

ПРИКЛАДНЫЕ ОНТОЛОГИИ ПРОЕКТИРОВАНИЯ

УДК 004.89

DOI: 10.18287/2223-9537-2021-11-2-154-169

Формирование базы знаний для поддержки процесса архитектурного проектирования программных систем

Г.Ю. Гуськов, А.М. Наместников, А.А. Романов, А.А. Филиппов

Ульяновский государственный технический университет, Ульяновск, Россия

Аннотация

Рассмотрен подход к формированию базы знаний для автоматизации процесса архитектурного проектирования программных систем на основе опыта предыдущих проектов. Под архитектуризацией понимается представление программных систем в форме описания архитектуры, состоящей из множества артефактов проектирования. При разработке новой программной системы возможно повысить её качество за счёт привлечения опыта предыдущих проектов - удачных проектных и архитектурных решений, содержащихся в базе знаний проектной организации. Такая база знаний должна формироваться в процессе анализа артефактов проектирования, полученных в процессе работы над предыдущими проектами: исходный код проекта, диаграммы проектирования, модели данных, структурированные информационные ресурсы и т. д. В статье рассмотрена модель базы знаний проектной организации, позволяющая учитывать опыт предыдущих проектов. Представлена модель прикладного решения платформы 1С: Предприятие в качестве примера артефакта проектирования. Представлен метод для формирования фрагментов базы знаний в процессе анализа прикладного решения для платформы 1С и метод генерации диаграмм вариантов использования на основе содержимого базы знаний. Приведены результаты экспериментов оценки качества по точности (наличие в сгенерированной диаграмме элементов экспертной диаграммы) и полноте (наличие в сгенерированной диаграмме элементов, отсутствующих в экспертной диаграмме). Полученные оценки значений точности в среднем составляет 0.875, а полноты – 0.6.

Ключевые слова: архитектуризация, программная система, артефакт проектирования, база знаний, проектный опыт.

Цитирование: Гуськов, Г.Ю. Формирование базы знаний для поддержки процесса архитектурного проектирования программных систем / Г.Ю. Гуськов, А.М. Наместников, А.А. Романов, А.А. Филиппов // Онтология проектирования. – 2021. – Т.11, №2(40). - С.154-169. – DOI: 10.18287/2223-9537-2021-11-2-154-169.

Введение

В настоящее время сложность разработки и поддержки программных систем (ПС) достигла уровня, при котором необходимо применять принципы и процедуры процесса архитектуризации [1]. Под *архитектуризацией* понимается представление ПС в форме описания архитектуры, состоящей из множества артефактов проектирования. Свойства ПС определяют поведение системы, её состав и подходы к развитию, влияющие на выполнимость, полезность и жизнеспособность системы [2-4]. Описание архитектуры ПС, полученное в процессе архитектурного проектирования, используется для координации действий сторон, вовлечённых в процессы разработки, внедрения, поддержки и использования системы. В процессе архитектурного проектирования ПС формируются требования к системе на всех стадиях жизненного цикла, выделяются особенности процессов разработки, внедрения, эксплуатации и сопровождения ПС.

Архитектуру ПС можно рассматривать как целостную концепцию основных свойств системы, воспринимаемую наилучшим образом через деловое, физическое и техническое представления этой архитектуры [1]. Описание архитектуры ПС определяется как совокупность артефактов проектирования, полученных в процессе разработки ПС. Артефакты проектирования – элементы, составляющие описание архитектуры ПС.

Представление архитектуры ПС в виде множества артефактов проектирования позволяет систематизировать описание архитектуры ПС с позиций заинтересованных сторон в рамках предметных областей (ПрО). Особенности ПрО формируют контекст, в рамках которого разрабатывается и функционирует ПС. Контекст ПС определяет множество архитектурных требований и ограничений, непосредственно влияющих на состав и характеристики артефактов проектирования [5]:

- определение основных компонентов и элементов ПС;
- подходы к организации и взаимодействию элементов ПС;
- принципы организации ПС;
- принципы организации проекта ПС;
- принципы развития ПС в рамках её жизненного цикла.

Качество описания архитектуры ПС влияет на качество процесса разработки и качество системы [6, 7]. Чем на более позднем этапе разработки ПС обнаружены ошибки проектирования, тем больше ресурсов требуется для их исправления.

Описание архитектуры ПС также может быть получено в процессе переработки описания архитектуры предыдущих проектов [1, 4]. Удачные проектные решения, полученные при работе над предыдущими проектами, должны быть использованы в процессе работы над новыми проектами.

1 Постановка задачи

Современная проектная организация обладает значительным по объёму проектным заделом, состоящим из артефактов проектирования, полученных в процессе работы над предыдущими проектами. Артефакты проектирования содержат в себе опыт и знания (единицы проектного опыта, ЕПО) большого числа специалистов, которые на протяжении многих лет занимались разработкой ПС. Под ЕПО понимаются удачные проектные и архитектурные решения, полученные при работе над предыдущими проектами.

ЕПО зафиксированы в различных артефактах проектирования:

- проектная документация, представленная в виде слабоструктурированного текста на естественном языке;
- проектные диаграммы, обычно представленные в нотации *UML*;
- исходные коды;
- модели данных объектов ПрО, представленные в виде сущностей и связей реляционных баз данных;
- различные структурированные информационные ресурсы, описывающие особенности разработки, развёртывания и настройки окружения проекта, обычно представленные в виде вики-ресурсов.

При увеличении количества гетерогенных неструктурированных данных затрудняется поиск необходимых сведений. В результате опыт предыдущих разработок, зафиксированный в артефактах проектирования, остаётся невостребованным, и, как следствие, увеличивается время выполнения цикла разработки ПС, снижается качество конечного продукта.

В работах [8, 9] отмечается, что на начальных этапах разработки новых проектов важным является использование опыта предыдущих разработок. Решение указанной проблемы может

основываться на применении интеллектуальных методов и алгоритмов анализа артефактов проектирования с целью построения базы знаний (БЗ) проектной организации.

Разработка ПС состоит из нескольких этапов, качество выполнения которых влияет на качество системы [6, 7]. Основным этапом разработки новой ПС является проектирование, в рамках которого формируется описание архитектуры и модель разрабатываемой системы. На данном этапе должны быть решены следующие задачи:

- поиск критических участков проекта;
- формирование окончательного описания архитектуры ПС;
- объектное моделирование и проектирование основных элементов ПС.

Форматы представления и методы хранения артефактов проектирования различны, что затрудняет их анализ и формирование фрагментов БЗ. Полученная в результате такого анализа БЗ проектной организации может быть применена для:

- накопления множества проектных решений, используемых в проектной организации;
- извлечения часто используемых ЕПО;
- анализа качества полученных ЕПО;
- определения предпочтений в выборе ЕПО разработчиками;
- определения показателя изменения динамики интереса к различным ЕПО во времени;
- прототипирования проекта на основе анализа технического задания или спецификации и множества ЕПО, содержащихся в БЗ.

Учёт специфики проектного опыта, полученного при работе над предыдущими проектами, приводит к необходимости формирования БЗ проектной организации особой структуры, основанной на системе онтологий.

2 База знаний проектной организации

Исследователи в области инженерии знаний отмечают актуальность исследований, основанных на онтологическом подходе [10-13], а в работах [14-18] - важность онтологического инжиниринга в процессах проектирования и разработки ПС. Так, в [14, 17] предлагается использовать вместо традиционных языков моделирования ПС (например *UML*) онтологии, которые позволяют контролировать логическую целостность и непротиворечивость полученной модели. Однако существующие методы формирования БЗ в виде набора онтологий для поддержки и автоматизации процесса архитектуризации ПС предполагают привлечение экспертов ПрО и специалистов в области инженерии знаний, что требует существенных временных затрат на формирование такой БЗ.

Следовательно, актуальной является разработка моделей и методов построения БЗ проектной организации в процессе автоматизированного анализа артефактов проектирования, полученных в процессе работы над предыдущими проектами. Сложность здесь заключается в необходимости унификации артефактов проектирования для формирования БЗ, являющейся основой информационного обеспечения комплекса средств автоматизации архитектурного проектирования ПС, а также в необходимости включения в БЗ сведений о различных ПрО, в рамках которых осуществлялась работа над предыдущими проектами.

2.1 Онтология ПрО

Онтология ПрО позволяет адаптировать БЗ проектной организации к особенностям ПрО, в рамках которых проводилась работа над предыдущими проектами. Онтология ПрО выступает семантическим базисом, позволяющим сформировать единый понятийный аппарат, объединяющий и унифицирующий сущности различных ПрО.

Онтология ПрО представляется в виде множества:

$$O^{dom} = \{O^{dom}, \dots, O_i^{dom}, \dots, O_n^{dom}\}, \quad (i = 1, \dots, n), \quad (1)$$

где n – количество ПрО, в рамках которых проводилась работа над предыдущими проектами; O_i^{dom} – i -я онтология ПрО, содержащая описание особенностей i -й ПрО. Формальное представление любой O^{dom} запишется в виде кортежа:

$$O^{dom} = \langle C^{dom}, T^{dom}, R^{dom}, F^{dom} \rangle, \quad (2)$$

где C^{dom} – множество понятий ПрО;

T^{dom} – множество терминов ПрО, описывающих понятия ПрО;

F^{dom} – множество функций интерпретации, определяющих множество отношений R^{dom} ;

R^{dom} – множество отношений вида:

$$R^{dom} = \{R_{CC}^{dom}, R_{CT}^{dom}\},$$

где R_{CC}^{dom} – множество отношений обобщения, определяющих иерархию понятий ПрО («понятие-понятие»);

R_{CT}^{dom} – множество отношений ассоциации, формирующих связи между понятиями и терминами, описывающими данные понятия («понятие-термин»).

2.2 Онтология артефактов проектирования

Онтология артефактов проектирования содержит формализованные описания артефактов проектирования в виде фрагментов БЗ проектной организации. Базовыми элементами данной онтологии являются:

- *сущность* – объект некоторой ПрО, смоделированный с помощью артефакта проектирования, например, класс, таблица и т.д.;
- *атрибут сущности* – свойство сущности, например, поле класса, столбец таблицы и т.п.;
- *ограничение атрибута сущности* – свойства атрибута сущности, например, тип поля класса, размер поля класса с типом «строка» и т.д.;
- *бизнес-процесс* – процесс в некоторой ПрО, в который вовлечена некоторая сущность, например, метод класса, вариант использования и т.д.

Онтология артефактов проектирования представляется следующим кортежем:

$$O^{exp} = \langle E^{exp}, A^{exp}, C^{exp}, R^{exp}, F^{exp} \rangle, \quad (3)$$

где E^{exp} – множество сущностей;

A^{exp} – множество атрибутов сущностей;

C^{exp} – множество ограничений атрибутов сущностей;

R^{exp} – множество бизнес-процессов, в которые вовлечены сущности;

F^{exp} – множество функций интерпретации, определяющее множество отношений R^{exp} ;

R^{exp} – множество отношений вида:

$$R^{exp} = \{R_{EE}^{exp}, R_{EA}^{exp}, R_{AC}^{exp}, R_{EP}^{exp}\},$$

где R_{EE}^{exp} – множество отношений обобщения, определяющие иерархию сущностей («сущность-сущность»);

R_{EA}^{exp} – множество отношений ассоциации, определяющие связь между сущностью и её атрибутом («сущность-атрибут»);

R_{AC}^{exp} – множество отношений ассоциации, определяющие связь между атрибутом сущности и его ограничением («атрибут-ограничение»);

R_{EP}^{exp} – множество отношений ассоциации, определяющие связь между сущностью и бизнес-процессом, в который она вовлечена («сущность - бизнес-процесс»).

2.3 Модель базы знаний проектной организации

Рассмотренные в подразделах 2.1 и 2.2 онтологии формируют компоненты БЗ проектной организации. Модель БЗ проектной организации можно представить в виде кортежа:

$$O = \langle O^{dom}, O^{exp}, R, F \rangle, \quad (4)$$

в котором O^{dom} – онтология для учёта сведений о различных ПрО, в рамках которых производилась работа над предыдущими проектами (1);

O^{exp} – онтология для представления артефактов проектирования, формируемая в процессе анализа артефактов проектирования (3);

R – множество отношений ассоциации между понятиями онтологии ПрО и артефактами проектирования;

F – множество функций интерпретации, определяющих множество отношений R .

3 Формирование фрагментов базы знаний в процессе анализа артефактов проектирования

Формирование фрагментов БЗ в процессе анализа структуры прикладного решения разработано с применением платформы 1С: Предприятие 8 [19]. Платформа 1С широко используется для автоматизации различных типов учёта: управленческий, оперативный, бухгалтерский и т.д. Для каждого типа учёта разрабатываются типовые прикладные решения, выполняемые в рамках платформы 1С. Прикладное решение платформы 1С представляет собой комплексный артефакт проектирования, содержащий опыт и знания большого числа высококвалифицированных специалистов. Прикладное решение платформы 1С можно представить в виде следующего выражения:

$$Conf = \{S_1^{Conf}, \dots, S_i^{Conf}, \dots, S_n^{Conf}\}, i = 1, \dots, n,$$

где S_i^{Conf} – i -я подсистема прикладного решения. Подсистема используется для группировки объектов прикладного решения в рамках некоторого контекста, например: «Закупки», «Кадры», «Склад» и т. д. Подсистему можно представить в виде следующего выражения:

$$S_i^{Conf} = (Name, E^{Conf}, P^{Conf}), \quad (5)$$

где $Name$ – уникальное имя подсистемы;

$E^{Conf} = \langle C^{Conf}, N^{Conf} \rangle$ – множество сущностей прикладного решения; для представления сущностей ПрО в рамках прикладного решения 1С используется множество справочников C^{Conf} и множество перечислений N^{Conf} ;

$P^{Conf} = \langle D^{Conf}, R^{Conf} \rangle$ – множество бизнес-процессов прикладного решения, в которые вовлечены сущности прикладного решения. Множество бизнес-процессов P^{Conf} содержит множество документов D^{Conf} и множество регистров R^{Conf} . Документы используются для изменения состояния сущностей прикладного решения. Регистры прикладного решения используются для хранения истории состояний сущностей. Записи регистров формируются на основе данных документов.

Каждый элемент из множеств C^{Conf} и D^{Conf} можно представить в виде следующего выражения:

$$C_i^{Conf}, D_i^{Conf} = \langle Name, V^{Conf}, T^{Conf} \rangle,$$

где $V^{Conf} = \{V_1^{Conf}, \dots, V_i^{Conf}, \dots, V_m^{Conf}\}$ – множество атрибутов сущности;

$V_i^{Conf} = \langle Name, Type, Constraint \rangle$ – каждый атрибут элемента характеризуется именем $Name$, типом данных $Type$ и ограничениями $Constraint$. При этом тип данных $Type$ может представлять собой не только простой тип данных, но и ссылку на другой объект прикладного решения (элемент справочника, документ и т. д.);

$T^{Conf} = \{T_1^{Conf}, \dots, T_i^{Conf}, \dots, T_m^{Conf}\}$ – множество таблиц, связанных с сущностью. Таблицы в рамках платформы 1С используются для формирования связей вида «один ко многим» для группировки данных рамках одной сущности, например, элемент справочника «Сотрудник» может содержать таблицу со списком его детей; $T_i^{Conf} = \langle Name, V^{Conf} \rangle$ – каждая таблица сущности содержит уникальное имя и множество атрибутов.

Элемент множества R^{Conf} можно записать как $R_i^{Conf} = \langle Name, V^{Conf} \rangle$, а элемент множества N^{Conf} имеет вид $N_i^{Conf} = \langle Name \rangle$

Таблица 1 содержит соответствие объектов – i -й подсистемы прикладного решения и аксиом онтологии артефактов проектирования.

Алгоритм формирования фрагментов БЗ представлен на примере анализа прикладного решения платформы 1С следующей структуры:

- подсистемы: «Торговля».
- справочники: «Товары», «ЕдиницыИзмерения», «Склады», «Поставщики».
- документы: «ПоставкаТовара».
- атрибуты справочника «Товары»: «Код» (строка), «Наименование» (строка), «ЕдиницаИзмерения» (элемент справочника «ЕдиницыИзмерения»).
- атрибуты справочника «ЕдиницыИзмерения»: «Код» (строка), «Наименование» (строка).
- атрибуты справочника «Склады»: «Код» (строка), «Наименование» (строка).
- атрибуты справочника «Поставщики»: «Код» (строка), «Наименование» (строка).
- атрибуты документа «ПоставкаТовара»: «Склад» (элемент справочника «Склады»), «Поставщик» (элемент справочника «Поставщики»), «Товары» (таблица).
- атрибуты таблицы «Товары»: «Товар» (элемент справочника «Товары»), «Цена» (число), «Количество» (число), «Сумма» (число).

Алгоритм формирования фрагментов БЗ в процессе анализа прикладного решения платформы 1С можно записать в виде следующей последовательности шагов.

- 1) для каждой подсистемы прикладного решения сформировать сущность онтологии артефактов проектирования:

$$S_i^{Conf} \rightarrow E_i^{exp},$$

например, Торговля $\in E^{exp}$.

- 2) для каждой сущности текущей подсистемы прикладного решения создать сущность онтологии артефактов проектирования с учётом связи с подсистемой:

$$C^{Conf} \rightarrow \hat{E}^{exp} \in E^{exp}, \hat{R}_{EE}^{exp} \in R_{EE}^{exp},$$

$$N^{Conf} \rightarrow \hat{E}^{exp} \in E^{exp}, \hat{R}_{EE}^{exp} \in R_{EE}^{exp},$$

например, Товары $\in E^{exp}$, ЕдиницыИзмерения $\in E^{exp}$,
Склады $\in E^{exp}$, Поставщики $\in E^{exp}$,
Товары R_{EE}^{exp} Торговля, ..., Поставщики R_{EE}^{exp} Торговля.

Таблица 1 – Соответствие объектов подсистемы прикладного решения и аксиом онтологии O^{exp}

Прикладное решение	Онтология
$S_i^{Conf}.Name$	E_i^{exp}
$C_i^{Conf}.Name$	$E_i^{exp}, R_{EE_i}^{exp}$
$N_i^{Conf}.Name$	$E_i^{exp}, R_{EE_i}^{exp}$
$T_i^{Conf}.Name$	$E_i^{exp}, R_{EE_i}^{exp}$
$D_i^{Conf}.Name$	$P_i^{exp}, R_{EE_i}^{exp}$
$R_i^{Conf}.Name$	$P_i^{exp}, R_{EE_i}^{exp}$
$V_i^{Conf}.Name$	$A_i^{exp}, R_{EA_i}^{exp}$
$V_i^{Conf}.Type, V_i^{Conf} \in C_j^{Conf}$	$(C_i^{exp}, R_{AC_i}^{exp})$
$V_i^{Conf}.Type, V_i^{Conf} \in D_j^{Conf}$	$R_{EP_i}^{exp}$
$V_i^{Conf}.Constraint$	$C_i^{exp}, R_{AC_i}^{exp}$

- 3) если текущая сущность прикладного решения является справочником, создать множество атрибутов, ограничений атрибутов и отношений для данной сущности онтологии артефактов проектирования:

$$V^{Conf} \in C_i^{Conf} \rightarrow \hat{A}^{exp} \in A^{exp}, \hat{R}_{EA}^{exp} \in R_{EA}^{exp},$$

$$V^{Conf} \in C_i^{Conf} \rightarrow \hat{C}^{exp} \in C^{exp}, \hat{R}_{AC}^{exp} \in R_{AC}^{exp},$$

например, Код $\in A^{exp}$, Наименование $\in A^{exp}$, ЕдиницаИзмерения $\in A^{exp}$,
Товары R_{EA}^{exp} Код, Товары R_{EA}^{exp} Наименование, Товары R_{EA}^{exp} ЕдиницаИзмерения,

...,
Поставщики R_{EA}^{exp} Код, Поставщики R_{EA}^{exp} Наименование,
СтрокаТип $\in C^{exp}$, ЧислоТип $\in C^{exp}$, СсылкаТип $\in C^{exp}$,
Код R_{AC}^{exp} СтрокаТип, Наименование R_{AC}^{exp} СтрокаТип,
ЕдиницаИзмерения R_{AC}^{exp} СсылкаТип.

- 4) если текущая сущность прикладного решения имеет таблицы (справочник), то для каждой таблицы сформировать сущность онтологии артефактов проектирования, множество атрибутов, ограничений атрибутов и отношений для данной сущности:

$$T^{Conf} \in C_i^{Conf} \rightarrow \hat{E}^{exp} \in E^{exp}, \hat{R}_{EE}^{exp} \in R_{EE}^{exp},$$

$$V^{Conf} \in T_j^{Conf}, T_j^{Conf} \in C_i^{Conf} \rightarrow \hat{A}^{exp} \in A^{exp}, \hat{R}_{EA}^{exp} \in R_{EA}^{exp},$$

$$V^{Conf} \in T_j^{Conf}, T_j^{Conf} \in C_i^{Conf} \rightarrow \hat{C}^{exp} \in C^{exp}, \hat{R}_{EC}^{exp} \in R_{EC}^{exp}.$$

- 5) для каждого бизнес-процесса текущей подсистемы прикладного решения создать бизнес-процесс онтологии артефактов проектирования с учётом связи с подсистемой:

$$D^{Conf} \rightarrow \hat{P}^{exp} \in P^{exp}, \hat{R}_{EE}^{exp} \in R_{EE}^{exp},$$

$$R^{Conf} \rightarrow \hat{P}^{exp} \in E^{exp}, \hat{R}_{EE}^{exp} \in R_{EE}^{exp},$$

например, ПоставкаТовара $\in P^{exp}$, ПоставкаТовара R_{EE}^{exp} Торговля.

- 6) создать множество отношений для связи текущего бизнес-процесса и сущностей онтологии артефактов проектирования на основе анализа атрибутов текущего бизнес-процесса прикладного решения:

$$V^{Conf} \in D_i^{Conf} \rightarrow \hat{R}_{EP}^{exp} \in R_{EP}^{exp},$$

$$V^{Conf} \in R_i^{Conf} \rightarrow \hat{R}_{EP}^{exp} \in R_{EP}^{exp},$$

например, Поставщики R_{EP}^{exp} ПоставкаТовара, Склады R_{EP}^{exp} ПоставкаТовара.

- 7) если текущий бизнес-процесс прикладного решения имеет таблицы (документ), то для каждой таблицы сформировать множество отношений для связи текущего бизнес-процесса и сущностей онтологии артефактов проектирования:

$$V^{Conf} \in T_j^{Conf}, T_j^{Conf} \in D_i^{Conf} \rightarrow \hat{R}_{EP}^{exp} \in R_{EP}^{exp},$$

например, Товары R_{EP}^{exp} ПоставкаТовара.

Для представленного прикладного решения платформы 1С формируется онтология ПрО. Например, такая онтология может включать следующие элементы:

Товар $\in C^{dom}$, Склад $\in C^{dom}$, Поставщик $\in C^{dom}$,
ЕдиницаИзмерения $\in C^{dom}$, ПоставкаТовара $\in C^{dom}$,
Товар $\in T^{dom}$, Склад $\in T^{dom}$, Поставщик $\in T^{dom}$,
Единица измерения $\in T^{dom}$, ..., Поставка товара $\in T^{dom}$,
Товар R_{CT}^{dom} , ..., ПоставкаТовара R_{CT}^{dom} Поставка товара.

Полученный в результате анализа прикладного решения платформы 1С фрагмент БЗ в виде онтологии представлен на рисунке 1.

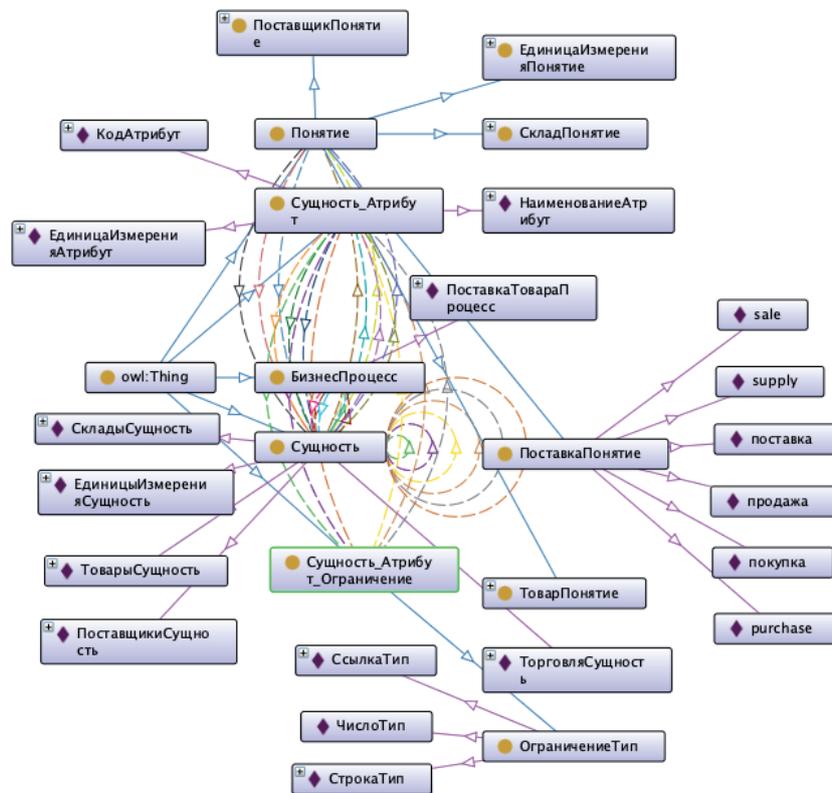


Рисунок 1 – Пример сформированной онтологии

4 Генерация диаграмм вариантов использования на основе базы знаний

Диаграмму вариантов использования можно записать в виде следующего выражения:

$$UCD = \langle A^{UCD}, P^{UCD}, R^{UCD} \rangle, \tag{6}$$

где A^{UCD} – множество акторов диаграммы вариантов использования;

P^{UCD} – множество прецедентов (вариантов использования);

R^{UCD} – множество отношений диаграммы вариантов использования:

$$R^{UCD} = \{R_C^{UCD}, R_I^{UCD}, R_E^{UCD}\},$$

в котором R_C^{UCD} – множество отношений обобщения;

R_I^{UCD} – множество отношений включения;

R_E^{UCD} – множество отношений расширения.

Тогда постановка задачи генерации диаграммы вариантов использования заключается в преобразовании онтологии O^{exp} (3) в элементы модели диаграммы (6).

Таблица 2 содержит описание сопоставления структурных компонентов диаграммы вариантов использования с сущностями онтологии O^{exp} . В настоящий момент при формировании диаграммы вариантов использования на основе содержимого БЗ формируется

Таблица 2 – Соответствие компонентов онтологии артефактов проектирования и диаграммы вариантов использования

Онтология	Диаграмма вариантов использования
P_i^{exp}	Прецедент P_i^{UCD}
P_i^{exp}, P_j^{exp}	Отношение включения R_I^{UCD}

только отношение включения. Для формирования отношений расширения и обобщения требуется дополнительная проработка метода.

Алгоритм генерации диаграммы вариантов использования на основе содержимого БЗ состоит из следующих шагов:

- 1) выбор бизнес-процессов онтологии в качестве основы диаграммы вариантов использования, например, $\text{ПоставкаТовара} \in P^{exp}$;
- 2) формирование списка сущностей онтологии, с которыми связаны выбранные бизнес-процессы, например: $\text{Товары} \in E^{exp}$, $\text{ЕдиницыИзмерения} \in E^{exp}$, $\text{Склады} \in E^{exp}$, $\text{Поставщики} \in E^{exp}$;
- 3) формирование иерархии сущностей онтологии для генерации диаграммы вариантов использования на основе анализа отношений вида R_{EE}^{exp} онтологии, например: $\text{Товары} R_{EE}^{exp}$ Торговля , ..., $\text{Поставщики} R_{EE}^{exp}$ Торговля ;
- 4) замена сущностей полученной иерархии на связанные с ней бизнес-процессы (если такого бизнес-процесса ещё нет в диаграмме);
- 5) обход иерархии с корневого узла. Формирование прецедента для каждого узла, организация отношений включения между прецедентами, например: $\text{Торговля} R_I^{UML}$ Поставка товара и т.д.

После выполнения данного алгоритма формируется набор команд для системы *PlantUML* [20], например:

```
@startuml
    :user: - (Торговля)
    (Торговля)..>(Поставка товара):include
    (Поставка товара)..>(Внести данные Поставщик):include
    (Поставка товара)..>(Внести данные Склад):include
    (Поставка товара)..>(Внести данные Товар):include
    (Поставка товара)..>(Оформить Поставка товара):include
@enduml
```

На рисунке 2 для рассмотренного примера представлен результат генерации диаграммы вариантов использования.

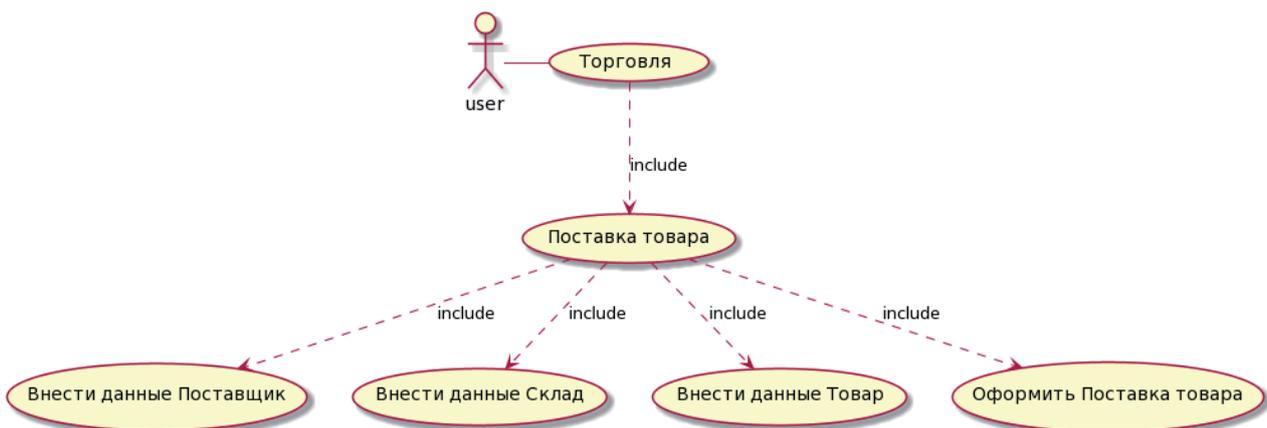


Рисунок 2 – Пример сгенерированной диаграммы вариантов использования

5 Эксперименты

Для оценки адекватности предложенных методов и моделей проведён ряд экспериментов, план которых включает:

- формирование содержимого БЗ проектной организации в виде *OWL* онтологии в процессе анализа прикладного решения для платформы 1С;
- проверка полученной *OWL* онтологии на корректность с применением механизма логического вывода редактора онтологий *Protégé* [21];
- генерация набора команд системы *PlantUML* для формирования диаграмм вариантов использования;
- оценка качества полученных диаграмм вариантов использования путём их сравнения с аналогичными диаграммами, построенными экспертом.

Для оценки качества полученных диаграмм вариантов использования применялись следующие метрики.

Точность:

$$Precision = \frac{|Gen \cap Exp|}{|Exp|},$$

где $|Gen \cap Exp|$ – количество совпадающих элементов в сгенерированной диаграмме и диаграмме, построенной экспертом;

$|Exp|$ – количество элементов в диаграмме, построенной экспертом.

Полнота:

$$Recall = \frac{|Gen - Exp|}{|Gen|},$$

где $|Gen - Exp|$ – количество элементов в сгенерированной диаграмме, которые отсутствуют в диаграмме, построенной экспертом;

$|Gen|$ – количество элементов в сгенерированной диаграмме.

В отличие от аналогичных метрик качества в информационном поиске, при оценке качества диаграмм вариантов использования, полученных с применением предложенного подхода, точность и полнота не являются взаимоисключающими значениями.

6 Оценка качества сгенерированных диаграмм вариантов использования

В качестве входных данных для формирования содержимого БЗ использовалось типовое прикладное решение для платформы 1С – «Управление торговлей».

Формирование *OWL* онтологии проводилось при анализе объектов прикладного решения, входящих в подсистему «Расхождения»:

- подсистемы «Закупки» (родительская подсистема) и «Расхождения»;
- регистр накопления «ГрафикиОтгрузки»;
- справочники «Организации», «Валюты», «Партнеры», «Контрагенты», «Пользователи», «Номенклатура», «Назначения», «Склады»;
- документы «АктОРасхожденияхПослеПеремещенияТоваров», «АктОРасхожденияхПослеПриемкиТоваров» и «АктОРасхожденияхПослеОтгрузкиТоваров».

Проверка полученной *OWL* онтологии в редакторе *Protégé* закончилась успешно.

Для проведения данного эксперимента в редакторе *Protégé* была составлена онтология *PrO* O^{dom} (2), содержащая названия прецедентов для объектов прикладного решения платформы 1С. Например, для понятия «Закупки» установлено описание «Закупка товаров», для понятия «Склады» установлено описание «Данные о складах», а для понятия «АктОРасхожденияхПослеПеремещенияТоваров» описание – «Акт о расхождениях после перемещения».

В сгенерированной на основе содержимого *OWL* онтологии диаграмме вариантов использования (рисунок 3) содержится 14 прецедентов, из которых 9 не содержится в диаграмме, построенной экспертом (рисунок 4). Количество прецедентов диаграммы вариантов использования, построенной экспертом – 5. Таким образом, получены следующие значения метрик качества: $Precision = 5/5=1$, $Recall = 9 / 14 = 0.643$.

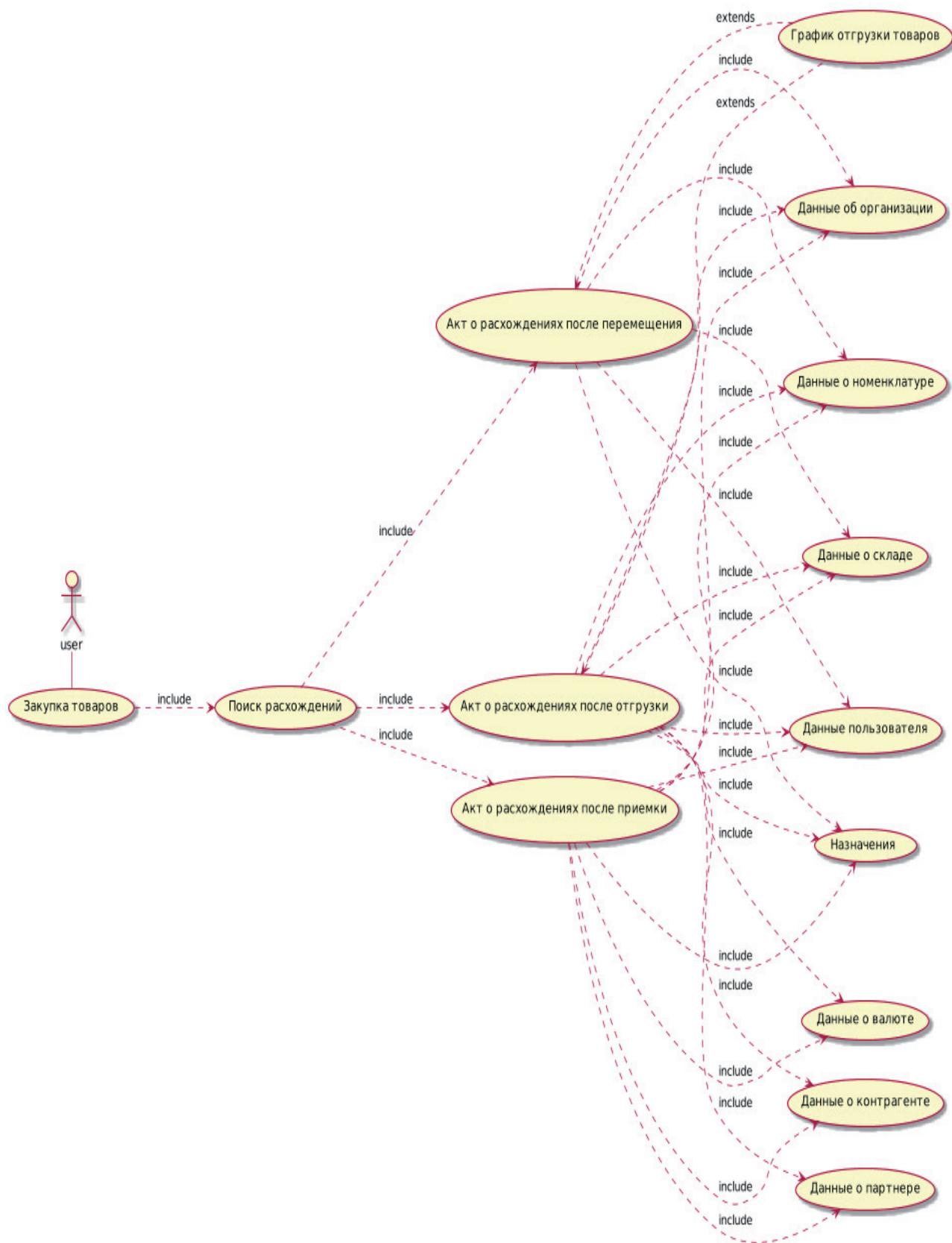


Рисунок 3 – Сгенерированная диаграмма вариантов использования для прикладного решения на платформе 1С «Управление торговлей»

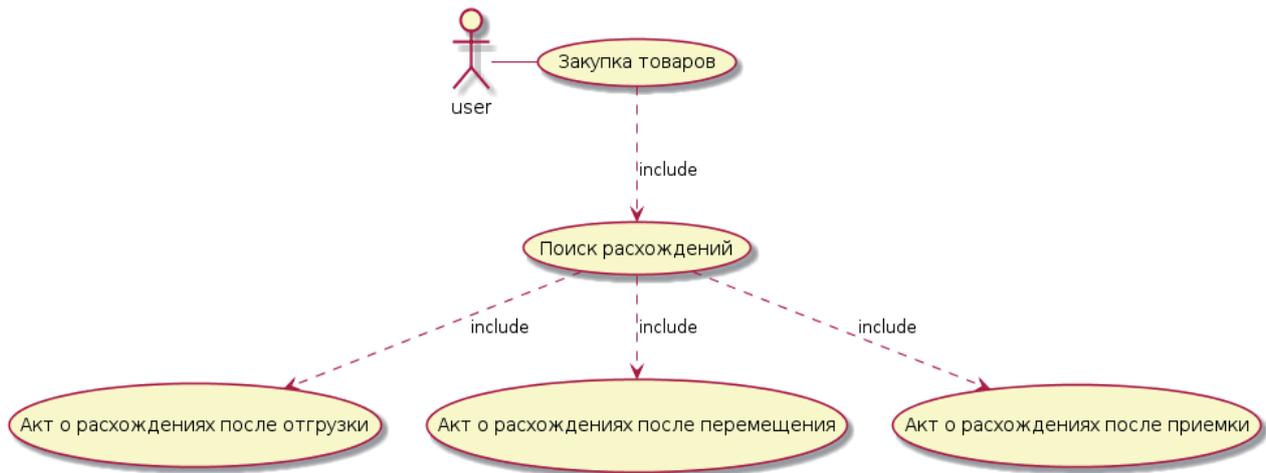


Рисунок 4 – Диаграмма вариантов использования для прикладного решения на платформе 1С «Управление торговлей», построенная экспертом

Время, затраченное на автоматическую генерацию диаграммы вариантов использования, составило 3 минуты (без учёта времени формирования онтологии Про O^{dom}). Экспертом на решение аналогичной задачи было потрачено 17 минут.

Таблица 3 содержит результаты экспериментов по формированию диаграмм вариантов использования на основе содержимого БЗ, сформированной на основе анализа различных прикладных решений для платформы 1С.

Таблица 3 – Результаты проведённых экспериментов

Конфигурация	Подсистема	Кол-во элементов			Точность	Полнота
		Метод	Эксперт	Совпадающие		
Управление торговлей	Расхождения (закупки со сверкой товаров)	14	5	9	1	0.643
Зарплата и управление персоналом	Кадры (регистрация переработок)	8	4	4	1	0.5
Комплексная автоматизация	Склад (излишки, недостачи, порча продукции)	12	6	4	0.667	0.667
Управление торговлей	Банк и касса (работа с кассой)	13	6	5	0.833	0.615

Полученные оценки значений точности в среднем составляет 0.875, а полноты – 0.6. Как видно из результатов экспериментов, сгенерированные артефакты проектирования имеют высокий показатель точности, но при этом такие артефакты проектирования содержат большее количество структурных элементов, что повышает показатель полноты. Высокий показатель полноты связан с учётом связей между объектами анализируемого прикладного решения для платформы 1С, которые получены в процессе их жизненного цикла и не были спроектированы изначально.

Заключение

Предложен подход к автоматизации формирования фрагментов БЗ для поддержки архитектурного проектирования ПС. БЗ представляет собой систему онтологий для учёта особенностей Про и хранения формализованного представления артефактов проектирования, полу-

ченных при работе над предыдущими проектами. Дано описание теоретико-множественной модели БЗ проектов. В качестве примера артефакта проектирования использовано прикладное решение на платформе 1С.

Представлены результаты экспериментов оценки качества предложенных моделей и методов путём сравнения сгенерированных диаграмм вариантов использования с построенными экспертом.

Дальнейшее развитие предложенного подхода видится в работе с различными артефактами проектирования и в выделении ЕПО из содержимого БЗ с применением механизмов логического вывода. Выделение ЕПО в процессе анализа содержимого БЗ проектной организации позволит сократить пространство и время поиска удачных проектных решений.

Благодарности

Исследование выполнено в рамках государственного задания № 075-00233-20-05 по проекту «Исследование интеллектуального предиктивного мультимодального анализа больших данных и извлечения знаний из разных источников», а также при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научных проектов № 19-47-730006, 19-47-730005.

Список источников

- [1] ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011. Системная и программная инженерия. Описание архитектуры. М.: Стандартинформ, 2019. - <http://docs.cntd.ru/document/1200139542>.
- [2] **Heesch van, U.** A documentation framework for architecture decisions / U. van Heesch, P. Avgeriou, R. Hilliard // *Journal Syst. Softw.* 2012. vol. 85, 4, p.795-820. DOI:10.1016/j.jss.2011.10.017.
- [3] **Hilliard, R.** Using Aspects in Architectural Description / R. Hilliard // *Lecture Notes in Computer Science*, 2007. vol. 4765. P.65-68.
- [4] **Sosnin, P.** Substantially evolutionary theorizing in designing software-intensive systems / P. Sosnin // *Information (Switzerland)*, 2018. 9 (4). P.91.
- [5] **Sosnin, P.** Ontological controlling the lexical items in conceptual solution of project tasks / P. Sosnin, A. Pushkareva // *Lecture Notes in Computer*, 2017. Vol. 10409. P.31-46.
- [6] ISO/IEC 25010 2011. ISO/IEC 25010:2011: Systems And Software Engineering – Systems And Soft-Ware Quality Requirements And Evaluation (Square) – System And Software Quality Models.
- [7] **Макконнелл, С.** Совершенный код: Практическое руководство по разработке программного обеспечения / С. Макконнелл. СПб.: Русская редакция/БХВ, 2017.
- [8] **Маклаев, В.А.** Инструментально-технологическая среда формирования и использования опыта проектной организации / В.А. Маклаев, П.И. Соснин // *Автоматизация процессов управления*. 2009. № 2 (16). С.8-13.
- [9] **Соснин, П.И.** Архитектурный подход к прецедентно-ориентированному решению задач в разработке автоматизированных систем / П.И. Соснин, С.С. Шумилов, А.Е. Ивасев // *Автоматизация процессов управления*. 2019. № 1 (55). С.49-56.
- [10] **Загорюлько, Ю.А.** Использование паттернов онтологического проектирования для разработки онтологий предметных областей / Ю.А. Загорюлько, О.И. Боровикова, Г.Б. Загорюлько // *Сборник тр. конф. «Знания-Онтологии-Теории» (ЗОНТ-2017)*. Новосибирск, 2017. С.139-148.
- [11] **Муромцев, Д.И.** Интеграция wiki-технологии и онтологического моделирования в задаче управления знаниями предприятия / Д.И. Муромцев [и др.] // *Сборник тр. конф. «КИИ-2008»*. Дубна, 2008. С.360-368.
- [12] **Самойлов, Д.Е.** Паттерны структурной организации системы измеряемых свойств в онтологическом анализе данных / Д.Е. Самойлов, В.А. Семенова, С.В. Смирнов // *Сборник тр. конф. «Проблемы управления и моделирования в сложных системах»*. Самара, 2018. С.358-366.
- [13] **Shaaban, A.M.** Ontology-based security tool for critical cyber-physical systems / A.M. Shaaban, T. Gruber, C. Schmittner // *Proceedings of the 23rd International Systems and Software Product Line Conference*. Vol. B. 2019. P.207-210.
- [14] **Bhatia, M.P.S.** Ontologies for software engineering: past, present and future / M.P.S. Bhatia, A. Kumar, R. Beniwal // *Indian Journal of Science and Technology*. 2016. Vol. 9. P.1-16.

- [15] *Falbo, R.A. et al.* An ontology pattern language for service modeling / R.A. Falbo et al. // Proceedings of the 31st Annual ACM Symposium on Applied Computing. 2016. P.321-326.
- [16] *Ilyas, Q.M.* Ontology Augmented Software Engineering / Q.M. Ilyas // Software Development Techniques for Constructive Information Systems Design. IGI Global, 2013. P.406-413.
- [17] *Isotani, S. et al.* Ontology driven software engineering: a review of challenges and opportunities / S. Isotani et al. // IEEE Latin America Transactions. 2015. Vol.13. P.863-869.
- [18] *Pan, J.Z. et al.* Ontology-driven software development / J.Z. Pan et al. Springer Science & Business Media, 2012.
- [19] What is 1C:Enterprise? - https://1c-dn.com/1c_enterprise/what_is_1c_enterprise/.
- [20] PlantUML. UML Diagram Generator. - <https://plantuml.com>.
- [21] Protégé. Free, open-source ontology editor and framework for building intelligent systems. - <https://protege.stanford.edu>.

Сведения об авторах



Гуськов Глеб Юрьевич, 1992 г. рождения, кандидат технических наук, окончил факультет информационных систем и технологий Ульяновского государственного технического университета (УлГТУ). Доцент кафедры «Информационные системы» УлГТУ. Имеет около 30 работ в области онтологического инжиниринга и интеллектуального анализа данных. Author ID (RSCI): 812005; Author ID (Scopus): 57021635300. g.guskov@ulstu.ru.

Наместников Алексей Михайлович, 1974 г. рождения, доктор технических наук, доцент, окончил радиотехнический факультет УлГТУ. Профессор кафедры «Информационные системы» УлГТУ. Имеет более 80 работ в области автоматизированного проектирования и интеллектуальных систем. Author ID (RSCI): 392690; Author ID (Scopus): 9277806100. nam@ulstu.ru.



Романов Антон Алексеевич, 1986 г. рождения, кандидат технических наук, окончил факультет информационных систем и технологий УлГТУ, доцент кафедры «Информационные системы» УлГТУ. Имеет около 60 работ в области интеллектуальных систем хранения и обработки информации. ORCID: 0000-0001-5275-7628; Author ID (RSCI): 684949; Author ID (Scopus): 55903279200. romanov73@gmail.com.

Филиппов Алексей Александрович, 1987 г. рождения, кандидат технических наук, окончил факультет информационных систем и технологий УлГТУ, доцент кафедры «Информационные системы» УлГТУ. Имеет около 80 работ в области онтологического инжиниринга и интеллектуального анализа данных. ORCID: 0000-0001-5275-7628; Author ID (RSCI): 708454; Author ID (Scopus): 57191472723. al.filippov@ulstu.ru.



Поступила в редакцию 27.05.2021, после рецензирования 17.06.2021. Принята к публикации 23.06.2021.

Formation of a knowledge base to support the process of architectural design of software systems

G.Y. Guskov, A.M. Namestnikov, A.A. Romanov, A.A. Filippov

Ulyanovsk State Technical University, Ulyanovsk, Russia

Abstract

This article describes an approach to knowledge base (KB) formation for automating the process of architectural design of software systems (SS) based on the experience of previous projects. Software architecting is the presentation of software systems in the form of design artifacts and their architecture. When developing a new SS it is possible to improve its quality based on the experience of previous projects. The experience of previous projects is successful architectural

solutions contained in the knowledge base of the design organization. Such a KB should be formed in the process of analyzing design artifacts extracted from previous projects: source code, project diagrams, data models, structured text resources, etc. This article describes a KB model of a design organization and a model of the 1C: Enterprise 8 (1C) application solution as an example of a design artifact. The article also presents a method for generating fragments of a KB in the process of analyzing an applied solution for the 1C application and a method for generating use-case diagrams based on the KB content. A set of experiments was executed to evaluate the adequacy of the proposed models and methods. The results of experiments for assessing quality in terms of accuracy (the presence of elements of the expert diagram in the generated diagram) and completeness (the presence of elements in the generated diagram that are absent in the expert diagram) are presented. According to the results of the experiments, the average value of accuracy is 0.875, and the completeness is 0.6.

Key words: *architecting, software system, design artifact, knowledge base, design experience.*

Citation: *Guskov GY, Namestnikov AM, Romanov AA, Filippov AA. Formation of a knowledge base to support the architecting process of software systems [In Russian]. *Ontology of designing*. 2021; 11(2): 154-169. DOI: 10.18287/2223-9537-2021-11-2-154-169.*

Acknowledgment: The authors acknowledge that the work was supported by the framework of the state task of the Ministry of Science and Higher Education of the Russian Federation No.075-00233-20-05 "Research of intelligent predictive multimodal analysis of big data, and the extraction of knowledge from different sources". The reported study was funded by RFBR and the government of Ulyanovsk region according to the research projects: 19-47-730006 and 19-47-730005.

List of figures and tables

Figure 1 - Example of a generated ontology

Figure 2 - Example of generated use case diagram

Figure 3 - Use case diagram for the "Trade Management" 1C application generated by the system

Figure 4 - Use case diagram for the "Trade Management" 1C application built by the expert

Table 1 - Correspondence between the 1C solution objects and the O^{exp} ontology axioms

Table 2 - Correspondence between design artifact ontology components and use-case diagram components

Table 3 - Experimental results

References

- [1] ISO/IEC/IEEE 42010:2011 Systems and Software Engineering - Architecture Description. 2011. <https://www.iso.org/standard/50508.html>.
- [2] **Heesch U, Avgeriou P, Hilliard R.** A documentation framework for architecture decisions. *Journal Syst. Softw.* 2012; 85(4): 795-820. DOI:10.1016/j.jss.2011.10.017.
- [3] **Hilliard R.** Using Aspects in Architectural Description. *Lecture Notes in Computer Science*, 2007; 4765: 65-68.
- [4] **Sosnin P.** Substantially evolutionary theorizing in designing software-intensive systems. *Information* (Switzerland), 2018; 9 (4): 91.
- [5] **Sosnin P, Pushkareva A.** Ontological controlling the lexical items in conceptual solution of project tasks. // *Lecture Notes in Computer*, 2017; 10409: 31-46.
- [6] ISO/IEC 25010 2011. ISO/IEC 25010:2011: Systems And Software Engineering – Systems And Soft-Ware Quality Requirements And Evaluation (Square) – System And Software Quality Models.
- [7] **McConnell S.** Code complete. – Pearson Education, 2004.
- [8] **Maklaev VA, Sosnin PI.** Instrumental and technological environment for design organization experience formation and usage [In Russian] // *Automation of Control Processes*. 2009; 2(16): 8-13.
- [9] **Sosnin PI, Shumilov SS, Ivasev AE.** An architectural approach to the precedent-oriented problem solving in the designing of automated systems [In Russian]. *Automation of Control Processes*. 2019; 1(55): 49-56.
- [10] **Zagorulko YA, Borovikova OI, Zagorulko GB.** Using Ontological Design Patterns to Domain Ontologies Development [In Russian]. Proceedings of the Conference "Knowledge-Ontology-Theory" (3OHT-2017). Novosibirsk, 2017. P.139-148.
- [11] **Muromtsev DI et al.** Integration of wiki technology and ontological modeling in the enterprise knowledge management [In Russian]. Proceedings of the RCAI Conference (КИИ-2008). Dubna, 2008. P.360-368.

- [12] **Samoylov DE, Semenova VA, Smirnov SV.** Patterns of the structural organization of the system of measured properties in ontological data analysis [In Russian]. Proceedings of the Conference «Control and modeling problems in complex systems». Samara, 2018. P.358-366.
 - [13] **Shaaban AM, Gruber T, Schmittner C.** Ontology-based security tool for critical cyber-physical systems // Proceedings of the 23rd International Systems and Software Product Line Conference. 2019; B: 207-210.
 - [14] **Bhatia MPS., Kumar A, Beniwal R.** Ontologies for software engineering: past, present and future. Indian Journal of Science and Technology. 2016; 9: 1-16.
 - [15] **Falbo RA et al.** An ontology pattern language for service modeling. Proceedings of the 31st Annual ACM Symposium on Applied Computing. 2016. P.321-326.
 - [16] **Ilyas QM.** Ontology Augmented Software Engineering. Software Development Techniques for Constructive Information Systems Design. IGI Global, 2013. P.406-413.
 - [17] **Isotani S. et al.** Ontology driven software engineering: a review of challenges and opportunities. IEEE Latin America Transactions. 2015; 13: 863-869.
 - [18] **Pan JZ. et al.** Ontology-driven software development. Springer Science & Business Media, 2012.
 - [19] What is 1C:Enterprise? https://1c-dn.com/1c_enterprise/what_is_1c_enterprise/.
 - [20] PlantUML. UML Diagram Generator. <https://plantuml.com>.
 - [21] Protégé. Free, open-source ontology editor and framework for building intelligent systems. <https://protege.stanford.edu>.
-

About the authors

Gleb Yurievich Guskov (b. 1992), graduated from Ulyanovsk State Technical University in 2014, PhD (2018). He is an Associate Professor at Ulyanovsk State Technical University (Department of information systems). He is a co-author of about 30 scientific articles and abstracts in the field of ontology engineering and data mining. Author ID (RSCI): 812005; Author ID (Scopus): 57021635300. g.guskov@ulstu.ru.

Aleksey Mihaylovich Namestnikov (b. 1974), graduated from Ulyanovsk State Technical University in 1996, D. Sc. Eng. (2018). He is a Professor at Ulyanovsk State Technical University (Department of information systems). He is a co-author of about 80 scientific articles and abstracts in the field of CAD and AI. Author ID (RSCI): 392690; Author ID (Scopus): 9277806100. nam@ulstu.ru.

Anton Alekseevich Romanov (b. 1986), graduated from Ulyanovsk State Technical University in 2009, PhD (2013). He is an Associate Professor at Ulyanovsk State Technical University (Department of information systems). He is a co-author of about 60 scientific articles and abstracts in the field of intelligent systems. ORCID: 0000-0001-5275-7628; Author ID (RSCI): 684949; Author ID (Scopus): 55903279200. romanov73@gmail.com.

Aleksey Aleksandrovich Filippov (b. 1987), graduated from Ulyanovsk State Technical University in 2009, PhD (2013). He is an Associate Professor at Ulyanovsk State Technical University (Department of information systems). He is a co-author of about 80 scientific articles and abstracts in the field of ontology engineering and data mining. ORCID: 0000-0001-5275-7628; Author ID (RSCI): 708454; Author ID (Scopus): 57191472723. al.filippov@ulstu.ru.

Received 27 May, 2021. Revised June 17, 2021. Accepted 23 June, 2021.
