Building A SMP2-based Operational Effectiveness Simulation Model Framework for UAV Systems

Ning Zhu
College of Information System and Management
National University of Defense Technology
Changsha, China
zhun870525@126.com

Yonglin Lei
College of Information System and Management
National University of Defense Technology
Changsha, China
samuelyonglin@gmail.com

Xiaobo Li
College of Information System and Management
National University of Defense Technology
Changsha, China
lixiaobo@nudt.edu.cn

Yifan Zhu
College of Information System and Management
National University of Defense Technology
Changsha, China
nudtzyf@163.com

Abstract—Operational effectiveness assessment is one of the most important warrants for UAV Systems demonstration and design, and simulation is a key method. Operational effectiveness simulation relies on large quantity of simulation applications, and composable modeling is an approach to support rapid development of simulation applications. Currently, it is more effective for composable modeling to proceed with simulation models libraries based on a standard simulation modeling specification and constrained by a uniform simulation model framework. However, present model frameworks own weak extensibility for lacking of standard model specifications and explicit platform independent descriptions. SMP2 is a simulation model specification and it utilizes catalogues in XML format to describe model frameworks. Ontology focuses on the representation of knowledge and facts, and it is very adaptive to collect and describe domain knowledge of UAV systems. So in this paper, an ontology model is designed by referring to current UAV systems related simulation systems and environments. And then a SMP2 based effectiveness simulation model framework for UAV systems is built by transforming from the ontology model.

Keywords-UAV Systems; effectiveness simulation; ontology; SMP2; model framework

I. INTRODUCTION

As a new type of military equipment, unmanned aerial vehicle (UAV) systems have played an important role in the several local wars. Considering of their predominant advantages, design, development and demonstration of UAV systems have become an emphasis of major military powers.

The effectiveness of UAV systems is their ability to complete specified tasks under provided situation. It determines the success or failure of design and product of UAV systems and has been one of the most warrants for demonstration of UAV systems in the military domain. Massive operational weapons, equipment and personnel are involved in assessment of operational effectiveness, with complex counteraction and quantities of uncertainties at the same time. Compared to traditional methods such as physical tests or analytic methods, simulation has become a key method for operational effectiveness assessment [1].

UAV systems can perform multiple activities such as Near-Real-Time (NRT) Surveillance, NRT Targeting and Precision Strike Support, Precision Strike, Battle Damage Assessment, Communication, Air-to-Air Refueling and Air-to-Air Combat [2]. Then assessments on operational effectiveness of UAV systems need to synthesize their performance in accomplishing different operational tasks. Therefore their operational effectiveness simulation relies on multiple simulation applications correspondent to their supporting activities. These simulation applications can be developed separately, but lots of simulation models such as detection models, missile models, and communication models can be reused. So the demonstration and design departments of UAV systems demand an environment to rapidly develop different kinds of simulation applications in a composable way. However, there are few simulation systems or environments supporting composable modeling so far. Composable modeling and simulation have been a hot research issue in system simulation, especially military simulation domain for many years [3, 4].

Currently there are four categories of methods for composable modeling of effectiveness simulation, unified simulation protocol, unified model specification, unified formalism and unified simulation environment. It's far not enough to realize composable modeling of effectiveness simulation by the first three categories of methods in the technological hierarchy [3, 5]. Taking those unified simulation environment for reference, it is a better way to solidify the domain invariant knowledge (DIK) [3] as a unified model framework which describes different kinds of simulation models in a simulation system and their relationship. And then models are developed according to a model specification and simulation environments are built under the constraint of this unified model framework.

However, present methods for developing simulation applications either lack a unified model framework, or are out of an authoritative modeling specification or platform independent and explicit description of model frameworks.

Ontology is originally a philosophical concept and now is pervasively used in the different domains of computer science such as domain engineering, artificial intelligence [6]. Ontology focuses on the representation of knowledge and facts, and can provide technology independent formal description of the knowledge in the problem domain, so it is very adaptive to collect and describe domain knowledge of UAV systems [5]. Web ontology language (OWL) is a representative ontology modeling language and has been adapted by World Wide Consortium (W3C) as a standard for ontology modeling [7].

Protégé [8] is an open-source ontology modeling tool to support OWL based ontology modeling. It coheres to OWL and is composed of four main views. In the class view, the concepts and their hierarchical relationships can be modeled. Properties of a concept should be described in the Data Property view, while relationships of concepts in the Object Property view. And the relationships in the Object Property use domain and range properties to show its direction. The Individual view composes of all the instances of the concepts [5].

Simulation model portability standards 2 (SMP2) is a simulation model specification proposed by Europe Space Agency (ESA) and supports simulation models to be composable, portable and reusable [9].

So in this paper, domain knowledge of UAV systems are captured from present related effectiveness simulation systems and environment are modeled as an ontology model of effectiveness simulation model framework. And then a transformation mechanism from ontology model to a SMP2 based simulation model framework is designed and implemented. Finally a SMP2-based operational effectiveness simulation model framework is built to support the operational simulation assessment of UAV systems.

II. RESEARCH BACKGROUND AND STATUS

A. Effectiveness Simulation Composable Modeling of UAV Systems

In the future, miniaturization, great-height, high-speed, stealth, far-distance, long-endurance, multipurpose will be the development trends of UAV systems. Their operations patterns will be extended to support attack, communication and space counteractions, even fight with other UAV systems [10]. So effectiveness simulation of UAV systems needs to be modularized to provide different operational capabilities. Simulation models need to be normalized to make them portable, composable and reusable, improving developing efficiency of simulation applications. However, the present four categories of methods for effectiveness simulation own disadvantages to support the effectiveness simulation assessment of UAV systems.

1) Unified simulation protocol

Components in the effectiveness simulation domain of UAV systems are massive and own complex relationships. However, methods of unified simulation protocol (such as High Level Architecture, HLA [11]) are difficult to develop, integrate and maintain simulation applications. And it's also complex to manage experiments because conceptual models based on HLA lack ample description ability and lots of intrinsic complexity are changed to pull-in ones and left in the implementation level [12]. At the same time, the execution efficiency is low while it is hard for model reuse to be in a composable way.

2) Unified model specification

Developers of effectiveness simulation applications are always experts to design and validation of UAV systems, so the methods need to be domain-friendly, easy to learn and use. But methods of unified model specification (such as Base Object Model (BOM) or SMP2) are parts of the technological specifications and not confined into a specified domain. In addition, it doesn't contain domain invariant knowledge, not to mention the specified design for a specified domain. So for the users, though there is tool support for simulation and modeling, it's still hard to develop simulation models and applications. What's more, developments of multiple simulation applications involve lots of replicated work, so it doesn't appeal to the users so much [13].

3) Unified formalism

Same as unified model specification, methods of unified formalism are also not to solidify domain invariant knowledge of a specified domain. What's more, it's hard to support a great number of behavior modes such as states and actions.

4) Unified simulation environment

There are two subcategories, simulation development environment and simulation application environment. The simulation development environments are not constraint in an application domain. Though some mature simulation development environments have provided mechanisms to encapsulate common components in a domain to let other applications to use such as block libraries in Simulink, and domain object libraries of Dymola, common components are maintained by the software providers and it's hard to provide a comprehensive component in a specified domain such as UAV systems.

Simulation application environments such as EADSim, JMASS provide a unified model framework for a specified domain and model libraries. So it can support the semantic composability of different models to support rapid development of simulation applications. However, though these model frameworks use the Object-Oriented and component based modeling thoughts, their models don't be described in an authoritative, unified and open model specification. And the model framework is also embedded in the codes and lacks an explicit and platform independent description, which makes them lack of extensibility.

B. Simulation Modeling Portability Standards 2

Simulation Modeling Portability Standards 2 (SMP2) is a simulation model specification standard proposed by ESA. The initial purpose of SMP is to provide probability of simulation models in different simulation platforms and applications. As thoughts of MDA are introduced, SMP2 adopts modern software architecture modeling technology such as Object-Oriented, Component Based Development, Design Pattern and Platform Specific Model generation.

This makes SMP2 neither own the representative ability compared to the other model specifications such as HLA/BOM to satisfy composable modeling requirement in all aspects, nor rely on the implementation technologies in the low level to avoid the implementation problems of HLA/BOM [9].

There are three parts in SMP2 specification, SMP2 meta-model, SMP2 component model and SMP2C++ mapping [9, 14]. SMP2 meta-model is also called Simulation Model Definition Language (SMDL). SMP2 models include Catalogue, Assembly and Schedule. Catalogues are composed of type information of models and fields and they describe simulation model frameworks in XML format. The meta-model of the catalogue is shown in Fig.1.

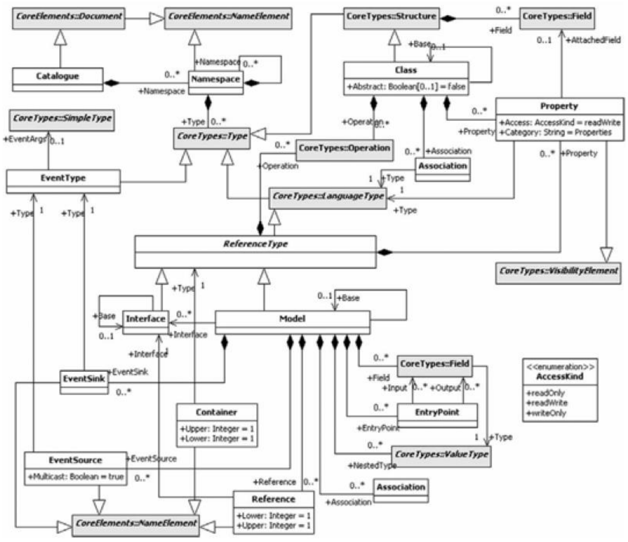

Figure 1. The meta-model of the catalogue in SMDL [14].

Catalogues define the model information of SMP2. Catalogues utilize the namespaces to classify models, and a namespace can include 0 or more namespaces and types. In the same namespace, the names of the models and types must be unique. There are mainly four types in a SMP2 model, Class, Model, Interface and Event Type. Only the instances of Models can be scheduled by the SMP2 simulator, while other types provide direct or indirect services for model instances.

C. UAV Systems Related Simulation Environment and Model Frameworks

1) OpenEaagels [15]

OpenEaagles (Open Extensible Architecture for the Analysis and Generation of Linked Simulation) is an open-source edition of Eaagles used in USA air force. It is published in 2006 and focus on the virtual and constructive simulation. Its architecture is shown in Fig.2.

OpenEaagles provides a simulation model framework for simulation application development using several C++ libraries such as the basic library for storage and manage physical entities and the simulation library for providing simulation models including UAV, warship, tank and so on.

Though OpenEaagels is a very powerful simulation framework, the simulation models are not based on a standard simulation modeling specification, but interface and inheritance in the code level. What's more, there is not an explicit and platform independent description of the model framework underling OpenEaagles, so users of the framework can only use inheritance, polymorphic or other programming mechanisms to expand the model framework in the code level. It's not so easy and feasible to extend.

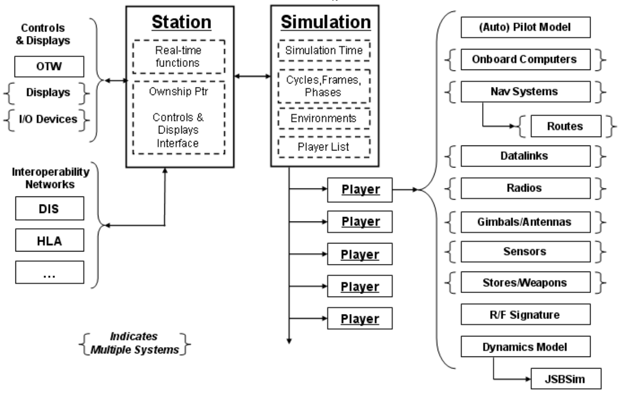

Figure 2. The architecture of openeaagles [14].

2) FlightGear [16]

FlightGear, published in 1997, is an open-source flight simulator. It provides lots of simulation models such as airplanes, airfields with massive maps and world data and so on.

FlightGear provides a high-reality simulation for flying an air vehicle, but it only focus on the flight posture and control and don't embed complex counteractions in the air combat for operational effectiveness simulation. At the same time, the simulation model framework is hidden in the codes, without a platform independent specification as OpenEaagles, and simulation models lack a standard model specification.

3) MultiUAV2 [17]

MultiUAV2 is a multi-UAV corporation and control platform based on MATLAB, SIMULINK and C++. The initial edition was published in 2002. In 2004, MultiUAV2 came into use. MutliUAV2 consists of simulation models such as sensors, targets, weapons, communication and their relationships to provide researchers a simulation environment for developing, executing and analyzing the control algorithms of UAV systems.

So MultiUAV2 is developed from the technology view in the purpose of developing, executing and analyzing algorithms. Though there is a model framework, the purpose of the model framework is specified, so it has confinements for effectiveness simulation of UAV systems, not mention to lacking a standard model specification and platform-independent description of model framework.

4) X-Plane [18]

X-Plane is a flight simulator developed by a game company Austin Meyer. It uses Blade Elements Simulation method to provide a near-real flight model and can be used as a project tool for predicting the flight quality of an air vehicle.

Thought X-Plane owns extrusive advantages in flight simulation, there are several problems for using X-Plane for operational effectiveness simulation of UAV system. Firstly, the purpose of X-Plane is to simulate the flight status of the air vehicles in different conditions, velocities, postures, so it doesn't contain models about the operational applications of UAV systems. Secondly, X-Plane is commercial software, and extensibility for X-Plane needs its developers to help. That's always impossible and so it's hard to solidify the domain invariant knowledge of UAV systems in X-Plane.

III. An Ontology Model of UAV Systems Effective Simulation Model Framework

Referring to simulation frameworks and environments related to UAV systems, domain invariant knowledge in the effectiveness simulation domain of UAV systems are captured and constructed as an ontology model for effectiveness simulation model framework. The classes and their hierarchy relationships in Protégé are shown in Fig .3. The classes of the ontology models are divided into two large categories, respectively systems and players. Players include space vehicles, air vehicles, ground vehicles and weapons while systems consist of navigation systems, pilots, external stores, R/F systems, data links, antennas, IR systems and so on.

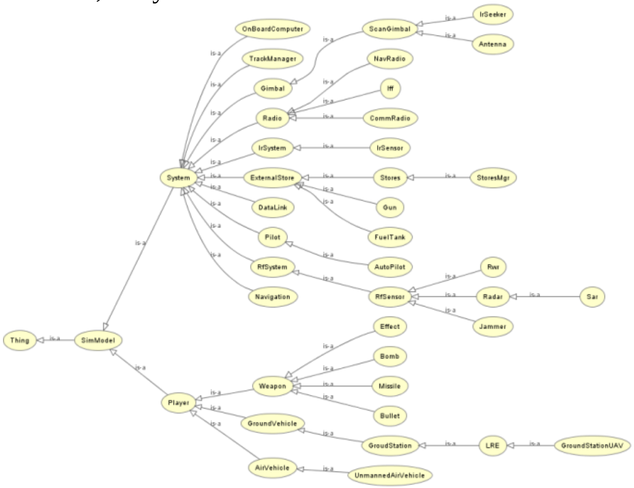

Figure 3. The classes and hierarchy realtionship of the ontology model of effectiveness simulation model framework for UAV systems

The relationships of the elements in UAV systems are shown in Fig.4. The relationships in the domain of operational effectiveness simulation of UAV systems are divided into two categories, assembly relationships and interaction relationships. Players can assemble navigation systems, on-board computers, IR systems, R/F systems and communication systems. And players can own gimbal components to receive and reflect IR or electromagnetic signals. On-board computers can assemble track manager systems. Store managers are responsible for external stores such as fuel and guns and weapons while guns comprise bullets. And R/F systems and IR systems can contain 0 or more antenna or IR signal receivers.

The interaction relationships in the UAV systems are divided into four categories:

1) *Crash&Kill*

When a player collides with something, crash events emerge between the player and the collider. When a player is killed by something, killing events happen between the player and the killer.

2) *Communication*

Player communicates with each other by radio systems. If radios of players are in the same wave band and channel, then a data link is created to send or receive messages by the DataLinkComm interaction. The messages consist of message types and contents. The receiver of the message reacts to the message according to the type and content of the message.

3) *Detection*

IR signal receivers of IR systems wait for commands such as SensorRTS, ScanStart and ScanEndto to return to search, start scanning or end scanning. During scanning, IR systems send IrQuery messages to detected players, and the players send back IrQueryReturn messages to the IR systems to report their IR signal.

R/F systems send electromagnetic signals RFEmission to detected players by antennas and the players send back electromagnetic signals RFReflectionEmission to report their reflected electromagnetic signals. For passive R/F systems, RFReflectionRequest signals are sent to detected players to subscribe electromagnetic signals of the R/F systems on the players. Then when the player sends out electromagnetic signals, the passive R/F systems will be noticed by RFEmission. And the subscribe relationships can be cancelled by sending RFReflectionCancel signals to the players.

4) *Weapon*

Players wait for WeaponRel commands to let their store manager systems launch weapons. And they are ready to respond to TGTStep events to replan and reallocate their attack targets. TriggerSW commands will be handled by players to fire guns. Store managers on the players waits for the WeaponReload commands to reload weapons. And for some weapons jetting to launch, external stores wait for Jettsion signals to decide if jetting these weapons.

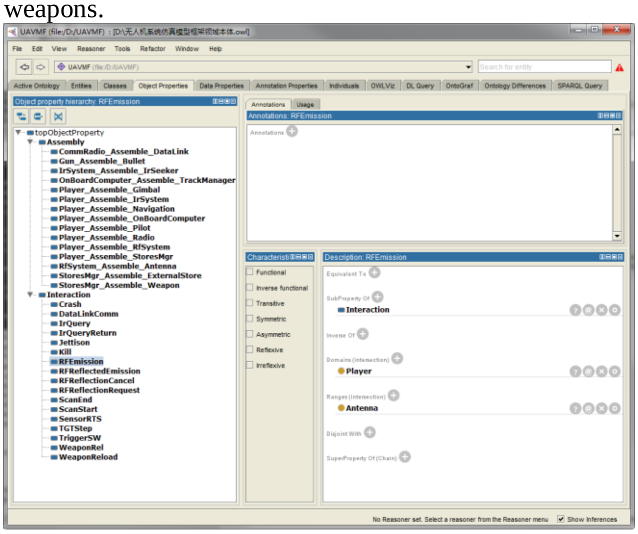

Figure 4. The realtionships in the ontology model of effectiveness simulation model framework for UAV systems

IV. Transformation From Ontology Model to SMP2 BasedSimulation Model Framework

The OWL-based ontology model of the effectiveness simulation model framework for UAV systems provides a computation independent model. However, to apply in the effectiveness simulation, it is necessary to translate the ontology model into the software implementation of a simulation framework. In this section, the transformation mechanism is designed from the Ontology model to

SMP2-based effectiveness simulation model framework and according to SMP2C++ mapping, C++ codes of the framework codes are generated for further simulation development.

A. The mapping mechanism

The mapping mechanism between the ontology model in Section 3 and SMP2 based simulation model framwork is shown in Table 1. An ontology model is mapped to a SMP2catalogue. There are four other types of mapping:

- Class Mapping. The class in the domain ontology is mapped to the model in the SMP2 model framework, and the hierarchy relationships are mapped to the super and sub relationships between the models in SMP2. The models which own sub models are mapped to a namespace at the same time.
- Field Mapping. Data properties in the ontology model are mapped into the fields of the models in SMP2 model framework.
- Assembly Mapping. The assembly relationships in the ontology model are mapped to the container relationship between the models transformed from the domain and range properties of object properties in the ontology model.
- Interaction Mapping. The interaction relationships in the ontology model are mapped to the event type in the SMP2 model framework, the domain and range properties of interaction object properties in the ontology models are correspondent to the EventSink and Event Source of an event type in SMP2 based model framework.

TABLE I. MAPPING MECHANISM

Ontology model	SMP2 model framework
Ontology	Catalogue
Super Class	Namespace
Class	Model
SimModel	Model(Root)
DataProperty	Field
Assembly Relationship	Container
Interaction Relationship	EventType
Domains(in Interaction)	EventSource
Ranges(in Interaction)	EventSink

B. Transformation algorithms from ontology model to SMP2 model framework

The OWL-based ontology model and SMP2-based model framework are both saved in a tree-structural XML. The transformation from the ontology model to the SMP2 model framework is to traverse each node in the ontology model and transform it to the correspond element in SMP2 model framework according to the type of the node.

In this section, referring to the method proposed in [19], the transformation rules from nodes in the ontology model to the corresponding nodes in the SMP2 simulation model framework are designed and described in a formal way. And the process for transforming from the ontology model to the SMP2 model framework is shown in pseudo codes. And finally OWL API plugin for Protégé is used to implementation the transformation.

1) Formal definition of transformation rules

The transformation rules are shown in Table 2.

TABLE II. TRANSFOMATON RULES

Rule 1：Transformation Ontology to SMP2 Catalogue

Ontology to SMP2 Catalogue
{
Attribute:
Id = ontology IRI;
Name = ontology IRI;
}

Rule 2：Transfromation Super Class to Namespace

Ontology Class<some class's SubClass of = name> to SMP2 Namespace
{
Attribute:
Id = case<name = SimModel> name;
case<name!=SimModel> SuperModel's Namespace + .name;
Name = name;
}

Rule 3：Transfromation Class to Model

Ontology Class to SMP2 Model
{
Attribute:
Id=case<name=SimModel>SimModel.tmSimModel
case<name!=SimModel>SuperModel's Namespace + .name.+tm+name;
Uuid = Generate a new uuid;
Name=tm+name;
}

Rule 4：Transformation DataProperty to Field

Ontology DataProperty to SMP2 Field
{
Attibute:
Id = Model's Id + .name;
Name = name;
}

Rule 5：Transformation Assembly to Container

Ontology Assembly to SMP2 Container
{
Attribute:
Id= Model's Id + name;
}

Rule 6：Transformation Interaction to EventType

Ontology Interaction to SMP2 Container
{
Attribute:
Id= Model's Id + name;
}

Rule 7：Transformation Domain to EventSource

Ontology Domains(in Interaction) to SMP2 EventSource
{
Attribute:
Id=Model'Id+.+ObjectProperty's name + Source;
Name = ObjectPropery's name + Source;
}

Rule 8：Transformation Ranges to EventSink

Ontology Ranges(in Interaction) to SMP2 EventSink
{
Attribute:
Id=Model's Id+.+ObjectProperty's name + Sink;
Name = ObjectPropery's Name + Sink;
}

2) *Pseudo codes for transformation*

According to the transformation rules in the previous section, we use pseudo codes to show the transformation process in Table 3. This process is a transverse process to execute the transformation rules by depth first search method.

TABLE III. PSEUDO CODES FOR MODEL TRANSFORMATION

```
1 For each ontology in Ontology:
2--create a catalogue;
3--perform Rule 1;
4--For each class in ontology:
5----if class is a subclass of Thing:
6--------if class is visited
7----------continue;
8------perform Rule 2;
9------perform Rule 3;
10------for each subclass in class:
11--------if subclass is visited
12----------continue;
13--------perform Rule 2;
14--------perform Rule 3;
15--------for each dataProperty of subclass
16------------perform Rule 4;
17--------End for
18--------for each objectProperty of subclass:
19----------if objectProperty is subproperty of Assembly:
20---------------Perform Rule 5;
21----------else
22---------------Perform Rule 6;
23---------------Perform Rule 7;
24---------------Perform Rule 8;
25-----------End if
26-----------class = subclass;
27-----------goto 10;
28----------End if
29--------End for
30------End for
31----End if
32--End for
```

3) *Implementaion of the transformation*

Protégé owns a plugin OWL API [20] which provides a set of API for traversing each node in the ontology model document to visit and modify the ontology model. SMP2 provides the XSD file for catalogue to guide how the catalogue file is read or saved. So traversing each node of the ontology model by OWL API, and for each node transforming to correspondent element in SMP2 catalogue file according to the rule, the ontology model can be transformed into SMP2-based simulation model framework. The transformed framework from the ontology model in Section 3 is shown in Fig.5.

The SMP2 model framework transformed from the ontology model of the effectiveness simulation framework is foundational, but contains the most important simulation models and the event and assembly relationships between the models. More information needs to be added into the framework to support simulation application development such as the type of the filed, how the models are executed in computer, and how the simulation process are scheduled, and so on.

V. CONCLUSIONS

Operational effectiveness simulation is an important warrant for design and validation of UAV systems. In order to support simulation assessments of UAV systems, it's necessary to rapidly develop simulation applications. Currently, among development methods for simulation application, it's a better way to build a model framework under a standard model specification while providing extension points or mechanisms for the model framework for develop simulation application quickly.

In order to support effectiveness simulation assessment for UAV systems, domain knowledge are absorbed from some famous UAV systems related simulation framework and environments such as OpenEaagles and FlightGear. Then the domain invariant knowledge of the effectiveness simulation domain of UAV systems are modeled into an ontology model based on OWL ontology modeling languages. Finally the ontology model is transformed to a SMP2-based simulation model framework to support simulation models and environments development.

The method for building a SMP2-based effectiveness simulation model framework for UAV systems can also be applied in other domains to support domain specific model framework development.

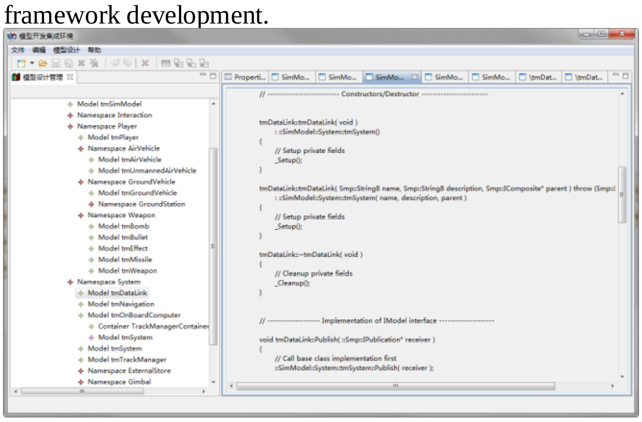

Figure 5. The tranfored SMP2 catalogue from the ontology model

ACKNOWLEDGMENT

This work is partly supported by the national Natural Science Foundation of China (No.61273198).

REFERENCES

[1] F. Yang, W. Wang, et al. Weapons and Equipment Operational Effectiveness Simulation and Evaluation. Beijing: Publishing House of Electronics Industry, 2010. (in Chinese)

[2] G. YoungSon, K. Baker, D. Kelleher, et al. Project Support Services for the Operational Mission and Scenario Analysis for Multiple UAVs/UCAVs Control from Airborne Platform Phase Ⅱ Summary Report . Canada:Defense National Canada, 2004.

[3] Y. Lei, Q. LI, F. Yang, et al. A composable modeling framework for weapon systems effectiveness simulation. Systems Engineering Theory & Practice, 2013, 33(11):2954-2966. (in Chinese)

[4] Z. Zhu, Y. Lei, N. Zhu, et al. Composable modeling frameworks for networked air & missile defense systems. NUDT Transactions, 2014, 36(5):186-190. (in Chinese).

[5] X. Li. A DSM-based Multi-Paradigm Composable Modeling Method for Effectiveness Simulation. Changsha: National University of Defense Technology, 2013. (in Chinese).

[6] G. Guizzardi. Ontological Foundations for Structural Conceptual Models. Thesis. Telematica Insituut Fundamental Research Series, 2005.

[7] D.L. McGuinness, F. Harmelen. OWL Web Ontology Language Overview. World Wide Web Consortium (W3C) recommendation, 2004.

[8] M. Horridge. A Practical Guide To Building OWL ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3, 2011.

[9] Q. Li, Y. Lei, H. HOU et al. Simulation Model Portability and Application , 2010. (in Chinese)

[10] Q. Feng, D. Zhou. The development trend of unmanned air vehicle. Electronics optics & control, 2003, 10(1):9-13. (in Chinese).

[11] IEEE Std 1516.1-2000. IEEE Standard for Modeling and Simulation High Level Architecture - Federate Interface Specification, 2000.

[12] F. P. Brooks. No Silver Bullet - Essence and Accident in Software Engineering. Proceedings of the IFIP Tenth World Computing Conference, 1986:1069-1076.

[13] Y. Lei, N. Su, J. Li, et al. New simulation model representation specification SMP2 and its key application techniques, 2010, 30(5):899-908. (in Chinese).

[14] ECSS. ECSS-E-TM-40-07 Volume 2A: Space engineering Simulation modeling platform-Volume 2: Metamodel, 2008.

[15] D. D. Hodson, D. P. Gehl, R. O. Baldwin. Building Distributed Simulations Utilizing the EAAGLES Framework. Proceedings of Interservice/Industry Training , Simulation , and Education Conference(I/ITSEC) , 2006.

[16] J. Rugarn. Rapid Development of Realistic UAV Simulations. Sweden: Linkopings Universitet, 2009.

[17] T. Shima, S. Rasmussen. UAV Cooperative Decision and Control Challenges and Practical Approaches. Translated by LIU Zhong, et al. Beijing: National Defense Industry Press, 2012.

[18] Austin Meyer Inc. X-Plane 10 Global. www.X-Plane.com. [2015-06-05]

[19] C. Li. UML-based SMP2 Model Design and Integration. Hunan: National University of Defense Technology, 2009.

[20] Protégé staff members. Protégé OWL API Programmer's Guide. Protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide. [2015-06-05]