Серия «Математика»
2009. Т. 2, № 2, С. 171—182

Онлайн-доступ к журналу:
http://isu.ru/izvestia

ИЗВЕСТИЯ
Иркутского
государственного
университета

УДК 510.62:004.82

Логические формализации спецификаций на основе ОО-проекций *

Н. О. Стукушин
Иркутский государственный университет

Аннотация. В статье рассматривается проблема построения логических формализаций стандартов и спецификаций. Формализации полезны как при разработке, так и при использовании спецификаций. В частности, если формализация осуществлена в конструктивном ключе, то она может служить в качестве основы для реализации релевантных информационных систем. В данной работе развивается подход к формализации спецификаций, использующий специальные подлогики дескриптивных логик. Подход рассматривается на примере спецификаций международного образовательного консорциума IMS.

Ключевые слова: онтология, дескриптивная логика, формализация, спецификация, ОО–проекция, семантический веб.

1. Введение

В этой работе рассматривается идея использования дескриптивных логик [2] и онтологий для формализации спецификаций. Дескриптивные логики являются универсальным инструментом для формального описания знаний. Они предоставляют богатые выразительные средства, которые позволяют включать формализованные спецификации в компьютерные системы знаний. Диапазон различных дескриптивных логик весьма широк, что позволяет делать среди них выбор наиболее подходящего для наших целей формализма. Благодаря стандартизированному представлению онтологий в виде языка OWL (Web Ontology Language, [5]), имеющего строгую логическую семантику, формальное описание спецификации в виде онтологии обеспечивает однозначное

* Работа выполнена при частичной финансовой поддержке программ «Фундаментальные исследования и высшее образование» (проект НОЦ-017 «Байкал») и «Развитие научного потенциала высшей школы (2009-2010 гг.)» (проекты РНП.2.2.1.1/5901 и 3.2.3/3488).

понимание концептов и конструкций спецификации. Обеспечивается и переносимость формальных описаний, возможность их повторного использования, а также автоматической обработки информации, связанной со спецификацией. Формализация также предоставляет большие возможности для анализа правильности спецификации, ее соответствия поставленным задачам.

В процессе формализации нужно обеспечить «прозрачное» и точное отображение спецификации в те структуры (классы и объекты), которые будут в дальнейшем использоваться для работы. Кроме этого, необходимо не забывать и о моделях взаимодействия этих структур. Для решения этой задачи нами используется концепция ОО–проекций [4]. ОО-проекции – подлогики дескриптивной логики $\mathcal{SHOIN}(D)$ [6], которые позволяют формировать в рамках $\mathcal{SHOIN}(D)$ «объектные» модели предметных областей.

Технически концепция ОО-проекций реализована на платформе системы OntoBox [3]. OntoBox предоставляет спектр инструментов для обработки онтологий, описывающих предметную область. Данная платформа моделирует объектно-ориентированный подход к представлению информации и, кроме того, реализует свой собственный язык запросов к онтологиям - OntoBox QL [3].

В процессе формализации спецификации IMS QTI [7] потребовалось провести объемную работу по описанию всех классов стандарта в виде компонент ОО-проекции. В данной работе эта техника иллюстрируется на примере ключевого класса спецификации – AssessmentItem («Тест»). На его примере не только демонстрируется описание классов спецификации с помощью дескриптивной логики, но и иллюстрируется проверка корректности формального описания относительно определения ОО-проекции.

2. ОО-проекции

Логикой для построения логических архитектур, включающих ОО-проекции, была выбрана дескриптивная логика $\mathcal{SHOIN}(D)$[6]. Этот выбор обусловлен рядом причин, и в первую очередь тем, что $\mathcal{SHOIN}(D)$ лежит в основе языка описания онтологий OWL-DL [5]. Это обеспечивает совместимость подхода, разрабатываемого нами, с базовыми конструкциями семантического веба.

Пусть

$$\mathcal{D} = \langle D_1, \ldots, D_k; \Omega \rangle$$

некоторая область данных. Примерами D_i могут служить целые и вещественные числа, строки, даты и т.д. Ω содержит символы «встроенных» операций и отношений, работающих на данных D_i. Мы считаем, что все

элементы типов данных – выделенные, поэтому Ω содержит константы для каждого элемента каждого D_i. В дальнейшем через v, v_i будем обозначать константы из Ω, а через w, w_i основные термы (без переменных) на Ω.

Словарь дескриптивной логики включает символы концептов, ролей, атрибутов и объектов. Концепты выделяют в множестве объектов предметной области совокупности объектов, обладающих определенными свойствами. В дальнейшем концепты обозначаются через A, B, C, E, причем A, B, C обозначают именованные (атомарные) концепты, а E – произвольные концепты (формулы). Роли (о-свойства) определяют отношения между объектами предметной области и обозначаются через R, R_i. Атрибуты (т-свойства), обозначаемые через P, P_i, привязывают элементы области данных к объектам. Через O, O_i обозначаем имена объектов предметной области. Словарь – это четверка $\langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$, где $\mathcal{C}$ – множество именованных концептов, $\mathcal{R}$ – множество ролей, $\mathcal{P}$ – множество атрибутов, $\mathcal{O}$ – множество имен объектов. Область данных $\mathcal{D}$ считается фиксированной, поэтому ее компоненты в словари не входят.

Определение 1. [Язык $\mathcal{SHOIN}(D)$] Если $c \in \mathcal{C}$, то c называется примитивным концептом $\mathcal{SHOIN}(D)$. Примитивные концепты являются концептами. Если a, b – концепты, $id \in \mathcal{O}$, $r \in \mathcal{R} \cup \mathcal{P}$, $p \in \Omega$, то $a \sqcap b$, $a \sqcup b$, $\neg a$, $\exists r.a$, $\forall r.a$, $\leq_n r$, $\geq_n r$, $\exists(x_1, \ldots, x_n).p$, $\forall(x_1, \ldots, x_n).p$, $\{id\}$ также концепты. Если $r \in \mathcal{R}$, то $\exists r'.a$ и $\forall r'.a$ – концепты, где $r' \in \{r^+, r^-, r^\pm\}$.

Описание предметной области в $\mathcal{SHOIN}(D)$, как и в любой другой дескриптивной логике, состоит из двух блоков - А-бокса и Т-бокса. TBox содержит описание знаний концептуального (терминологического) уровня и состоит из аксиом включения:

$E_1 \sqsubseteq E_2$	интерпретируется как $\forall x.(E_1(x) \rightarrow E_2(x))$
$R_1 \sqsubseteq R_2$	интерпретируется как $\forall x \forall y.(R_1(x, y) \rightarrow R_2(x, y))$
$P_1 \sqsubseteq P_2$	интерпретируется как $\forall x \forall v.(P_1(x, v) \rightarrow P_2(x, v))$

$E_1 \equiv E_2$ – сокращенная запись двух аксиом $(E_1 \sqsubseteq E_2)$ и $(E_2 \sqsubseteq E_1)$. ABox описывает предметную область на уровне конкретных данных. ABox может содержать аксиомы следующих типов: $C(O)$ – принадлежность объекта концепту, $R(O_1, O_2)$ – отношение между объектами, $P(O, v)$ – значение атрибута объекта.

В работе [4] в рамках $\mathcal{SHOIN}(D)$ строится «объектно-ориентированная» подсистема, ответственная за построение объектных подмоделей произвольных логических моделей предметных областей. Такая подсистема назовается *ОО-проекцией*. ОО-проекции можно также рассматривать как модификацию и расширение простой дескриптивной логики $\mathcal{FL}_0$ [2].

Язык $\mathcal{FL}_0$ включает атомарные концепты $C_i \in \mathcal{C}$, роли $R_i \in \mathcal{R}$, конъюнкцию $\sqcap$, и квантор всеобщности $\forall R.C$. Также у нас имеются

две логические константы $\top$ (top) и $\bot$ (bottom). Так как $\mathcal{F}L_0$ является дескриптивной логикой, то любая интерпретация I этих конструкций на множестве объектов Δ должна удовлетворять правилам:

$$\top^I = \Delta$$

$$\bot^I = \emptyset$$

$$C_i^I \subseteq \Delta$$

$$R_i^I \subseteq \Delta \times \Delta$$

$$(F \sqcap G)^I = F^I \cap G^I$$

$$\forall R.C^I = \{o \mid o \in \Delta \text{ and } \forall o'.(R(o, o') \rightarrow C(o'))\}$$

Мы расширяем базовый формализм $\mathcal{F}L_0$ средствами для привязки объектов к данным из области $\mathcal{D}$. Это означает, что вместе с ролями R_i, которые описывают связи между объектами (как объектные роли `spouse` – отношение «быть супругом», и `hasChild` – отношение «иметь ребенка»), мы можем определять атрибуты объектов (например, целое число возраста `age`, и строку имени `name`). Для этого используются атрибуты $P_1, \ldots, P_k \in \mathcal{P}$. Наконец, мы предоставляем возможность явного именования объектов с помощью идентификаторов из $\mathcal{O}$.

Введем теперь оператор типизации θ, который типизирует роли и атрибуты:

1. привязывает роли $R \in \mathcal{R}$ к парам $\theta(R) = [C_d, C_r]$, где $C_d, C_r \in \mathcal{C}$;
2. привязывает атрибуты P к парам $\theta(P) = [C_d, D_r]$, где $C_d \in \mathcal{C}$ и $D_r \in \mathcal{D}$.

C_d называется областью определения роли R (атрибута P), C_r (D_r) называется областью значений роли R (атрибута P).

Будем говорить, что интерпретация I согласована с θ, если

$$\forall R \in \mathcal{R}.(\theta(R) = [C_d, C_r] \ \rightarrow \ R^I \subseteq C_d^I \times C_r^I)$$
$$\forall P \in \mathcal{P}.(\theta(P) = [C_d, D_r] \ \rightarrow \ P^I \subseteq C_d^I \times D_r^I)$$

Интерпретация I согласована со словарем $\langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$, если для любого объекта $o \in \top^I$ существует имя $O \in \mathcal{O}$ такое, что $o = O^I$ (т.е. все объекты выделенные). В дальнейшем будем рассматривать только интерпретации, согласованные с θ и словарем. $\mathcal{K} \models E$ будет обозначать, что E истинна в любой согласованной интерпретации $\mathcal{K}$. В отличие от этого $\mathcal{K} \vdash E$ означает, что E выводима из $\mathcal{K}$ в некотором полном исчислении (например, с помощью табличных алгоритмов), а значит, истинна в любой интерпретации.

Пусть $\mathcal{V} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$ – словарь.

Определение 2. [ОО-концепт] Концепт вида

$$C_1 \sqcap \ldots \sqcap C_f \sqcap \forall R_1.E_1 \sqcap \ldots \sqcap \forall R_g.E_g \sqcap \forall P_1.D_1 \sqcap \ldots \sqcap \forall P_h.D_h$$

называется ОО-концептом, где E_i – ОО-концепты, $C, C_i \in \mathcal{C}$, $R_i \in \mathcal{R}$, $P_i \in \mathcal{P}$ и $D_i \in \mathcal{D}$. Если все $E_i \in \mathcal{C}$, то ОО-концепт называется примитивным.

Определение 3. [ОО-определение] Аксиома

$$C \equiv E \tag{2.1}$$

где E – примитивный ОО-концепт, называется объектно-ориентированным определением (ОО-определением).

Пример ОО-определения:

$$\texttt{Person} \equiv \texttt{Mammal} \sqcap \forall \texttt{hasChild.Person} \sqcap \forall \texttt{name.String}$$

Неформальная интерпретация этих конструкций с помощью объектно-ориентированных концептов (что оправдывает использование префикса «ОО–») прямая. В ОО-определении

$$C \equiv C_1 \sqcap \ldots \sqcap C_n \sqcap \forall R_1.B_1 \sqcap \ldots \sqcap \forall R_k.B_k \sqcap \forall P_1.D_1 \sqcap \ldots \sqcap \forall P_l.D_l$$

атомарные концепты C, C_i, B_i обозначают ОО-классы, $C_1 \sqcap \ldots \sqcap C_n$ обозначает наследование, и если $n > 1$, тогда мы имеем множественное наследование. Каждое выражение $\forall R_i.B_i$ обозначает поле R_i, значения которого являются объектами класса B_i. Каждое выражение $\forall P_i.D_i$ соответствует полю, значения которого являются элементами предопределенного типа данных D_i.

Мы говорим, что концепт C в ОО-определении *непосредственно наследует* от $C_1, \ldots, C_n$. Пусть $\mathcal{A} = \{C_i \equiv E_i\}$ – множество ОО-определений. C_i *наследует* от C_j, если он непосредственно наследует от C_j, или существует C_k такой, что C_i наследует от C_k и C_k наследует от C_j. $\mathcal{A}$ называется ацикличным, если оно не содержит C_i, наследующих от самих себя.

Схожим образом, если концепт C определен с помощью ОО-определения в множестве ОО-определений $\mathcal{A}$, мы говорим, что концепт C *непосредственно зависит от* ролей $R_1, \ldots, R_k$ и атрибутов $P_1, \ldots, P_l$. C *зависит* от роли R (атрибута P), если он непосредственно зависит от роли (атрибута), или существует C' в $\mathcal{A}$, который непосредственно зависит от R (P), и C наследует от C'.

Теперь обратимся к определению ОО-проекции. Пусть $\mathcal{L}$ – дескриптивная логика, содержащая $\mathcal{SHOIN}(D)$ как подлогику, и $\mathcal{K} = TBox_{\mathcal{K}} \cup ABox_{\mathcal{K}}$ – $\mathcal{L}$–описание предметной области в словаре $\mathcal{V} = \langle \mathcal{C}, \mathcal{R}, \mathcal{P}, \mathcal{O} \rangle$ (в дальнейшем такие описания мы называем онтологиями). Пусть $\mathcal{V}_o = \langle \mathcal{C}_o, \mathcal{R}_o, \mathcal{P}_o, \mathcal{O}_o \rangle$ – подсловарь $\mathcal{V}$, $\mathcal{O}_o = \{O_1, \ldots, O_l\}$, а θ – оператор типизации на $\mathcal{V}_o$.

Определение 4. [ОО-проекция]

$$\mathcal{K}_o = TBox_o \cup ABox_o$$

– ОО-проекция словаря $\mathcal{V}_o$ и оператора θ, если

1. $TBox_o$ – ацикличное множество ОО-определений, любая интерпретация которого согласована с θ;
2. для каждой аксиомы $Ax \in TBox_o$ выполняется $\mathcal{K} \vdash Ax$;
3. любая роль и атрибут из словаря $\mathcal{V}_o$ входит в ОО-определения $TBox_o$ не более одного раза;
4. $ABox_o \subseteq ABox_{\mathcal{K}}$ и
   - для любого $C(O) \in ABox_o$: $O \in \{O_1, \ldots, O_l\}$ и $C \in \mathcal{C}_o$, причем для каждого O такой C является единственным;
   - для любого $R(O, O') \in ABox_o$: $O, O' \in \{O_1, \ldots, O_l\}$, $R \in \mathcal{R}_o$ такие, что $C(O), C'(O') \in ABox_o$, причем $\mathcal{K}_o \models C \sqsubseteq C_1$, $\mathcal{K}_o \models C' \sqsubseteq C_1'$ и $\theta(R) = [C_1, C_1']$;
   - для любого $P(O, v) \in ABox_o$: $O \in \{O_1, \ldots, O_l\}$, $P \in \mathcal{P}_o$, $C(O) \in ABox_o$ и $v \in |D|$, где $\theta(P) = [C_1, D]$ и $\mathcal{K}_o \models C \sqsubseteq C_1$.

Далее демонстрируется метод формального описания спецификаций с использованием формализма ОО-проекций.

3. Формализация IMS

В этой работе рассматривается формализация спецификаций международного образовательного консорциума IMS [7]. Наша цель - формализация спецификации IMS QTI, задающей стандарты построения образовательных систем тестирования. Дело усложняется тем, что IMS QTI зависит от спецификации описания метаданных образовательных ресурсов IMS MD и спецификации пакетирования образовательных ресурсов IMS CP. Поэтому эти спецификации также должны быть формализованы. Каждая из спецификаций содержит описание своих основных конструкций в виде XML (XML binding). XML binding формально описывает объектную модель, которая сосредоточена на синтаксических и структурных аспектах, но не затрагивает вопросы семантики. Семантика этих конструкций описана отдельно на естественном языке (английском).

Задача состоит в том, чтобы представить каждую из трех спецификаций IMS в форме онтологий. Каждая спецификация представляется в виде отдельной онтологии. Таким образом, для формализации IMS QTI необходимо построить три онтологии, которые мы будем называть *терминологическими*, поскольку они содержат базовое терминологическое описание структур. При этом каждый конкретный тест формализуется в отдельной онтологии, и включает в себя объекты с конкретными

данными о вопросах теста и ответах. Общее знание о «поведении» теста наследуется этими онтологиями от терминологических онтологий спецификации. Такая структура формализации позволяет превратить конкретный тест в удобный объект для распространения и повторного использования, поскольку её семантика и структура зависят только от терминологических онтологий IMS и не зависят от конкретных форматов систем тестирования.

Онтологии, описывающие конкретные тесты, будут корректно использоваться всеми сообществами, которые принимают терминологические онтологии IMS. Заметим также, что если у нас есть инструмент для эффективной работы с информацией, хранимой в онтологиях, то терминологические онтологии IMS могут использоваться непосредственно для реализации систем тестирования, в которых будут играть и другую полезную роль - роль структур представления данных внутри системы. Эту возможность мы продемонстрируем ниже, используя систему OntoBox.

Приступим непосредственно к формализации. Этот процесс предполагает описание всех элементов спецификации с помощью некоторого набора формул, которые описывают все класссы спецификаций с помощью выбранной ранее дескриптивной логики $\mathcal{SHOIN}(D)$. Причём в процессе описания нельзя забывать, что мы строим объектную модель спецификации, используя для этого ОО - проекции. Это значит, что структура классов должна быть построена таким образом, чтобы чётко прослеживалось наследование и связываение этих классов, а формализация не должна выходить за рамки возможностей, предоставляемых ОО-проекциями.

Приведём выдержку из спецификации IMS QTI, которая описывает понятие «теста»:

Class : assessmentItem
Attribute : identifier [1]: string
Attribute : title [1]: string The title of an assessmentItem is intended to enable the item to be selected in situations where the full text of the itemBody is not available, for example when a candidate is browsing a set of items to determine the order in which to attempt them. Therefore, delivery engines may reveal the title to candidates at any time but are not required to do so.
Attribute : label [0..1]: string256
Attribute : lang [0..1]: language
Attribute : adaptive [1]: boolean = false Items are classified into Adaptive Items and Non-adaptive Items.
Attribute : timeDependent [1]: boolean
Attribute : toolName [0..1]: string256 The tool name attribute allows the tool creating the item to identify itself. Other processing systems may use this information to interpret the content of application specific data, such as labels on the elements of the items itemBody.
Attribute : toolVersion [0..1]: string256 The tool version attribute allows the tool creating the item to identify its version. This value must only be interpreted in the context of the

toolName
Contains : responseDeclaration [*]
Contains : outcomeDeclaration [*]
Contains : templateDeclaration [*]
Contains : templateProcessing [0..1]
Contains : stylesheet [0..*]
Contains : itemBody [0..1]
Contains : responseProcessing [0..1]
Contains : modalFeedback [*]

Формализация понятия «тест» в виде формулы дескриптивной логики представлена на рис. 1. Выражение $(n..m)\ r$ является сокращением для формулы $\geq_n r \sqcap \leq_m r$. Выражение $(n..)\ r$ – сокращение для $\geq_n r$, а $(=n)\ r$ – для $\geq_n r \sqcap \leq_n r$.

$$
\begin{aligned}
&assessmentItem \equiv \\
&\quad (=1)\ identifier \sqcap (=1)\ title \sqcap \\
&\quad (0..1)\ label \sqcap (0..1)\ lang \sqcap (=1)\ adaptive \sqcap (=1)\ timeDependent \sqcap \\
&\quad (0..1)\ toolName \sqcap (0..1)\ toolVersion \sqcap responseDeclaration \sqcap \\
&\quad outcomeDeclaration \sqcap templateDeclaration \sqcap \\
&\quad (0..1)\ templateProcessing \sqcap (0..)\ stylesheet \sqcap \\
&\quad (0..1)\ itemBody \sqcap (0..1)\ responseProcessing \sqcap modalFeedback
\end{aligned}
$$

Рис. 1. Определение класса «вопрос» в $\mathcal{SHOIN}(D)$

Формализовав подобным образом все элементы спецификации, кроме атомарных (не имеющих ни атрибутов, ни связанных элементов, кроме тех, что уже формализованы), мы полностью опишем стандарт IMS QTI с помощью $\mathcal{SHOIN}(D)$. Атомарные элементы будут являться членами в правых частях формализующих выражений. Опишем такую формализацию.

Несложно проверяется, что если опустить ограничения кардинальности, то приведённое выше описание класса «вопрос» соответствует определению ОО-проекции. Правая часть данного выражения полностью соответствует определению ОО-концепта.

$$C_1 \sqcap \ldots \sqcap C_f \sqcap \forall R_1.E_1 \sqcap \ldots \sqcap \forall R_g.E_g \sqcap \forall P_1.D_1 \sqcap \ldots \sqcap \forall P_h.D_h$$

с примитивными ролями и атрибутами R_i и P_i. А значит, всё выражение соответствует ОО-определению вида

$$C \equiv E$$

Согласно спецификации IMS QTI, существует несколько независимых абстрактных классов, от которых наследуются основные классы,

использующиеся для создания вопросов, вариантов ответов, оценки и прочего. Они, в свою очередь, расширяются классами уровня ниже и т.д. Ацикличность $TBox_o$, состоящего из описаний этих классов/концептов, проверяется непосредственно.

Чтобы проверить соответствие условию 2 определения ОО-проекции, отметим, что сама формализация IMS укладывается в рамки ОО-проекций, а это значит, что $\mathcal{K}$ и $\mathcal{K}_o$ совпадают, что тривиальным образом влечет $\mathcal{K} \vdash Ax$ для каждой аксиомы $Ax \in TBox_o$.

Доказательство того, что любая роль и атрибут из словаря $\mathcal{V}_o$ входит в ОО-определения $TBox_o$ не более одного раза следует из самой структуры спецификации. Очевидно, что если каждый класс расширяет класс более высокого уровня, а тот, в свою очередь наследует от одного из независимых абстрактных классов, то атрибуты и роли будут входить в ОО-определение $TBox_o$ не более одного раза.

Утверждение $ABox_o \subseteq ABox_{\mathcal{K}}$ из условия 4 определения доказывается аналогично доказательству условия 2. Истинность подпунктов условия 4 определения ОО-проекции для указанной формализации следует из структуры специфкации. Формализация спецификации IMS QTI построена таким образом, что каждый его класс, кроме «корневых», расширяет родительский, а следовательно, выполняется:

- для любого $C(O) \in ABox_o$: $O \in \{O_1, \ldots, O_l\}$ и $C \in \mathcal{C}_o$, причем для каждого O такой C является единственным;
- для любого $R(O, O') \in ABox_o$: $O, O' \in \{O_1, \ldots, O_l\}$, $R \in \mathcal{R}_o$ такие, что $C(O), C'(O') \in ABox_o$, причем $\mathcal{K}_o \models C \sqsubseteq C_1$, $\mathcal{K}_o \models C' \sqsubseteq C'_1$ и $\theta(R) = [C_1, C'_1]$;
- для любого $P(O, v) \in ABox_o$: $O \in \{O_1, \ldots, O_l\}$, $P \in \mathcal{P}_o$, $C(O) \in ABox_o$ и $v \in |D|$, где $\theta(P) = [C_1, D]$ и $\mathcal{K}_o \models C \sqsubseteq C_1$.

Это завершает обоснование того, что наша формализация спецификаций IMS представляет собой ОО-проекцию. Это означает, что к этой формализации могут быть применены методы обработки данных и знаний, действующие в рамках ОО-проекций, и реализованные, в частности, в системе OntoBox.

4. Методы и интерфейсы

После того, как спецификация была формализована и транслированна в онтологию, необходимо создать интерфейсы и методы для работы с полученной онтологией. Эти методы и интрефейсы носят общий характер и необходимы разработчикам для работы со структурой классов и самими классами, включая операции добавления, удаления и изменения классов и структур. Эти методы и интерфейсы должны быть максимально прозрачны и просты, чтобы сторонние разработчики могли быстро и легко понять их и использовать в своей работе. Данный

подход соответствует общей концепции логической архитектуры [4], которая сочетает «расслоение» базового логического формализма на слои (страты) по уровню сложности и выразительности, и при этом каждая страта снабжается методами и интерфейсами работы с данными, описываемой на данном уровне.

Разрабатанная нами система методов и интерфейсов, построенных на формализации спецификаций IMS, реализована в объектно-ориентированном стиле. Это позволяет приблизить «элитную» логическую формализацию к «обычным» разработчикам – через формирование привычной для них схемы взаимодействия, и сокрытие сложностей логической формализации от тех, кто не знаком с тонкостями дескриптивных логик. Фактически получается так, что на базе логической формализации строится привычная объектно-ориентированная программистская модель предметной области.

Погрузив формализацию спецификаций в OntoBox, мы получим возможность использовать в качестве интерфейсов его язык запросов – OntoBox QL [3]. С помощью языка запросов происходит взаимодействие внешних агентов с терминологическими онтологиями IMS и онтологиями конкретных тестов. Таким образом, достигаются две цели:

1. перед разработчиком предстанет привычная объектно - ориентированная модель, инкапсулирующая логические структуры и операции, в которых пользователю разбираться совершенно не обязательно;
2. сама структура классов и объектов будет в существенной степени защищена от ошибок разаработчика, поскольку онтология содержит информацию, позволяющую верифицировать структуры и методы в соответствии с формализованной спецификацией.

Основные методы работы с онтологиями IMS содержатся в разработанной нами библиотеке QTI-Lib [8]. Система основных методов содержит:

Метод генерации терминологических онтологий IMS QTI в OntoBox. Для этого описание спецификации в виде выражений $\mathcal{SHOIN}(D)$ переводится с помощью OnotoBox в онтологическое описание. Для каждого класса онтологии создаётся свой класс на объектно-ориентированном языке программирования, который в дальнейшем разработчик может использовать для своих целей.

Метод импорта XML/QTI–описания тестов и трансляции тестов в объекты отдельной онтологии, погруженной в OntoBox. Как упоминалось выше, тесты в формате IMS QTI хранятся в виде xml-файлов. Необходимо эти тесты сформировать в виде отдельной онтологии, которая будет наследовать общие знания о тестах из терминологических онтологий IMS. Импортирующий метод создает конкретные экземпляры тестов, вопросов и других объектов, действующих в рамках терминоло-

гии IMS QTI и наследующих «знания» о тестах из терминологических онтологий, сформированных предыдущим методом.

Метод экспорта объектов из онтологии QTI, хранящейся в OntoBox, в тесты в виде XML/QTI. Этот метод обратен предыдущему. Некоторую сложность в его реализации определяет то, что «прямая» трансляция объектов в xml-дерево не всегда даёт корректный с точки зрения формата IMS QTI, xml-документ. Это объясняется тем, что xml-дерево, образно говоря, «плоское», а онтология «объёмная». Поэтому в процессе экспорта приходится программно учитывать особенности трансляции онтологичского класса и его объекта в xml-сущность.

Метод выборки компонента «тест» из онтологии – для дальнейшей обработки в образовательных системах. Этот метод позволяет из множества тестов, «погружённых» в онтологии, выбрать именно один, «собрав» его из составных частей типа вопросов, вариантов ответов и т.д.

Метод поддержки генерации компонента «тест» (помогает составлять шаблоны тестов из маленьких частей тестов QTI). Этот метод предоставляет дополнительные инструменты для создания тестов из вопросов, вариантов ответов, реакций на действия пользователя и т.д. Все эти компоненты описаны в виде классов и объектов онтологий.

Онтологии вместе с этими методами служат общей платформой для построения систем тестирования. Существенным качеством этой платформы является то, что она допускает многократное использование и распространение. Взаимодействие между OntoBox и методами может быть организовано как непосредственно, через собственный интерфейс библиотеки QTI-lib, так и с помощью языка запросов OntoBox QL. Здесь можно использовать клиент - серверную технологию, организовав обмен запросами по протоколу HTTP. Отметим, что на основе данной библиотеки была разработана практическая система тестирования в рамках научно-образовательного портала http://lake.baikal.ru.

Автор благодарен А.В. Манциводе за поддержку в работе и ценные замечания.

Список литературы

1. The Semantic Web http://www.w3.org/2001/sw/
2. Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider (Eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press 2003.
3. A.Malykh, and A. Mantsivoda. A Query Language for Logic Architectures. Proceedings of 7th International Conference 'Perspectives of System Informatics', June 2009.
4. А.А. Малых, А.В. Манцивода, В.С. Ульянов / Логические архитектуры и объектно-ориентированный подход // Вестник НГУ. Серия: Математика, механика, информатика. 2009. Т9. Вып. 3. С. 64-85

5. Web Ontology Language (OWL). www.w3.org/2004/OWL
6. Ian Horrocks, and Peter F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability // In: Fensel, D., Sycara, K. and Mylopoulos, J., eds., Proc. of the 2003 International Semantic Web Conference (ISWC 2003). - number 2870 in Lecture Notes in Computer Science. - p.17-29. - Springer.
7. IMS Global Learning Consortium. http://imsglobal.org/
8. IMSQ QTI & Ontologies http://meta2project.org/ru/qti.html

N. O. Stukushin
Logical Formalizations of Specifications Based on OO-projections

Abstract. In this paper the formalization development problem for standards and specifications is considered. Such formalizations are useful at both specification development and specification application phases. In particular, if a formalization is made in a constructive way, it can serve as a basis for the implementation of relevant information systems. In this paper an approach to specification formalization is developed, which is based on special sub-logics of description logics. The approach is illustrated on the example of the specifications of the international educational consortium IMS.

Keywords: ontology, description logic, formalization, specification, OO-projection, semantic web

Стукушин Никита Олегович, Институт математики,экономики и информатики, Иркутский государственный университет, 664000, Иркутск, ул. К. Маркса, 1 тел.: (3952)242210 (stukushin@baikal.ru)

Nikita Stukushin, Irkutsk State University, 1, K. Marks St., Irkutsk, 664003 Phone: (3952)242210 (stukushin@baikal.ru)