterogeneous Shipbuilding PLM Systems, *Transactions of Society of CAD/CAM Engineers*, 16(3), pp.197-206.

7. Diaz-Calderon, *et al.*, 2000, Architectures and Languages for Model Building and Reuse: Organization and Selection of Reconfigurable Models, *Proceedings of the 32nd Conference on Winter Simulation*, pp.386-393.
8. Gruber, T.R., 1993, *A Translation Approach to Portable Ontology Specification*, Knowledge Systems Laboratory Technical Report KSL 92-71
9. Hills, P.R., 1971, "HOCUS". PE Group: Egham, Surrey, UK.
10. Kasputis Stephen and Henry C.N., 2000, Model Composability: Formulating a Research Thrust: Composable Simulations, *Proceedings of the 32nd Conference on Winter Simulation*, pp.1577-1584.
11. Mackulak, G.T. and Cochran, J.K., 1990, The Generic-Specific Modeling Approach: An Application of Artificial Intelligence to Simulation, *Proceedings of the IIE Integrated Systems Conference*, pp.82-87.
12. Mackulak, G.T., Frederick, P.L. and Theron, C., 1998, Effective Simulation Model Reuse: A Case Study for AMHS Modeling, *Proceedings of the 30th Conference on Winter Simulation*, pp.979-984.
13. Nance, R.E. and Overstreet, C.M., 1987, Diagnostic Assistance Using Digraph Representations of Discrete Event Simulation Model Specifications, *Society for Computer Simulation, Transactions*, 4, pp.33-57.
14. Pace, D.K., 2002, The Value of a Quality Simulation Conceptual Model, *Model Simul Mag, Ver. 1(1)*, pp.9-10.
15. Pace, D.K., 2003, Thoughts About the Simulation Conceptual Model, *Proceedings of the 2003 Spring Simulation Interoperability Workshop*, Available online.
16. Page, E.H. and Opper, J.M., 1999, Observation on the Complexity of the Composable Simulation, *In Proceedings of the 1999 Winter Simulation Conference*, pp.553-560.
17. Pidd, M., 2004, *Computer Simulation in Management Science* (5th ed.), Wiley.
18. Rabe, Markus, and Avel Gocev, P., 2012, Applying Semantic Web Technologies for Efficient Preparation of Simulation Studies in Manufacturing, *Proceedings of the 2012 Winter Simulation Conference*, pp.1-12.
19. Silver, G.A., *et al.*, 2011, DeMO: an Ontology for Discrete-event Modeling and Simulation, *Simulation*, 87(9), pp.747-773.
20. Richter Hendrik and März, L., 2000, Toward a Standard Process: the Use of UML for Designing Simulation Models, *Proceedings of the 32nd Conference on Winter Simulation*, pp.394-398.
21. Robinson, S., 2004, *Simulation: the Practice of Model Development and Use*. Wiley.
22. Robinson, S., 2008, "Conceptual Modeling for Simulation Part1: Definition and Requirements", *Journal of the Operational Research Society*, 59, pp.278-290.
23. Robinson, S., 2008, Conceptual Modeling for Simulation Part2: A Framework for Conceptual Modelling, *Journal of the Operational Research Society*, 59, pp.291-304.
24. Ryan, J. and Heavey, C., 2006, Requirements Gathering for Simulation, *Proceedings of the Third Operational Research Society Simulation Workshop*, pp.175-184.
25. Schruben, L.W., 1983, Simulation Modeling with Event Graphs, *Communications of the ACM*, 26(11), pp.957-963.
26. Som, T.K. and Sargent, R.G., 1989, A Formal Development of Event Graphs as an Aid to Structured and Efficient Simulation Programs, *ORSA Journal on Computing*, 1(2), pp.107-125.
27. Son, Y.J., Albert T. Jones, and Richard A. Wysk, 2000, Automatic Generation of Simulation Models from Neutral Libraries: An Example, *Proceedings of Winter Simulation Conference*, pp.558-567.
28. Steele, M.J. *et al.*, 2002, Generic Simulation Models of Reusable Launch Vehicles, *Proceedings of the Winter Simulation Conference*, pp.747-753.
29. Studer, R., Benjamins, V.R. and Fensel, D., 1998, Knowledge Engineering: Principles and Method, *Data & knowledge engineering*, 25(1), pp.161-197.
30. Torn, A.A., 1981, Simulation Graphs: A General Tool for Modeling Simulation Designs, *Simulation*, 37(6), pp.187-194.
31. Van der Zee D.J., 2006, Building Communicative Models—A Job Oriented Approach to Manufacturing Simulation, *Proceedings of the Third Operational Research Society Simulation Workshop*, pp.185-194.
32. Wang, W. and Brooks, R.J., 2007, Improving the Understanding of Conceptual Modelling, *Journal of Simulation*, 1(3), pp.153-158.
33. W3C, 2006, OWL 1.1 Web Ontology Language Tractable Fragments.

**박 지 성**

2011년 성균관대학교 시스템경영
공학 학사
2013년 성균관대학교 산업공학과
석사
2013년 삼성 SDS
관심분야: Semantic Web, Ontology,
Web Service, Modeling and
Simulation

**정 성 환**

2012년 성균관대학교 시스템경영
공학 학사
2012년~현재 성균관대학교 산업공
학과 석박통합과정
관심분야: Semantic Web, Ontology,
Web Service, Linked Data,
Modeling and Simulation

**손 미 애**

1985년 성균관대학교 산업공학과
학사
1988년 KAIST 산업공학과 석사
2002년 KAIST 경영정보공학 박사
1988년~2004년 한국국방연구원
2004년~현재 성균관대학교 시스템
경영공학과 부교수
관심분야: Semantic Web, Ontology,
Linked Data, Internet of Things,
Web Service, Architecture
Framework, Modeling and
Simulation