

УДК 681.3

## ПОСТРОЕНИЕ ОНТОЛОГИИ АРХИТЕКТУРЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Грегер С.Э., Поршневу С.В.

*ГОУ «Уральский федеральный университет*

*имени первого Президента России Б.Н. Ельцина», Екатеринбург, e-mail: sergey\_porshnev@mail.ru*

Проведен анализ процесса построения архитектуры информационной системы. Выявлено, что описание общей архитектуры информационной системы представляет собой композицию описаний моделей, построенных в разных нотациях, с использованием отличающихся друг от друга семантик. Средства и методы построения этих моделей могут быть выделены в независимые подсистемы в составе системы, обеспечивающей разработку целевой информационной системы, и общая архитектура целевой системы зависит от характеристик обеспечивающей системы. Выявлен состав подсистем, образующий обеспечивающую систему периода разработки веб-информационной системы. Разработана онтология для формализации описания архитектуры информационной системы, предоставляющая возможность определять конфигурацию проекта разработки, системную модель и взаимосвязи онтологий, используемых для представления подсистем. Предложен метод согласования описаний моделей, позволяющий формировать описание общей архитектуры целевой системы.

**Ключевые слова:** информационная система, архитектура, онтология, база знаний

## BUILDING AN ONTOLOGY INFORMATION SYSTEM ARCHITECTURE

Greger S.E., Porshnev S.V.

*Ural Federal University named after the first President of Russia*

*B.N. Yeltsin, Ekaterinburg, e-mail: sergey\_porshnev@mail.ru*

Analysis of the process of building information system architecture. It was revealed that the description of the general architecture of the information system is a composition of descriptions of models built in different notations, using differing semantics. Means and methods for constructing these models can be separated into independent subsystems in the system providing target information system design, and the general architecture of the target system depends on the characteristics of providing the system. Revealed the composition of subsystems forming system provides design-time Web information system. Ontology developed to formalize the description of information system architecture that provides the opportunity to determine the configuration of the project development, system model, and the relationship of ontologies used to represent the sub-systems. We propose a method of matching the descriptions of models allows you to shape the description of the overall architecture of the target system.

**Keywords:** information system architecture, ontology, knowledge base

Информационные системы (ИС) относятся к классу сложных систем. При разработке ИС вопросам интеграции этих систем уделяется пристальное внимание. Характерной особенностью современных веб-ориентированных информационных систем (ИС) является переход к архитектуре разнородных, в том числе распределённых, объектов, сервисов и инфраструктур. Например, в качестве объектов могут быть рассмотрены автономные внешние системы. В этой ситуации одной из наиболее важных проблем становится проблема создания общей архитектуры информационной системы и сохранение ее целостности на всех этапах ее жизненного цикла. Под архитектурой мы понимаем фундаментальную организацию системы, воплощённую в её компонентах, их отношениях друг с другом и с внешним окружением, а также принципы, управляющие её созданием и развитием.

Архитектура призвана определять согласование часто противоречивых требований к ИС со способами их реализации, определяемыми характеристиками взаимо-

действующих подсистем. Сложность создания архитектуры ИС определяется, как минимум, двумя причинами. Во-первых, нет общепринятой понятийной базы для описания подсистем. Как правило, понятийная база описания архитектуры каждой подсистемы отражает ее предметную область. Во-вторых, на архитектуру ИС оказывает существенное влияние внешняя по отношению к ней система, обеспечивающая процесс разработки ИС (обеспечивающая система периода разработки). В этой связи существует необходимость, обеспечения взаимодействия как на уровне целевой, так и на уровне обеспечивающей системы. На уровне обеспечивающей системы взаимодействие проявляется в необходимости согласования функциональных, технологических и структурных описаний и их согласований.

Анализ существующих подходов показывает [1,5,4], что для решения указанных проблем используются декларативные способы описания данных, средств их обработки и конфигураций, являющихся объектами исследования семантического

моделирования (СМ), в создании универсальной семантической модели представления и обработки знаний. В настоящее время активно обсуждаются как проблемы использования методов семантического подхода для согласования описаний взаимодействующих систем, так способы и технической реализации таких семантических моделей [2, 6, 8, 9]. Однако целый ряд вопросов, связанных с практической реализацией указанных подходов, по-прежнему остается актуальным. В их числе: отсутствие согласованных методов формирования онтологических моделей, их интеграция и трансформация, в частности, отсутствие методов создания и согласования онтологических моделей архитектурных описаний.

В статье рассматриваются подходы к созданию единой онтологии архитектуры ИС, основанные на методах и технологиях управления знаниями, а также способ ее использования для согласования функциональных и структурных описаний целевой и обеспечивающей систем. Показано, что база знаний, сформированная на основе указанной онтологии, позволяет аккумулировать опыт построения архитектур и системных отношений веб-информационных систем и повторно использовать его для разработки новых информационных систем.

#### Обоснование подхода к построению единой онтологии ИС

Анализ известных подходов к использованию ресурсов обеспечивающей системы при разработке целевой ИС позволяет сделать вывод о том, что описание предметной области целевой ИС, включающее в себя концептуальную модель и модель задач, является входной информацией, а сама архитектура ИС – результатом выполнения функции

$$S_{App} = S_{Об}(S_{Domain}),$$

где  $S_{App}$  – разрабатываемая ИС,  $S_{Domain}$  – система описания предметной области,  $S_{Об}$  – обеспечивающая система этапа разработки и эксплуатации ИС.

Рассмотрим обеспечивающую систему как сложную систему и выделим входящие в нее подсистемы, предполагая, что каждый вид деятельности обеспечивается отдельной, независимой от остальных, обеспечивающей системой. При разработке веб-информационной системы как отдельные виды деятельности обычно выделяют задачи выявления и согласования задач, решаемых в рамках целевой ИС, разработку ее структур данных, средств для программной обработки данных, пользовательского интерфейса и модели переходов между

отдельными элементами веб-приложения – веб-страницами и сервисами портала. Тогда

$$S_{App} = \langle S_{Domain}, S_{Task}, S_{Info}, S_{Comp}, S_{Gui}, S_{Nav} \rangle,$$

где  $S_{App}$  – система, обеспечивающая описание программных элементов разрабатываемой информационной системы;  $S_{Domain}$  – система, предоставляющая возможности описания предметной области;  $S_{Task}$  – система, обеспечивающая описание процессов и задач для различных предметных областей, и управление ими;  $S_{Info}$  – система управления информационной моделью, позволяющая формировать описания информационных объектов и структур данных;  $S_{Comp}$  – система разработки программных компонент;  $S_{Gui}$  – система разработки пользовательского интерфейса;  $S_{Nav}$  – система разработки навигационной модели приложения.

Общепринятый подход к описанию архитектуры зафиксирован в стандарте ISO 42010 (Рекомендованная практика архитектурного описания программных систем). В этом стандарте архитектура системы представляется через архитектурные описания, учитывающие интересы заинтересованных сторон и включающие в себя наборы моделей, описывающих связь функции системы и ее конструкции. Архитектурное описание создается для каждой заинтересованной стороны (stakeholders), имеющей некоторый набор интересов (concerns), связанных с системой, и стремящейся их удовлетворить. Для каждого из интересов создаются отдельные группы описаний (ГО) системы (views). Каждая группа описаний раскрывает отдельный аспект системы, а набор групп образует ее полное описание. Соглашения, по которым ГО создается, отображается и анализируется, устанавливаются методом описания (viewpoint). Таким образом, архитектура  $Arch = \{ADesc_i\}$ , где  $ADesc_i = \langle M_i, A_i, R_i, Z_i \rangle$ ,  $i = 1, \dots, n$ , где  $i$  – аспект, характеризующий некоторую группу описаний  $ADesc_i$ , отвечающую способу описания  $Z_i$ , который связывает множество элементов  $M_i$ , определенных на множестве атрибутов  $A_i$ , через отношения  $R_i$ . Используя семантический метод описания, каждой частной обеспечивающей системе сопоставим свою группу описаний и некоторую онтологию предметной области:

$O_{App}$  – онтология целевой системы, являющаяся результатом разработки и описывающая техническую реализацию ИС;

$O_{Domain}$  – онтология предметной области;

$O_{Task}$  – онтология задач;

$O_{Nav}$  – онтология навигации;

*OInfo* – онтология информационных элементов;

*OComp* – онтология компонентов;

*OGui* – онтология пользовательского интерфейса.

Каждая из групп описаний характеризуется уникальным набором моделей, задач, инструментов и исполнителей, и ее структурой, и функционированием. Для каждой группы описаний необходимо определить методы рассмотрения и описания систем и правила их применения. С целью повторного использования будем предполагать, что для каждой ГО и метода описания определены отдельные онтологические модули, составляющие онтологию ГО. Для совместного использования онтологий определим онтологическую систему [7]:

$$\Sigma^0 = \langle O^{meta}, \{O^{d\&t}\}, \Sigma^{inf} \rangle,$$

где  $O^{meta}$  – онтология верхнего уровня, в данном случае онтология архитектуры;  $\{O^{d\&t}\}$  – множество предметных онтологий и онтологий задач предметной области;  $\Sigma^{inf}$  – модель машины вывода, ассоциированной с онтологической системой  $\Sigma^0$ .

Онтология  $O^{meta}$  должна предоставлять классы, позволяющие организовать набор определений понятий предметной области и их взаимосвязей и позволять интегрировать знания подсистем и использовать эти знания при решении своих задач.

Для решения поставленных нами задач разработана онтология архитектуры информационной системы с набором необходимых онтологических классов. Определим концепты онтологии архитектуры и связи между ними. Курсивом будем обозначать концепты, а «жирным» курсивом – отношения между ними. Примеры высказываний, представленных в онтологии архитектуры:

*модуль онтологии* = {онтология концептов, онтология операций, онтология нотаций},

*система* = {целевая система, обеспечивающая система},

*целевая система* **имеет** *обеспечивающая система*,

*система* **включает подсистему *система*,**

*архитектура* **имеет** *группа описаний*,

*группа описаний* = {*domainview*, *taskview*, *infoview*, *navview*, *guiview*, *compview*}.

Здесь

*domainview* – ГО концептуальной модели предметной области;

*navview* – ГО навигационной структуры приложения;

*infoview* – ГО информационных элементов и их схем данных;

*guiview* – ГО графического интерфейса пользователя;

*taskview* – ГО процессов и задач;

*compview* – ГО программных компонентов ИС.

Отображение дерева наследования онтологических классов и часть связей между ними в разработанном нами редакторе онтологий представлено на рисунке (рис. 1).

Онтология состоит из двух групп классов – классов предметной области и классов классификаторов. Для пар классов из этих двух групп устанавливается связь:

*классификатор классифицирует класс*.

Набор объектов классификатора задает разбиение множества объектов класса предметной области на группы. Набор классификаторов вместе с их взаимными связями задают класс шаблона отношений, поскольку структура отношений их объектов задает структуру связей между объектами классов предметной области. Между классификаторами устанавливается система наследования, а между шаблонами – система отношений специализаций шаблонов. Пример отношений:

*вид модели соответствует – вид примитива моделирования*;

*вид модель интерфейса пользователя подкласс – вид примитив моделирования интерфейса пользователя*;

*вид модель интерфейса пользователя соответствует – вид примитив моделирования интерфейса пользователя*

*модель состоит из примитив моделирования*;

*модель интерфейса пользователя – подкласс модель*;

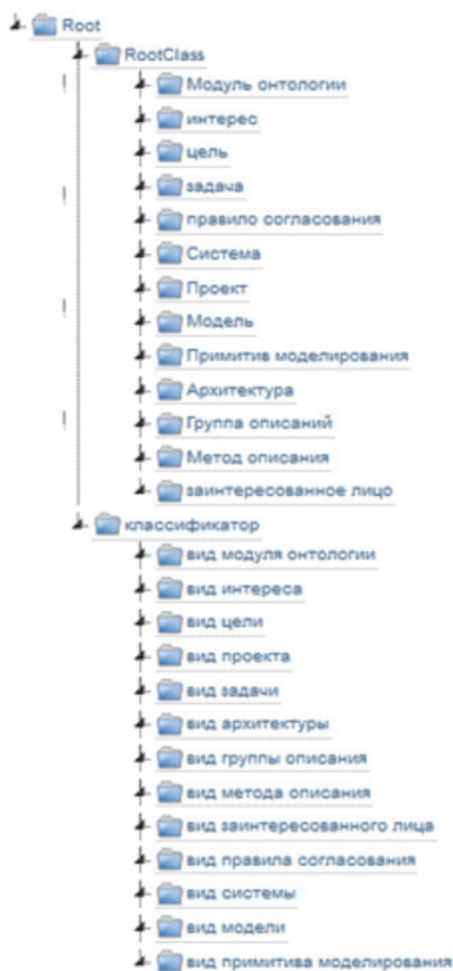
*модель интерфейса пользователя состоит из примитив моделирования интерфейса пользователя*

Связи между целевой и обеспечивающей системами, а также состав онтологических модулей определяются шаблонами, задаваемыми классом проект и его классификаторами. Множество онтологических модулей в процессе создания и использования базы знаний могут быть выделены в библиотеку модулей (введением необходимых классов и связей) и повторно использоваться в различных проектах. Для обеспечения интеграции между онтологиями групп описаний используем механизм аннотации. Рассмотрим:

онтологию  $O$  верхнего уровня описывающую набор конструкций, используемых в достаточно широком классе информационных моделей;

онтологии групп описаний  $V$  и  $U$ ;

множества аннотаций  $A_V$  и  $A_U$  соответственно в терминах онтологии  $O$ , связанных с классами онтологий групп описаний.



#### Ontology relations

модель предметной области	имеет концептуальную модель	Концептуальная модель
модель предметной области	имеет модель задач	Модель задач
Система	включает подсистему	Система
Система	имеет вид системы	вид системы
Система	имеет архитектуру	Архитектура
Обеспечивающая система	имеет вид системы	Вид обеспечивающей системы
информационная система	имеет обеспечивающую систему	Обеспечивающая система
веб-информационная система	вид веб системы	вид веб-системы
Проект	включает проект	Проект
Проект	имеет целевую систему	веб-информационная система
Проект	обеспечивающая система	Обеспечивающая система
Модель	использует примитив	Примитив моделирования
Модель	представлена в	Модуль онтологии
Примитив моделирования	имеет вид	вид примитива моделирования
Архитектура	имеет метод описания	Метод описания
Архитектура	включает группу описаний	Группа описаний
Группа описаний	включает описание	Модель
Метод описания	представлен в	Модуль онтологии
заинтересованное лицо	имеет цель	интерес

Рис. 1. Схема онтологии архитектуры

Назовем *аннотацией* пару  $\langle u, c \rangle$ , где  $c$  – понятие из  $O$ , либо композиция понятий из  $O$ ,  $u$  – элемент эталонной схемы, являющийся экземпляром понятия  $c$ . Таким об-

разом, классы и объекты онтологий групп описаний являются экземплярами понятий, которые в свою очередь есть подтипы понятий онтологии  $O$  (рис. 2).



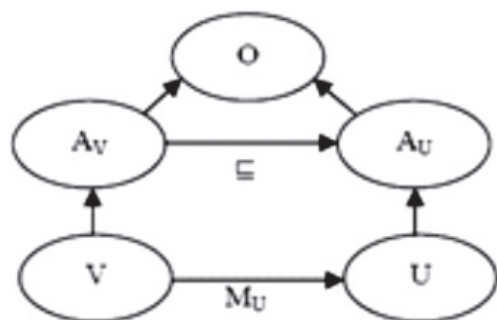


Рис. 2. Схема механизма аннотаций

Для проведения операции аннотации нужно создать специальное отношение между подграфом из  $V$  или  $U$  и понятием  $O$ . В нашем случае для этого введен специальный класс связи имеет метатип. Обеспечение интероперабельности моделей вводится отображение  $M_U(V)$  модели  $V$  в модель  $U$ . Однако модели изначально никак не связаны, а следовательно, необходимо найти семантически близкие конструкции моделей. Будем считать семантически близкими пары конструкций  $\langle u, v \rangle$ ,  $u \in U$ ,  $v \in V$ , для которых существуют аннотации  $\langle v, c \rangle \in A_V$  и  $\langle u, d \rangle \in A_U$ , такие, что  $c$  является подпонятием  $d$ . Определим  $task_m, info_p, gui, navlink, f_k$  как элементы из ГО:  $taskview, infoview, guiview, navview, compview$  соответственно. В онтологии архитектуры для каждой из ГО определим классы (в более общем случае дерево наследования классов) примитивов моделирования для каждой из онтологий ГО. Определение связей между классами примитивов и применение механизма аннотации позволит представить модели из разных групп описаний и, в конечном итоге, определить архитектуру системы.

**Заключение**

В результате проведенного исследования была разработана онтология, позволяющая формализовать описание архитектуры информационной системы и интегрировать модели, созданные в рамках разных онтологических моделей. Представленная онтология была использована при разработке интеллектуальной системы проектирования веб-приложений [3]. В результате применения онтологии выявлена потребность введения специальных модулей онтологий, носящих общий характер. Одной из таких онтологий является онтология жизненного цикла информационной системы. В настоящее время ведутся работы по ее созданию.

Работа выполнена в рамках договора № 02.G25.31.0055 (проект 2012-218-03-167).

**Список литературы**

1. Боровикова О.И., Булгаков С.В., Загорюлько Ю.А., Сидорова Е.А., Циркин Б.Г., Холлошкин Ю.П., Андреева О.А. Археологический портал знаний: содержательный доступ к знаниям и информационным ресурсам по археологии // Труды 10-й

национальной конференции по искусственному интеллекту с международным участием. – М., 2006. – С. 832–840.

2. Гулякина Н.А., Голенков В. В. Графодинамические модели параллельной обработки знаний: Принципы построения, реализации и проектирования // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III Междунар. научн.-техн. конф., 21–23 февраля. – Минск, 2013. – С. 23–52.

3. Грегер С.Э. Объектные системы – 2013 // Интеллектуальная система проектирования веб-приложений: материалы VII Международной научно-практической конференции. – Ростов-на-Дону, 2013. – С. 50–54.

4. Заливако С.С., Шункевич Д.В. Семантическая технология компонентного проектирования интеллектуальных решателей задач // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III Междунар. научн.-техн. конф. (21–23 февраля 2013 г.). – Минск, 2013. – С. 297–313.

5. Пивоварчик О.В., Лазуркин Д.А. Гулякина Н.А. Языки и технологии программирования, ориентированные на обработку семантических сетей. – Минск, 2013. – С. 221–228.

6. Тимченко В.А. Модель классов семантических сетей и их преобразований // Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2013): материалы III Междунар. научн.-техн. конф., 21–23 февраля. – Минск, 2013. – С. 63–69.

7. Хорошевский В.Ф. Гаврилова Т.А. Базы знаний интеллектуальных систем. – СПб.: Питер, 2000. – 384 с.

8. Guarino N. Formal Ontology and Information Systems // N. Guarino (ed.) Formal Ontology and Information Systems. – Amsterdam, 1998. – С. 3–15.

9. Hruby P. Ontology-Based Domain-Driven Design // [Soft-MetaWare] URL: www.softmetaware.com/oopsla2005/hruby.pdf.

**References**

1. Borovikova O.I., Bulgakov S.V., Zagorulko Ju.A., Sidorova E.A., Tsirkin B.G., Kholjushkin Ju.P. Andreeva O.A., Arkheologicheskij portal znaniy: soderzhatelnyj dostup k znaniyam i informatsionnym resursam po arkheologii. Trudy 10 natsionalnoj konferentsii po iskusstvennomu intellektu s mezhdunarodnym uchastiem. Moscow, 2006, pp. 832–840.

2. Gulyakina N.A. Golenkov V.V. Grafodinamicheskie modeli parallelnoj obrabotki znaniy: Printsipy postroeniya, realizatsii i proektirovaniya. Otkrytye semanticheskie tekhnologii proektirovaniya intellektualnyh sistem (OSTIS-2013): materialy III Mezhdunar. nauchn. tekhn. konf., 21–23 fevralya. Minsk, 2013, pp. 23–52.

3. Greger S.E. Obektnye sistemy – 2013: materialy VII Mezhdunarodnoj nauchno- prakticheskoy konferentsii. Intellektualnaya sistema proektirovaniya veb-prilozhenij. Rostov-na-Donu, 2013, pp. 50–54.

4. Zalivako S.S., Shunkevich D.V. Semanticheskaya tekhnologiya komponentnogo proektirovaniya intellektualnyh reshatelnej zadach. Otkrytye semanticheskie tekhnologii proektirovaniya intellektualnyh sistem (OSTIS-2013): materialy III Mezhdunar. nauchn. tekhn. konf. (21–23 fevralya 2013 g.). Minsk, 2013, pp. 297–313.

5. Pivovarchik O.V., Lazurkin D.A. Gulyakina N.A., Yazyki i tekhnologii programmirovaniya, orientirovannye na obrabotku semanticheskikh setej. Minsk, 2013, pp. 221–228.

6. Timchenko V.A. Model klassov semanticheskikh setej i ikh preobrazovaniy. Otkrytye semanticheskie tekhnologii proektirovaniya intellektualnyh sistem (OSTIS-2013): materialy III Mezhdunar. nauchn. tekhn. konf., 21–23 fevralya. Minsk, 2013, pp. 63–69.

7. Khoroshevskij V.F. Gavrilova T.A. Bazy znaniy intellektualnyh sistem. SPb., Piter, 2000. 384 p.

8. Guarino N. Formal Ontology and Information Systems. Formal Ontology and Information Systems. Amsterdam, 1998, pp. 3–15.

9. Hruby P. Ontology-Based Domain-Driven Design. [Soft-MetaWare] URL: www.softmetaware.com/oopsla2005/hruby.pdf.

**Рецензенты:**

Гадзиковский В.И., д.т.н., доцент, профессор кафедры радиоэлектроник информационных систем Уральского федерального университета им. первого Президента России Б.Н. Ельцина, г. Екатеринбург;

Доросинский Л.Г., д.т.н., профессор, заведующий кафедрой информационных технологий Уральского федерального университета им. первого Президента России Б.Н. Ельцина, г. Екатеринбург.

Работа поступила в редакцию 08.11.2013.