

СПИСОК ЛІТЕРАТУРИ

1. Дьячук, Т. С. Дослідження ефективності роботи алгоритму планування ресурсів для розподіленої системи / Т. С. Дьячук, Р. К. Кудерметов, А. М. Літава, Р. І. Сметанін // Вісник КДПУ імені Михайла Остроградського. – Кременчук : КДУ, 2010. – Вип. 1/2010 (60) частина 1. – С. 47–50.
2. Эвристики распределения задач для брокера ресурсов Grid [Электронный ресурс] / А. И. Аветисян, С. С. Гайсарян, Д. А. Грушин, Н. Н. Кузюрин, А. В. Шокуров // Труды Института системного программирования РАН – 2004. – Режим доступа: <http://citforum.univ.kiev.ua/nets/digest/grid/index.shtml> – Загл. с экрана.
3. Клейнрок, Л. Теория массового обслуживания. Пер. с англ. / Клейнрок, Л. ; пер. И. И. Грушко; ред. В.И. Нейман. – М. : Машиностроение, 1979. – 432 с.

Стаття надійшла до редакції 17.03.2011.

УДК 004.9:004.82

Дьячук Т. С.

ОЦІНКА ХАРАКТЕРИСТИК РОЗПОДІЛЕНОЇ СИСТЕМИ

Виконано узагальнений аналіз розподіленої системи з точки зору систем масового обслуговування. Визначено характеристики, які можуть бути досліджені за допомогою систем масового обслуговування.

Ключові слова: система масового обслуговування, Grid, кластер, ресурс, завдання.

Diachuk T. S.

EVALUATION OF DISTRIBUTED SYSTEM'S CHARACTERISTICS

The distributed system in terms of queueing systems is analyzed. The characteristics that can be studied using queueing systems are presented.

Key words: queueing system, Grid, cluster, resource, task.

Гонтарь Н. А.¹, Кудерметов Р. К.²

¹Ассистент Запорожского национального технического университета

²Канд. техн. наук, профессор Запорожского национального технического университета

РАЗРАБОТКА ОНТОЛОГИИ СИСТЕМНОГО ИНЖИНИРИНГА КОСМИЧЕСКИХ СИСТЕМ

В статье обоснована необходимость создания онтологии системного инжиниринга (СИ). Онтология СИ вместе с переходом к модели-ориентированному СИ повысит эффективность разработки сложных систем. Предложен прототип онтологической модели, основанный на дескриптивной логике и формализованный с помощью языка OWL DL.

Ключевые слова: системный инжиниринг, онтология, MBSE, дескриптивная логика, OWL.

ВВЕДЕНИЕ

Согласно определению Международного Совета по Системной Инженерии (International Council on Systems Engineering, INCOSE) системный инжиниринг (СИ) – это междисциплинарный подход и средства для создания успешных систем [1, 2]. На протяжении всего жизненного цикла (ЖЦ) систем (концептуальное проектирование, разработка, изготовление, испытания, эксплуатация и утилизация) СИ фокусирует внимание разработчиков на глубоком анализе и отслеживании потребностей пользователей создаваемой системы и функциональных требованиях к ней. СИ интегрирует дисциплины и коллективы разработчиков в единое распределенное виртуальное пространство проекта, в котором согласованы процессы разработки и изготовления системы, взаимодействия разработчиков, применяемые ими понятия, правила, методы, инструменты и т. д.

СИ – итерационный процесс и каждое новое, принятое в ходе ЖЦ системы, решение отражается на конечном результате разработки. Поэтому важным является создание на основе современных информационных технологий благоприятной среды, отражающей текущие

процессы СИ, множество моделей системы, возможные варианты решений и ошибки.

В одной из последних инициатив INCOSE – «Systems Engineering Vision 2020» декларируются принципы модели-ориентированного подхода к СИ, который получил название MBSE (Model-based Systems Engineering) [3, 4]. MBSE – это формализованное применение моделирования для поддержки системных требований, разработки, анализа, верификации и валидации систем на фазе концептуального проектирования и далее при разработке и последующих фазах ЖЦ системы. Основными задачами MBSE являются:

– обеспечить согласованное взаимодействие между заказчиками и разработчиками систем, а также между коллективами разработчиков;

– уменьшить риски и неопределенности и контролировать их в течение всего ЖЦ системы;

– поддерживать процессы контроля границ системы, ее агрегации, декомпозиции, интеграции, сертификации и прогнозирования функциональных возможностей;

– обеспечить согласованность множества системных аспектов, видений и моделей системы;

- обеспечить обзоримость и возможность отслеживания состояния разработки больших и сложных систем;
- за счет ясного, определенного представления системы в виде моделей, устранять неоднозначности при участии разработчиков с разными культурами и языками и использовании ими различных средств проектирования;
- обеспечить необходимое понимание вклада всех составных частей системы в ее миссию.

В качестве языка моделирования для СИ и MBSE перспективным считается язык SysML, построенный на базе языка UML и принятый как индустриальный стандарт консорциумом Object Management Group (OMG) в 2007 г. [5]. Язык SysML включает не только все возможности языка UML моделировать программные системы, но и дополняет его диаграммами, необходимыми для моделирования любых технических систем, в частности диаграммами требований, блоков (рис. 1) и параметрической диаграммой. Однако следует отметить, что SysML включает много избыточных и практически неиспользуемых диаграмм, а также, из-за неоднозначности некоторых конструкций и наличия в спецификации неформальных описаний, язык не всегда позволяет точно выразить семантические особенности системы.

ПОСТАНОВКА ЗАДАЧИ

В настоящее время интенсивно создаются онтологии различных предметных областей (ПрО) и исследуются возможности их использования в MBSE. В научных и промышленных сообществах системных инженеров сформировалось видение развития MBSE и интеграции его с онтологическим подходом, которое предполагает последовательное развитие онтологий от прототипов до стандартов (рис. 2) [6].

Для успешного перехода от документо-ориентированного СИ к модели-ориентированному СИ (т. е. к MBSE) и эффективного его использования в различных ПрО необходимо создание множества моделей самого MBSE и охватываемых им процессов. В частности, для однозначного понимания командами разработчиков этапов, процессов и задач MBSE, правильной обработки его артефактов приложениями необходимо создание машинно-интерпретированных представлений всех составляющих СИ и MBSE. Решению этой задачи будет способствовать создание онтологий СИ и MBSE. В данной работе ставится цель обосновать возможность создания таких онтологий.

ОНТОЛОГИЯ СИСТЕМНОГО ИНЖИНИРИНГА

Онтология формальная спецификация разделяемой концептуальной модели. Здесь под «концептуальной» моделью подразумевается абстрактная модель, описывающая систему понятий ПрО. «Разделяемая» модель означает согласованное ее понимание определенным сообществом (группой людей). «Спецификацией» является описание системы понятий в явном виде. «Формальность» модели подразумевает машиночитаемое представление данных [7].

Создание онтологии СИ позволит получить в распоряжение системных инженеров такую модельную стратегию СИ, которая объединит совокупность понятий, правил, отношений, данных и информации СИ, независимых от отдельных стандартов и терминов и не скрытых в программном коде. Если в MBSE SysML обеспечивает синтаксическую совместимость моделей процессов и систем, то онтология – их семантическую интероперабельность.

Онтология СИ не будет нести в себе новых знаний, но она формализует наработанные знания и явные заключе-

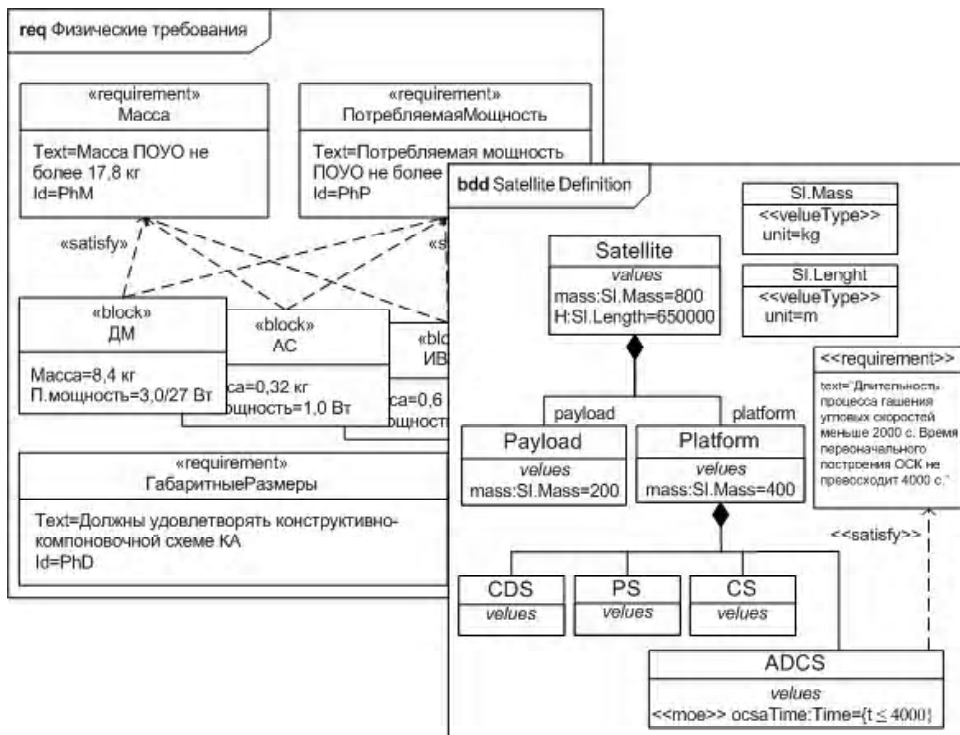


Рис. 1. Диаграммы требований и блоков для модели КС на языке SysML

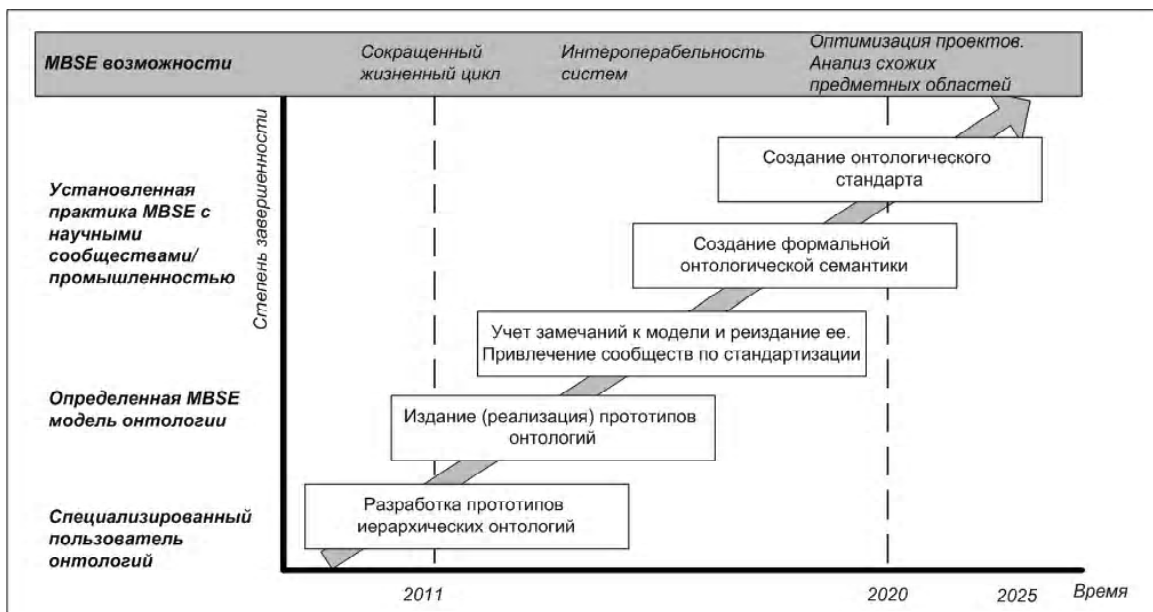


Рис. 2. Место онтологии в развитии СИ

ния и обеспечит их согласованность. Эти знания будут элементарными для восприятия системными инженерами и интероперабельными в распределенных компьютерных системах. Возможно, первые экземпляры онтологии СИ будут избыточными, громоздкими и противоречивыми в силу большого множества дисциплин самого СИ и дисциплин, которыми он управляет на протяжении ЖЦ системы.

В настоящее время имеются огромный международный опыт и знания в области создания космических систем (КС) и, поэтому становится вопрос об эффективном хранении и обработке этих знаний. Онтологическое представление СИ КС позволит специфицировать структуру и семантику терминов двух предметных областей, выразить различные формы сложных ограничений целостности и будет поддерживать работу с распределенными ресурсами, формализованную базу единой терминологии для разработчиков, автоматическую обработку запросов, интеграцию ее или в нее других специфицированных формализованных прикладных ПрО. На рис. 3 показаны возможные области использования онтологий и в частности онтологии СИ КС.

Определение характеристик КС обычно идет от функционального описания к физическому. Физическое описание формализуется в дереве продукта, которое представляет собой структуру, исходящую из упорядоченного и исчерпывающе разбитого на составные части конечного продукта, представленного в виде последовательности уровней, на которых расположены его составляющие [8, 9]. Эти составляющие конечного продукта меняют свою сущность по мере разбиения от верхнего уровня до нижних уровней декомпозиции. Как правило, они становятся менее сложными функционально, более компактными и самодостаточными в физическом смысле, включая при этом меньшее число используемых технологий. Поэтому онтология СИ КС должна отражать не только процессы СИ, но и техническую и структурную декомпозиции КС.

Разработку принципов СИ и MBSE для КС осуществляют организации NASA, ECSS, INCOSE, OMG и др. Стандарты, спецификации и рекомендации этих организаций должны стать источником информации для формирования понятий онтологии СИ КС. В настоящее время работы по инженерии знаний и онтологии в области КС проводятся NASA в рамках проекта NEXIOM (NASA Exploration Information Ontology Model) [10]. Проект NEXIOM разрабатывается с целью обеспечения:

- возможности отслеживать все изменения системы и ее элементов на протяжении всего ЖЦ (трассируемость);
- возможности проводить полнофункциональный анализ системы;
- поддержки интерпретации и доступа к информации, как со стороны разработчиков, так и программных агентов.

Данный проект должен регулировать техническую, экономическую и организационную части процесса разработки систем, вплоть до стандартизации документооборота и рабочего времени инженеров. NEXIOM нацелен на создание модели, которая будет содержать данные и знания о существующих стандартах хранения, организации и обработки данных, методах и средствах моделирования, опыте предыдущих успешных разрабо-



Рис. 3. Область применения онтологии СИ КС

ток систем. Предполагается, что стандарты и спецификации NEXIOM будут развиваться во времени с учетом потребностей разработчиков. Одной из целей рабочей группы проекта NEXIOM является поиск экономически эффективных способов обмена информацией между NASA и промышленностью и ожидается, что онтологии будут играть в этом ключевую роль.

Эффективность создания онтологий возрастает за счет того, что онтологии ПрО могут разрабатываться независимо, а затем проходить процессы согласования и адаптации между собой. Поэтому, учитывая, что проект NEXIOM направлен на организацию и формализацию знаний всего процесса создания КС, целесообразно создавать онтологии отдельных дисциплин этого процесса, в частности СИ КС, как дисциплины и процесса, интегрирующего множество других дисциплин и команд разработчиков на протяжении всего ЖЦ систем.

В качестве инструмента для создания онтологии СИ КС предполагается использовать программную среду проектирования онтологий *Protégé*, которая включает редактор онтологий, позволяющий проектировать онтологию, разворачивая таксономическую структуру абстрактных или конкретных классов и слотов. Структура онтологии создается аналогично иерархической структуре каталога. На основе сформированной онтологии *Protégé* может генерировать формы получения знаний для введения экземпляров классов и подклассов. *Protégé* позволяет вводить в онтологию классы методом заполнения форм, составленных из элементов мета-описания.

Одним из важных шагов при проектировании онтологии является распределение классов в таксономию. Смысловые подклассы одного и того же класса в онтологии должны находиться на одном уровне. Подклассов должно быть не меньше двух и не больше двенадцати, иначе, по мнению разработчиков *Protégé*, можно выделить еще один подкласс.

Основные концепты (понятия) СИ КС будут представлять собой объекты онтологии, реализованные как классы с характерными атрибутами. Такой способ представления информации позволяет работать как с СИ в целом, так и с ее структурными элементами в отдельности. Мы используем способ нисходящей разработки, который начинается с определения самых общих понятий ПрО СИ КС с последующей их структуризацией. Наиболее фундаментальные понятия СИ КС будут классами, которые находятся в верхней части таксономического дерева, корнем которого является класс «*THING*», охватывающий все множество концептов онтологии.

На рис. 4 показан простейший пример, демонстрирующий возможность создания онтологии СИ КС с помощью *Protégé*. В качестве источников данных для этого примера использовались основные понятия, отношения между ними и атрибуты, изложенные в стандартах ECSS, NASA, INCOSE и публикациях, посвященных разработке КС.

Не существует единственного «правильного» способа или методологии разработки онтологий, каждый аналитик составляет план и этапы онтологического анализа,

опираясь на свой опыт и существующие проекты онтологий. Тем не менее, можно выделить некоторые фундаментальные правила разработки онтологии [11]:

- необходимо рассматривать множество альтернативных способов моделирования и моделей ПрО. Лучшее решение почти всегда зависит от предполагаемого приложения и ожидаемых расширений;

- разработка онтологии – это обязательно итеративный процесс;

- понятия в онтологии должны быть близки к объектам (физическим или логическим) и отношениям в интересующей нас ПрО. Скорее всего, это существительные (объекты) или глаголы (отношения) в предложениях, которые описывают данную ПрО.

ФОРМАЛИЗАЦИЯ ОНТОЛОГИИ СИ С ПОМОЩЬЮ ДЕСКРИПТИВНОЙ ЛОГИКИ

Для выражения фундаментальных понятий при формализации данных о сложных системах и их концептуальном моделировании целесообразно использовать дескриптивные логики (ДЛ). ДЛ состоят из двух уровней: интенционального и экстенционального. Интенциональный уровень определяет множество понятий и отношений, описывающих концептуальную структуру ПрО (классы, свойства, отношения, аксиомы). Экстенциональный уровень содержит множество экземпляров элементов, описанных в интенциональном уровне [12]. С помощью ДЛ определяются основные элементы онтологии, которые позволяют отделить знания о ПрО от оперативных знаний и опыта.

Существует несколько диалектов дескриптивной логики, которые отличаются выразительными возможностями. Для описания онтологии СИ КС выбран диалект AL, который считается базовым для ДЛ и содержит необходимые и достаточные конструкции для описания созданного безэкземплярного прототипа онтологии, в котором имеются: T – концепт «*THING*»; \perp – концепт «*NOTHING*»; \cap – логическая связка (конъюнкция); \exists – квантор существования; \forall – квантор ограничение значения; \neg – логическая связка (отрицание).

Концепты онтологии определяются индуктивно следующим образом:

- все имена концептов, T и \perp являются AL-концептами;

- если A является AL-концептом, то и $\neg A$ является AL-концептом;

- если $A0$ и $A1$ являются AL-концептами, а r – имя роли, то $A0 \cap A1$, $\exists r.A0$, $\exists r.A1$, $\forall r.A0$, $\forall r.A1$ являются AL-концептами.

Онтологическая модель описывается унарными и бинарными предикатами. Унарные предикаты представляют собой фундаментальные понятия и соответствуют классам, которые находятся в корне таксономического дерева. Например, для онтологии СИ КС определены унарные предикаты: «*SYSTEMS – ENGINEERING*» концепт абстрактных классов онтологии, «*REQUIREMENT* –

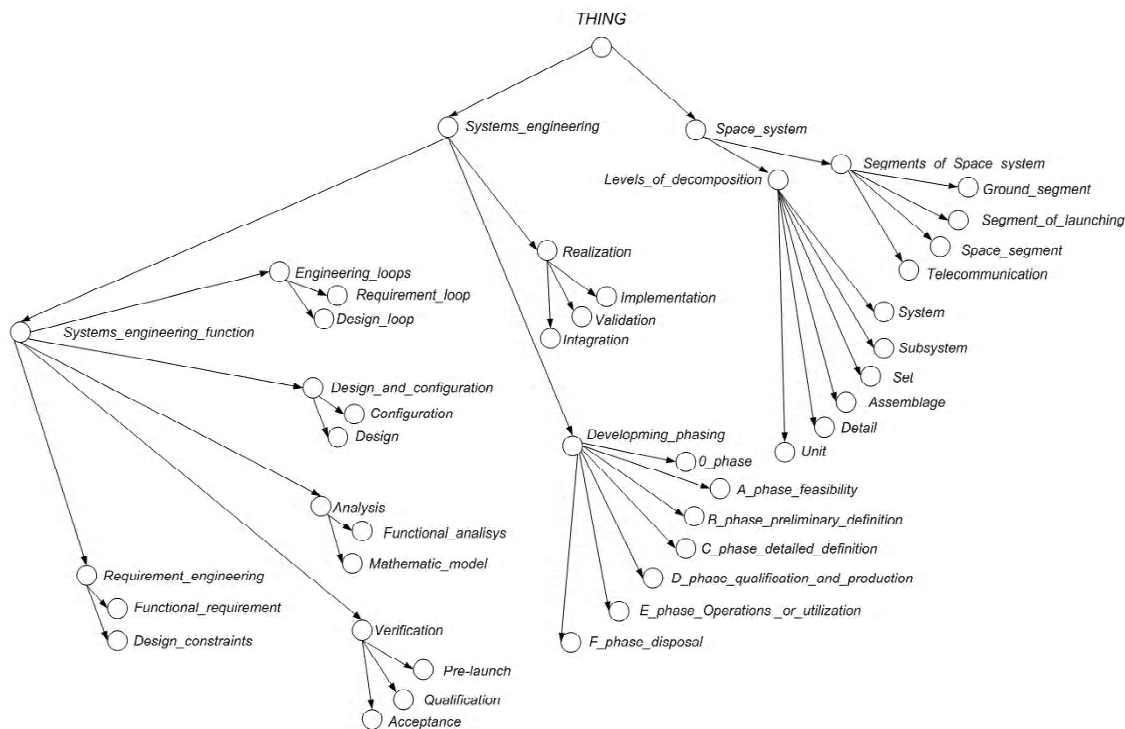


Рис. 4. Таксономия классов онтологии СИКС

ENGINEERING» выделенное понятие Про СИ, определенное как класс онтологии.

Бинарные предикаты – это роли. «hasInstance» бинарный предикат, определяющий понятие того, что с концептом, стоящим справа от роли, имеет место существование такого отношения «hasInstance.A».

С помощью предикатов в онтологиях строятся отношения и аксиомы. Например, следующая формула показывает отношение «существование концепта, определенного как экземпляр класса FUNCTIONAL – REQUIREMENT»:

$$\exists \text{hasInstance.FUNCTIONAL – REQUIREMENT} \quad (1)$$

В онтологиях всегда predeterminedены два класса «THING» и «NOTHING». «THING» это множество всех концептов онтологии. «NOTHING» это пустое множество концептов. Таким образом, если мы введем концепт «SYSTEMS – ENGINEERING», то он будет экземпляром класса:

$$\text{SYSTEMS – ENGINEERING} \exists \text{hasInstance.T} \quad (2)$$

Для части онтологии, представляющей онтологию дисциплины «инженерия требований», СИ КС в качестве концептов или унарных предикатов и прием и, которые будут выделенными понятиями СИ и определенными как классы онтологии.

В самом общем случае, терминологические аксиомы первого рода (включений) (3) и второго рода (равенств) (4) имеют вид:

$$A0 \sqsubseteq A1, r0 \sqsubseteq r1; \quad (3)$$

$$A0 \equiv A1, r0 \equiv r1. \quad (4)$$

Для онтологии СИ КС с помощью аксиом первого рода определим таксономию концептов онтологии. Например, класс «REQUIREMENT – ENGINEERING» является подклассом «SYSTEMS – ENGINEERING», а класс, в свою очередь, является надклассом класса:

$$\begin{aligned} \langle \text{REQUIREMENT – ENGINEERING} \rangle &\sqsubseteq \\ &\sqsubseteq \langle \text{SYSTEMS – ENGINEERING} \rangle, \end{aligned} \quad (5)$$

$$\begin{aligned} \langle \text{FUNCTIONAL – REQUIREMENT} \rangle &\sqsubseteq \\ &\sqsubseteq \langle \text{REQUIREMENT – ENGINEERING} \rangle. \end{aligned} \quad (6)$$

Таким образом, ДЛ позволяет описать модель онтологии системного инжиниринга с точки зрения математики, создать устойчивые непротиворечивые формулировки термов онтологии. ДЛ обладает мощным аппаратом формализации понятий и при дальнейшей работе с онтологией позволяет рассматривать все элементы как с точки зрения их семантики, так и со стороны логических законов.

Несмотря на существующее множество технологий и моделей хранения, интерпретации и обработки данных, онтология усиливает возможности концептуального моделирования. Работа с онтологической моделью реализует абстрактное, единое, семантическое моделирование Про (интенциональный уровень), независящее от спецификации этой модели (экстенциональный уровень) и служащее в качестве средства создания эталонного инструмента взаимодействия разработчиков между собой и между программными приложениями.

Следует учитывать, что создание онтологии требует экспертных знаний в исследуемой Про и значительных

затрат времени. Онтология содержит формализованные данные, но не отображает методы, функции и процессы ПрО, поэтому онтология СИ будет являться лишь универсальным источником данных о ПрО СИ.

OWL-ОНТОЛОГИЯ СИ

Для формального машинного представления онтологий Консорциум W3C предложил язык описания онтологий OWL (Ontology Web Language), который имеет различные по выразительности диалекты, спроектированные для использования отдельными сообществами разработчиков и пользователей [13].

Диалект OWL DL наиболее популярен и используется при необходимости представления знаний о ПрО в максимально выразительном формализованном виде без потери полноты вычислений (все заключения гарантировано будут вычисляемыми) и разрешимости (все вычисления гарантированно завершатся за определенное время). Формальной основой онтологий на языке OWL DL является ДЛ, т.е. логика предикатов, предназначенная для представления терминологического знания о ПрО. OWL DL включает все языковые конструкции OWL с ограничениями вроде разделения типа (класс не может быть частным свойством, а свойство не может быть индивидом или классом).

Каждый индивид в OWL представлении является членом класса `owl:Thing`. Таким образом, каждый определенный класс автоматически является подклассом `owl:Thing`. Специфичные для данной ПрО корневые классы (понятия) определяются простым объявлением именованного класса. OWL также определяет пустой класс `owl:Thing`.

Упомянутая выше программная среда проектирования онтологий *Protégé* позволяет разрабатывать OWL-онтологии, а также автоматически их генерировать из ранее созданных таксономий классов. Ниже приведены фрагменты сгенерированных в *Protégé* OWL-описаний концептов, определенных классов и аксиом онтологии СИ КС. Так, введенные

нами выше классы «*SYSTEMS – ENGINEERING*» и «*REQUIREMENT – ENGINEERING*», определяются следующим образом:

```
<owl:Class rdf:ID="Systems_engineering" />
<owl:Class rdf:ID="Requirement_engineering" />
```

Определение общего класса онтологии «*SYSTEMS – ENGINEERING*» и принадлежность его классу «*THING*» (1) выражается так:

```
<rdfs:domain>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<rdf:Description rdf:about="
"http://www.w3.org/2002/07/owl#Thing" />
<owl:Class rdf:about="#Systems_engineering" />
</owl:unionOf>
</owl:Class>
</rdfs:domain>
```

Аксиомы включения (5) и (6) онтологии СИ КС, записываются в виде:

```
<owl:Class rdf:about="#Functional_requirement">
<rdfs:subClassOf
rdf:resource="#Requirement_engineering" />
</owl:Class>
<owl:Class rdf:about="#Requirement_engineering">
<rdfs:subClassOf
rdf:resource="#Systems_engineering" />
</owl:Class>
```

OWL-онтологию (документ, описанный на языке OWL) СИ КС при необходимости можно преобразовать в XML-документ и использовать его для обмена в распределенной компьютерной системе поддержки СИ. С другой стороны, OWL-документ можно конвертировать в XMI-формат для обмена моделями и, затем, в модель на языке SysML (UML). Следовательно, в зависимости от целей конкретных процессов СИ и на различных этапах ЖЦ системы можно использовать языки OWL и SysML (рис. 5)

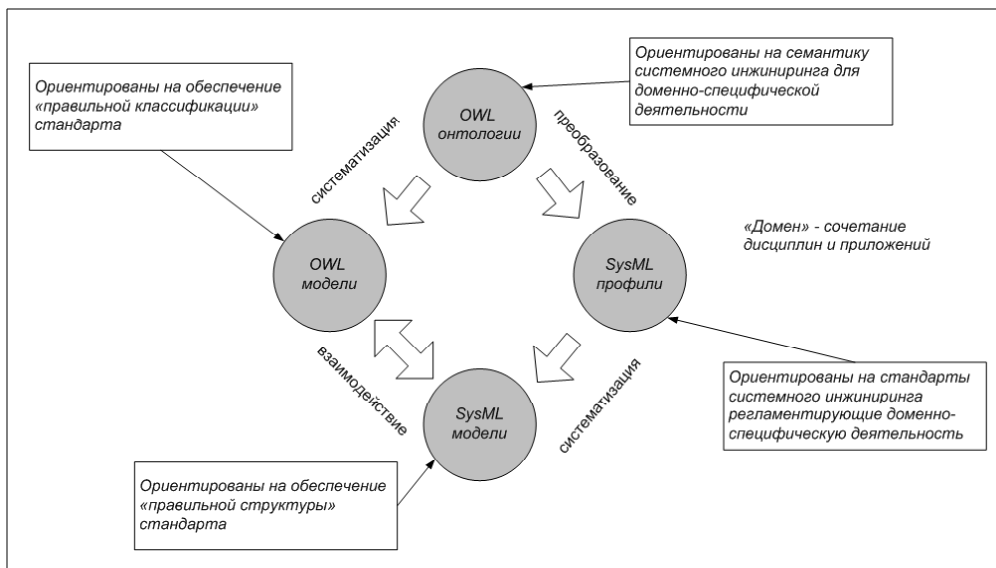


Рис. 5. Взаимодействие между OWL и SysML моделями

ВЫВОДЫ

Таким образом, в работе обоснована необходимость создания онтологии СИ КС, возможность ее формализации с помощью диалекта AL дескриптивной логики и представления ее на языке OWL DL. На примере разработанного прототипа продемонстрированы основные принципы, которые будут положены в основу дальнейшей разработки онтологии СИ КС. В качестве инструментального средства проектирования онтологии выбран пакет *Protégé*. Онтология СИ КС будет интегрирована в процесс MBSE, что позволит повысить эффективность использования MBSE для разработки сложных систем.

СПИСОК ЛИТЕРАТУРЫ

1. Systems Engineering Handbook [Text] / INCOSE. – TP-2003-016-02, Version 2a, 1 June 2004. – USA: INCOSE, 2004. – 300 p.
2. ГОСТ Р ИСО/МЭК 15288 – 2005. Системная инженерия. Процессы жизненного цикла систем [Текст]. – Введен 2005-12-29. – М. : Изд-во стандартов, 2006. – 57 с.
3. Systems Engineering Vision 2020 [Электронный ресурс] / INCOSE-TP-2004-004-02, September, 2007. – Режим доступа: http://www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf
4. Kossiakoff A. Systems engineering: principles and practice [Text] / Alexander Kossiakoff et al.–2nd ed. (Wiley series in systems engineering and management; 67)– WILEY, 2011. – 559 p.
5. OMG Systems Modeling Language (OMG SysML™) [Электронный ресурс]. – Режим доступа: www.sysml.org/docs/specs/OMGSysML-v1.2-10-06-02.pdf
6. Ontology in MBSE / INCOSE // International Council on Systems Engineering IW2011 MBSE Workshop, January 29, 2011. – USA, 2011. – 11 p.
7. Gruber, Thomas R. A translation approach to portable ontology specifications [Tex] / Thomas R. Gruber // Appeared in Knowledge Acquisition. – 1993. – №5(2). – P. 199–220.
8. ECSS-E-ST-10C. Space engineering. System engineering general requirements [Text]. – 6 March 2009. – ECSS Secretariat SA-ESTEC, Requirements & Standards Division. Noordwijk, Netherlands, 2009. – 100 p.

9. Systems Engineering Handbook [Text] / Aeronautics and Space Administration, Headquarters Washington DC 20546, SP-2007-6105 Rev1. – Washington, 2007. – 360 p.
10. NASA Exploration Information Ontology Model (NExIOM) primer and vision [Text] / National Aeronautics and Space Administration, Headquarters Washington DC 20546. – Washington, 2005. – 18 p.
11. Noy N. Ontology Development 101: A Guide to Creating Your First Ontology [Электронный ресурс] / Natalya F. Noy, D. L. McGuinness // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001. – Режим доступа: http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html.
12. Baader F. The Description Logic Handbook [Text] / F. Baader et al. – Cambridge University Press, 2003. – 574 p.
13. OWL Web Ontology Language Guide / W3C [Электронный ресурс]. – Режим доступа: <http://www.w3.org/TR/owl-guide/>

Стаття надійшла до редакції 23.05.2011.

Гонтар Н. А., Кудерметов Р. К.

РОЗРОБКА ОНТОЛОГІЇ СИСТЕМНОГО ІНЖИНІРИНГУ КОСМІЧНИХ СИСТЕМ

У статті обґрунтована можливість створення онтологій системного інжинірингу (СІ). Онтологія СІ разом з переходом до моделі-орієнтованого СІ підвищить ефективність розробки складних систем. Запропоновано прототип онтологічної моделі, створений за допомогою дескриптивної логіки і формалізований за допомогою мови OWL DL.

Ключові слова: системний інжиніринг, онтологія, MBSE, дескриптивна логіка, OWL.

Gontar N. A., Kudermetov R. K.

WORKING OUT ONTOLOGY OF SYSTEMS ENGINEERING OF SPACE SYSTEM

This article grounded the need to create ontology of systems engineering (SE). SE ontology with the transition to model-based approach SE will increase the efficiency of development of complex systems. The prototype of the ontology model based on description logic and formalized through language OWL DL is proposed.

Key words: systems engineering, ontology, MBSE, description logic, OWL.

УДК 004.627: 004.272.26

Скрупский С. Ю.¹, Луценко Н. В.², Скрупская Л. С.³

^{1, 3} Ассистент Запорожского национального технического университета

² Старший преподаватель Запорожского национального технического университета

ПАРАМЕТРЫ КОМПРЕССИИ ВИДЕОИНФОРМАЦИИ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Проанализированы основные параметры компрессии видеоинформации в распределенных системах. Экспериментально исследовано влияние этих параметров на коэффициент сжатия, уровень искажения и время сжатия видеоинформации в распределенной системе.

Ключевые слова: видеоинформация, распределенная система, компрессия, битрейт, уровень искажения.

ВВЕДЕНИЕ

В настоящее время видеоинформацию хранят в архивах в сжатом цифровом виде, поскольку в несжатом

виде она занимает значительные объемы памяти [1]. Проблема хранения видеоинформации особенно актуальна с распространением стандарта телевидения высо-