

УДК 681.5:519.819:331.45

Автоматизация процессов управления образовательным учреждением на основе онтологического моделирования и сервис-ориентированной архитектуры

А. А. Петрикин, И. В. Богуславский

(Донской государственный технический университет)

Затронута тема текущего состояния автоматизации отечественных высших учебных заведений. Определены аспекты автоматизации, которые являются ключевыми при построении информационно-управляющей системы (ИУС). Сформулированы задачи, которые решает внедрение ИУС в вузе. Сделан вывод о том, сервис-ориентированная архитектура (COA) лучше всего подходит для проектирования ИУС вуза. Приведено формальное определение сервис-ориентированной архитектуры, раскрыты положительные стороны внедрения сервис-ориентированной технологии в информационное пространство вуза. Схема многоуровневой ИУС вуза для наглядности представлена в графическом виде. Рассматривается также вопрос избыточности сервисов, характерной для большинства информационных систем при продолжительном сроке эксплуатации. В качестве решения данной проблемы предложено использование структурного метасервиса, который осуществляет координацию взаимодействия структурных компонентов ИУС. Предложена математическая модель ИУС. Обобщая политику управления ИУС на базе COA, авторы делают вывод о необходимости унифицированного описания предметной области. Рассмотрены преимущества онтологического подхода для концептуализации предметной области. Сделан вывод, что для эффективного функционирования и сопровождения ИУС необходимо разработать онтологическое описание понятий и отношений между понятиями, семантическое ядро и механизм трансформации онтологического описания. В графическом виде приведена схема концепции ИУС вуза, а также структура онтологической модели предметной области вуза как совокупность онтологических моделей отдельных сервисов ИУС.

Ключевые слова: автоматизация, информационно-управляющая система, сервис, сервис-ориентированная архитектура, структура, структурный компонент, онтология, онтологическое моделирование, предметная область, семантическое ядро, математическая модель, унификация, высшее учебное заведение.

Введение. В современных условиях одним из направлений повышения эффективности управления образовательным учреждением является внедрение информационно-управляющей системы (ИУС).

Внедрение в высшем учебном заведении информационно-управляющей системы решает следующие задачи: упорядоченное управление информационными потоками, создание единого информационного пространства, обеспечение распределённого доступа к данным и поддержка принятия решений.

В настоящий момент российские вузы находятся на стадии реформирования, связанного с присоединением России к Болонскому процессу, внедрением ЕГЭ, кредитных и рейтинговых систем. Закономерно, что большинство процессов вуза постоянно изменяются. Перемены в образовательных процессах требуют корректировки управления ими. При автоматизации этих процессов необходимо предоставить студентам доступ к информационным ресурсам вуза в полном объёме. Следует помнить также о том, что не каждый вуз может позволить себе единовременную автоматизацию своей деятельности — в этом случае временные издержки на обучение и адаптацию пользователей могут привести к срыву учебного процесса. Кроме того, потребуются весьма значительные материальные затраты. Таким образом, необходимо ещё на стадии проектирования закладывать в автоматизированную систему возможность гибкого внедрения.

Аспекты автоматизации вуза. Эффективность функционирования учебного заведения зависит от качества управления его процессами и своевременного принятия решений. ИУС — это своего рода инструмент реализации процессов учебного заведения и поддержки принятия решений, по-

этому от возможностей ИУС напрямую зависит эффективность функционирования вуза в целом [1]. На первый план выходят вопросы разработки системы с гибкой архитектурой, которая могла бы легко вписываться в текущую инфраструктуру любого вуза, а также подстраиваться под изменяющиеся процессы.

Вуз представляет собой динамично изменяющуюся среду, поэтому программы, автоматизирующие некоторый процесс, могут устареть ещё до момента внедрения ИУС. Чем шире распространяется автоматизация, тем сложнее процесс модификации и сопровождения информационных систем. Начиная с определённого уровня сложности, разработчики занимаются поддержкой уже разработанных систем, что, в свою очередь, тормозит развитие ИУС в целом [2].

К сложности объекта автоматизации добавляется сложность самой ИУС. ИУС вуза включает в себя множество функциональных блоков, сервисов, коммуникационных устройств, систем, технологий, данных, пользователей и т. д. Сложность эксплуатации ИУС обусловлена длительным жизненным циклом, в течение которого происходят непрерывная смена аппаратного обеспечения (замена серверов, компьютеров, модернизация линий связи), появление нового программного обеспечения, модернизация элементов ИУС и т. п. Это приводит к необходимости разработки архитектуры жизнеспособной ИУС с возможностью быстрого расширения и модернизации используемых элементов.

Архитектура ИУС должна обеспечивать масштабирование по целому ряду направлений: данные, функциональность, пользователи, — т. е. быть адаптивной по отношению к внешней среде. Под адаптивностью следует понимать способность ИУС самонастраиваться с учётом изменений во внешнем окружении. Для решения этих задач требуется сервис-ориентированная архитектура (COA), которая обеспечит автоматическое поддержание эффективной работы и качества данных ИУС.

Сервис-ориентированная архитектура ИУС. Приведём формальное определение сервис-ориентированной архитектуры, данное специалистами корпорации *IBM* [3]: «COA — это прикладная архитектура, в которой все функции определены как независимые сервисы с вызываемыми интерфейсами. Обращение к этим сервисам в определённой последовательности позволяет реализовать тот или иной бизнес-процесс».

С точки зрения разработчиков, ту же мысль можно передать несколько иными словами: «COA — это компонентная модель, в которой разные функциональные единицы, называемые сервисами, взаимодействуют по сети с пользователями посредством интерфейсов. При этом интерфейсы должны быть нейтральными к специфике реализации сервисов, которые определяются аппаратной платформой, операционной системой, языком программирования. Подобная нейтральность обеспечивает универсальность взаимодействия сервисов в разнородной (гетерогенной) среде, а сервисы, интегрированные посредством таких интерфейсов, являются слабо связанными».

Идея COA состоит в том, чтобы рассматривать ИУС как совокупность сервисов, которые выполняют единицу работы, и клиентов, которые обращаются к этим сервисам по простому документированному протоколу. В качестве сервисов можно выделять как повторяющиеся задачи внутри процессов [4], так и весь процесс.

Архитектура ИСУ предполагает использование структурных компонентов (СК), которые выполняют большую часть логики программы. Предназначение СК сводится к взаимодействию с базами данных и к обработке данных. Структурный компонент — довольно широкое понятие. Так можно назвать хранимые процедуры, функции баз данных, а также сами сервисы, если принимать к рассмотрению уровень подсистем.

Использование СК обеспечивает минимизацию изменений кода, так как, во-первых, способствует увеличению производительности процесса разработки приложений-сервисов, а, во-

вторых, позволяет разработчикам использовать унифицированные куски программного кода. Это даёт возможность уменьшить затраты на сопровождение ИУС. В результате использования компонентов сервисы становятся независимы от источника данных. Любые изменения источника данных никоим образом не влекут функциональных изменений сервиса, т. к. всю нагрузку по изменениям берут на себя СК, взаимодействующие с источником данных.

В общем случае архитектура ИУС является многоуровневой (рисунок 1). Уровни интерфейса и процессов взаимодействуют только с нижележащими уровнями. А на уровне функциональности, равно как и на уровне данных, взаимодействие может осуществляться между элементами уровня.

Развитие ИУС на основе сервис-ориентированного подхода рано или поздно может привести к тому, что сервисов (и, как следствие, структурных компонентов, которые они используют) становится достаточно много и без управления эффективность их использования снижается. Для решения этой проблемы необходимо использовать специализированный управляющий сервис (мета-сервис), основной задачей которого будет координация взаимодействия между структурными компонентами сервисов.

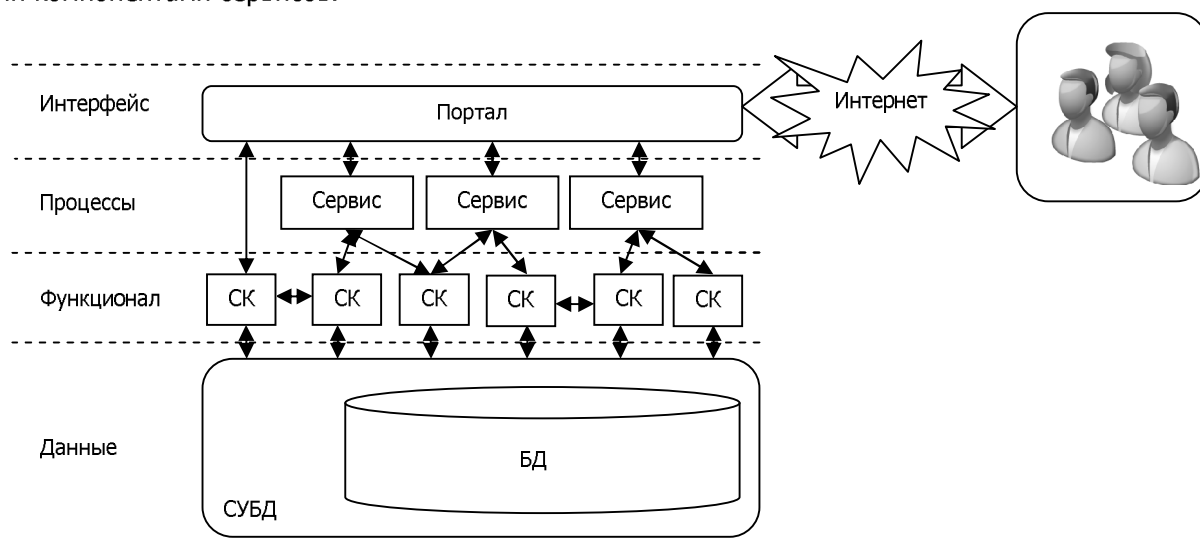


Рис. 1. Уровневая модель ИСУ

Проанализировав структурную модель ИУС с точки зрения сервис-ориентированной архитектуры, можно заключить следующее:

1) ИУС с сервис-ориентированной архитектурой S^{soa} включает множество структурных компонентов $S^c = \{S_n^c\}$ и множество сервисов $S = \{S_k\}$.

2) Структурные компоненты и сервисы находятся в таких отношениях, что каждый сервис для своего успешного функционирования требует определённого множества структурных компонентов.

3) ИУС с сервис-ориентированной архитектурой S^{soa} является динамической, т. е. для каждого определённого момента времени t вызывает некоторое множество сервисов $S_t = \{S_q\}$ и соответствующее им множество структурных компонентов $S_t^c = \{S_{qy}^c\}$, между которыми устанавливаются отношения зависимости.

4) Любая конфигурация ИУС с COA включает сервис управления.

5) Будем считать, что качество функционирования ИУС с COA (S^{soa}) в целом определяется качеством функционирования каждого сервиса в данный момент времени.

6) Каждый сервис $S = \{S_k\}$ и структурный компонент $S^C = \{S_n^C\}$ имеют множество требований $C = \{C_w\}$, критический уровень которых определяется соглашениями об уровне предоставления сервисов, стандартами, техническими регламентами, что и определяет качество функционирования S^{oa} в целом.

7) Существует множество регуляторов $A = \{A_u^w\}$ сервисов, когда каждому отклонению от требований поставлено в соответствие множество регуляторов $\sigma_u^w \rightarrow \{A_{ud}^w\}$.

Математическая модель мониторинга показателей структурных компонентов в сервис-ориентированной ИУС показывает, что функции F системы S^{oa} могут быть реализованы в момент времени t лишь при условии, что каждый структурный компонент обладает определёнными свойствами и параметрами, удовлетворяющими требованиям $C = \{C_w\}$.

$$F : ST^C \times H \times V \times C \times K \times M \rightarrow A,$$

где $ST^C = \{ST_{n1}^C\}$, $n1 = \overline{1, n1_0}$ — множество состояний элементов множества S^C ; $S^C = \{S_n^C\}$, $n = \overline{1, n_0}$ — множество структурных компонентов; $H = \{H_g\}$, $g = \overline{1, g_0}$ — множество характеристик, используемых для описания структурных компонентов; $V = \{V_s\}$, $s = \overline{1, s_0}$ — множество значений показателей структурных компонентов; $C = \{C_e\}$, $e = \overline{1, e_0}$ — множество требований к структурным компонентам; $K = \{K_j\}$, $j = \overline{1, j_0}$ — множество критериев, используемых для оценки соответствия предъявляемым требованиям состояния и свойств структурных компонентов; $M = \{M_r\}$, $r = \overline{1, r_0}$ — множество возможных регуляторов (действий) для приведения в соответствие предъявляемым требованиям показателей сервисов; $A = \{A_k\}$, $k = \overline{1, k_0}$ — множество регуляторов, необходимых для устранения отклонений текущих значений показателей структурного компонента от требуемых.

Требуется в определённые моменты времени t выявлять для каждого показателя $V = \{V_s\}$ сервисов отклонение от соответствующего критического фактора, входящего в множество $C = \{C_w\}$, и запускать в действие множество регуляторов $A_w \rightarrow \{A_{v1}^w\}$ для приведения показателя сервиса в заданные границы. Эти задачи реализуют в ИУС сервисы мониторинга показателей структурных компонентов, входящие в состав сервисов управления СОА и управления информационной безопасностью.

Согласно принципу повторного использования, каждый структурный компонент может быть сервисом корпоративного уровня или предоставляться внешним потребителям, может быть использован как атомарный сервис и как элемент композитного сервиса. Поэтому любые неконтролируемые изменения в структурных компонентах могут привести к сбоям, к прерыванию или остановке процессов. Таким образом, в сервис-ориентированной ИУС мониторинг показателей структурных компонентов играет важную роль в управлении СОА.

Грамотное и полноценное управление невозможно без целостного понимания тех компонентов, или столпов, которые поддерживают зрелый СОА-проект. Конечно, СОА-проект можно строить только на основных механизмах (механизме) поддержки. Однако зрелый проект подразумевает большой уровень поддержки и рост уровня ответственности. Каждая предметная область требует разного подхода к управлению СОА, что соответствующим образом отражается на «политике». Поэтому необходимо по возможности производить унифицированное описание предметной

области в виде взаимодействующих объектов, имеющих более понятные для специалистов-предметников средства. Примером может служить подход, использующий метаописание предметной области — онтологическое моделирование предметной области, ставшее особенно популярным в связи с идеей семантического *Web* [5], для которого разработан язык *OWL (Ontology Web Language)* [6].

Онтологический подход к описанию предметной области ИУС. Онтология является спецификацией концептуализации предметной области [7]. Онтологии содержат словарь понятий предметной области, описание их смысла, иерархию понятий, связанных друг с другом некоторым видом отношений [8]. Согласно онтологическому подходу, предметная область представляет собой совокупность взаимосвязанных понятий, имеющих некоторые атрибуты и свойства. Одной из сильных сторон онтологии является возможность создавать определённые пользователем отношения между классами [9].

Онтологии определяют термины для представления или описания некоторой предметной области. Онтологии включают определённые пользователем формальные базовые концепции предметной области и их взаимоотношения. Компоненты онтологии — это концепты, или *классы* (описывают понятия предметной области), *свойства* класса (описывают различные характеристики концепта) и *ограничения на свойства*.

Онтологии вместе с множеством экземпляров классов представляют собой базу знаний. Класс может иметь подклассы — это более специализированные концепты, чем надклассы. Свойства классов представляют собой либо атрибуты, либо отношения между классами. Собственно отношения являются свойствами классов. Согласно *OWL* [6], онтологии состоят из последовательности аннотаций, аксиом и фактов. Аннотации описывают некоторые сведения об онтологии. Факты представляют собой экземпляры классов (представители), типы или утверждения об эквивалентности или различии экземпляров. Аксиомы обеспечивают информацию о классах и свойствах. Каждая аксиома класса содержит коллекцию более общих классов, т. е. классов, являющихся базовыми к данному классу, и набор ограничений на свойства. Ограничения описывают надклассы свойств, сколько элементов допустимо и набор этих элементов. Также в аксиомах описывается эквивалентность классов и характеристики свойств.

Свойства могут иметь характеристики симметричности, транзитивности, функциональности и инверсной функциональности.

Отношение $R(X, Y)$ называется транзитивным, если для экземпляров понятий $\forall x \in \bar{X}, y \in \bar{Y}, z \in \bar{Z}$ выполняется соотношение:

$$p(x, y) \wedge p(y, z) \Rightarrow p(x, z),$$

где $\bar{X}, \bar{Y}, \bar{Z}$ — множества экземпляров понятий X, Y, Z соответственно, $p(x, y), p(y, z), p(z, x)$ — экземпляры отношений $R(X, Y), R(Y, Z), R(X, Z)$ соответственно.

Отношение $R(X, Y)$ называется симметричным, если $\forall x, y$ выполняется условие $p(x, y) \rightarrow p(y, x)$.

Отношение $R(X, Y)$ называется функционально-эквивалентным (или функциональным), если $\forall x, y, z$ верно соотношение $p(x, y) \wedge p(x, z) \Rightarrow y = z$.

Отношения $P_1(X, Y)$ и $P_2(Y, X)$ называются инверсными, если $\forall x, y$ верно соотношение $p_1(x, y) \Leftrightarrow p_2(y, x)$. На отношения наследования наложена по умолчанию характеристика транзитивности, т. е. если класс X является подклассом Y , а класс Y является подклассом Z , то класс X является подклассом Z . Транзитивность также является характеристикой отношений агрегации, т. е. отношений «быть частью». Если класс X содержит класс Y , а класс Y содержит класс Z , то класс X содержит класс Z .

На свойства могут накладываться ограничения. Первый тип ограничений — ограничение типов данных: свойство имеет тип данных из некоторого ограниченного набора типов данных («целое», «беззнаковое целое», «строка», «число с плавающей точкой», «булевская переменная», «длинное целое» и т. п.). Второй тип ограничений обычно связан с отношениями. Это ограничения на допустимые значения экземпляра класса Y в отношениях $P(X, Y)$. Допустимость значений определяется множеством существующих экземпляров класса $Y: \bar{Y}$. Ограничения могут накладываться и на мощность множества экземпляров отношений $P(X, Y): |P(X, Y)|$. Допускается указывать минимальное, максимальное и точное число в множестве. К ограничениям относится также требование на равенство экземпляров Y в отношениях $P(X, Y)$ некоторым определённым экземплярам понятия Y . Допускается и предикат из ограничений.

Онтологии предлагается рассматривать как концептуальную основу декларативного описания понятий и семантического ядра, выполняющих описательные и управляющие функции в ИУС.

В рамках описания ИУС предлагается выделить три основные области (рисунок 2) понятий (онтологии): понятия предметной области, понятия области управления сервисами, понятия ИТ-области, — а также определить базовую, общую для них онтологию.

Общая онтология содержит понятие базового объекта и некоторые другие понятия, которые в равной степени принадлежат к любой из онтологий.

Онтологии предметной области выделяются в связи с тем, что ИУС предназначена для решения задач жизнедеятельности вуза в различных предметных областях. Поэтому ИУС должна оперировать понятиями предметной области.

Область ИТ-онтологий выделяется для реализации идеи отображения предметной области на область ИТ. Область управления процессами выделяется в связи с тем, что, во-первых, ИУС представляет собой автоматизированные сервисы, а, во-вторых, управление сервисами в равной степени относится к предметной области и к области ИТ.

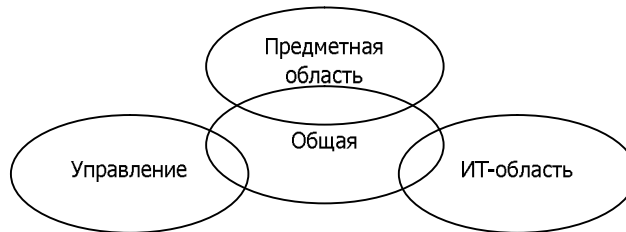


Рис. 2. Понятия в составе ИУС

Итак, ИУС можно представить как

$$IS = \{BS, SO, IT, R_s\} \quad (1)$$

где $BS = \{b_k, k = \overline{0, K-1}\}$ — понятия области управления сервисов (и отношений между ними),

$SO = \{s_i, i = \overline{0, I-1}\}$ — множество понятий предметной области (в том числе отношений между понятиями), $IT = \{t_j, j = \overline{0, J-1}\}$ — множество понятий управления информационной

средой и отношений между ними, $R_s = \{r_{ij}, i = \overline{0, I-1}, j = \overline{0, J-1}\}$ — отношения между понятиями предметной области, ИТ-области и области управления сервисами:

$$R_s = \{R_{ss}, R_{st}, R_{ot}, R_{bs}\} \quad (2)$$

Здесь:

$$R_{ss} = \{BS \otimes S\} \quad (3)$$

описывает множество отношений между областью управления сервисами и понятиями предметной области — сервисы могут создавать, изменять, просматривать, удалять экземпляры понятий предметной области.

Множество

$$R_{SI} = \{BS \otimes IT\} \quad (4)$$

описывает отношения между областью управления сервисами и понятиями ИТ-области. Эти отношения включают назначения прав пользователей на выполнение процессов, отношения сервисов со структурными компонентами и т. п.

Множество

$$R_{OI} = \{SO \otimes IT\} \quad (5)$$

описывает отношения между понятиями предметной области и ИТ-области, т. е. отражение объектов реального мира на ИТ-объекты.

Отношения R_{BS} описывают отношения, которые определены между понятиями в любой онтологии.

Таким образом, чтобы разработать эффективно сопровождаемую и эксплуатируемую ИУС вуза, необходимо разработать модель системы, представляющую собой онтологическое описание понятий и отношения между понятиями, семантическое ядро (с возможностью расширения) и механизм трансформации онтологического описания.

Семантическое ядро представляет собой элементарную функциональность (структурные компоненты). Данная функциональность может использоваться в различных подсистемах ИУС, из структурных компонентов могут быть составлены другие, более сложные, функциональности. Всё, что может быть выделено как общая функциональность, которая используется в ИУС более одного раза, реализует семантическое ядро. К семантическому ядру также относятся процедуры интерпретации понятий и отношений между ними.

Семантическое ядро позволяет в небольшие сроки разрабатывать новые сервисы и изменять существующие.

В обобщённую функциональность ИУС входит управление пользователями и их правами, маршрутизация запросов, управление эффективной работой СК, интерпретация понятий и отношений между ними, создание и редактирование экземпляров понятий и отношений, а также механизм создания отчётов из различных систем на основе интерпретации понятий и отношений между понятиями. Онтологическое описание содержит онтологии предметных областей деятельности вуза, управления процессами и описание понятий ИТ-области, а также отношения между понятиями.

Структурная схема концепции ИУС вуза представлена на рисунке 3. В основе концепции лежит онтологический подход, обеспечивающий разработку ИУС с использованием онтологической модели. Элементарная функциональность, входящая в семантическое ядро, реализует управляющие функции ИУС (авторизацию, аутентификацию, извлечение экземпляра понятия и т. п.) и функциональность предметной области (расчёт штатного расписания, составление отчётов и т. п.).

Кроме этого в семантическое ядро входят компоненты интерпретации правил поведения системы.

Для реализации алгоритмов управления процессами онтологическая модель содержит онтологии предметных областей деятельности вуза, области управления процессами и ИТ-области, а также отношения между онтологиями.

Возможность изменения онтологической модели реализуется с помощью инструмента создания понятий, отношений между понятиями, ограничений на атрибуты и создание, редактирование и удаление экземпляров понятий.

Онтологическое описание предметной области, в свою очередь, является совокупностью описаний предметных областей каждого сервиса (рисунок 4).

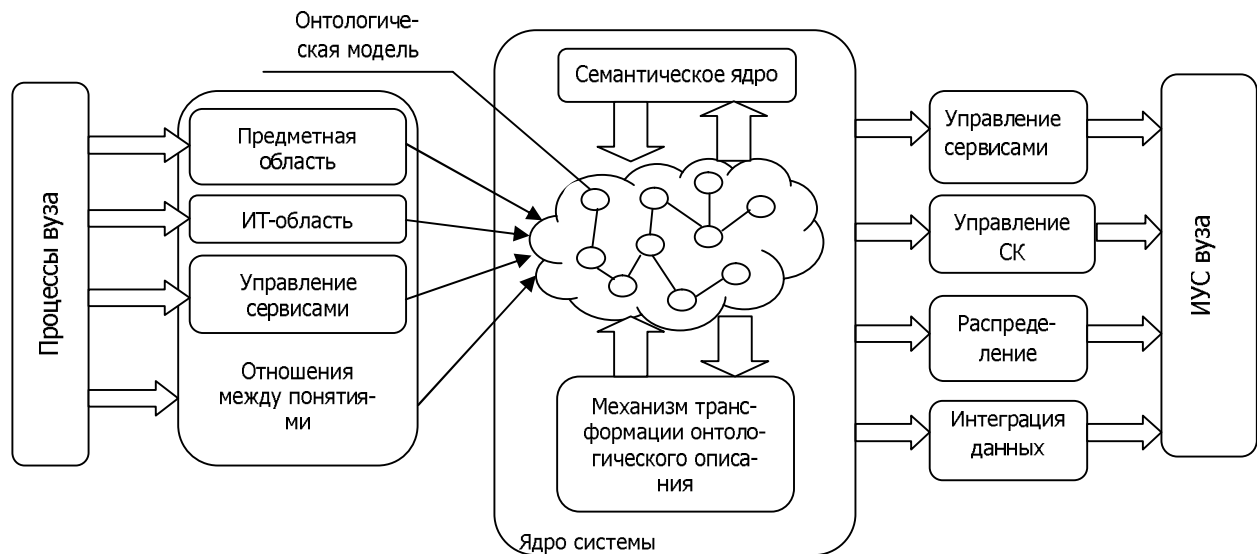


Рис. 3. Схема концепции ИУС вуза

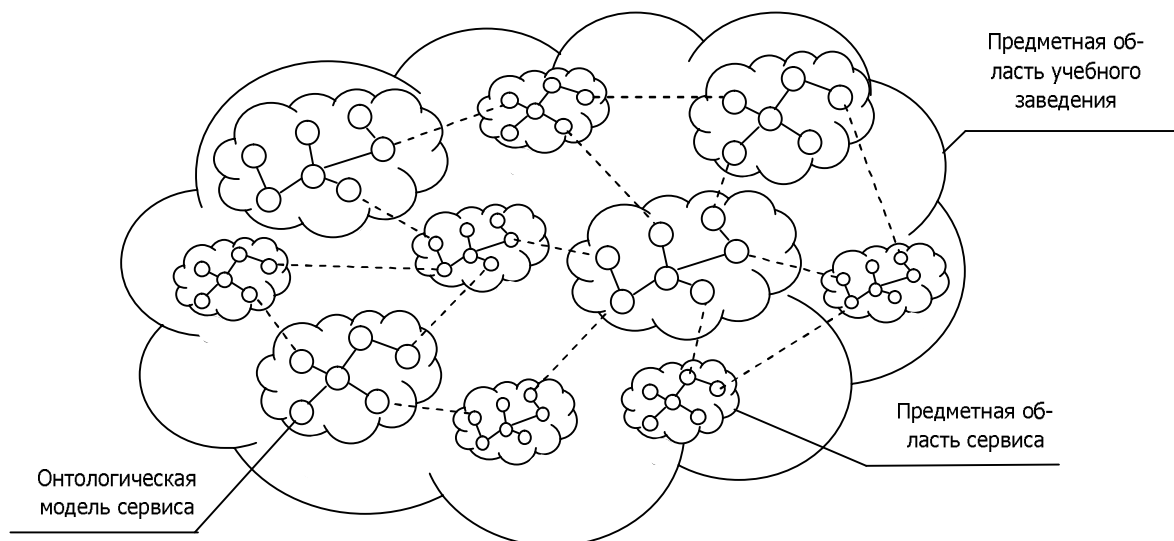


Рис. 4. Онтология предметных областей

Механизмы трансформации онтологических описаний представляют собой инструментарий, обеспечивающий создание новых и изменение существующих понятий, установление между ними связей, определение ограничений на атрибуты понятия, на отношения между понятиями и на их удаление. Кроме этого механизм онтологических описаний включает создание и редактирование экземпляров любого из существующих понятий и описание экземпляров отношений между понятиями.

Заключение. Предлагаемая архитектура построения ИУС вуза обеспечивает масштабирование по ряду направлений, что в свою очередь, позволяет говорить о построении адаптивной системы. Гибкие настройки системы позволяют адаптироваться к меняющимся условиям внешней среды. Использование сервис-ориентированной технологии даёт возможность автоматически поддерживать качество данных, а также снижает временные и материальные затраты. Таким образом, систему можно внедрять по мере автоматизации отдельных направлений деятельности.

Использование единообразного описания предметной области посредством онтологического моделирования обеспечивает высокую интеграцию данных при постепенной автоматизации. В итоге общая онтологическая модель является суммой онтологических моделей внедряемых сервисов и модели управления ими.

Библиографический список

1. Лодон, Дж. Управление информационными системами / Дж. Лодон, К. Лодон. — 7-е изд. — Санкт-Петербург: Питер, 2005. — 912 с.
2. Левенец, В. В. Один подход к проектированию и внедрению информационных систем (с применением методов КА и П) / В. В. Левенец // Теоретические и прикладные вопросы современных информационных технологий (ТПВСИТ): тр. VIII Всерос. науч.-техн. конф. — Улан-Удэ: ВСГТУ, 2005. — С. 274—276.
3. Требования к отраслевой информационной системе сферы образования Российской Федерации // Телекоммуникации и информатизация образования. — 2002. — № 1. — С. 18—28.
4. Бездушный, А. А. Схемы метаданных ЕНИП: практика применения OWL в ЕНИП / А. А. Бездушный, А. Н. Бездушный, В. А. Серебряков // Информационное обеспечение науки. Новые технологии. — Москва, 2005. — С. 155—182.
5. Ландэ, Д. В. Основы интеграции информационных потоков: [монография] / Д. В. Ландэ. — Киев: Инжиниринг, 2006. — 240 с.
6. OWL Web Ontology Language. Overview [Электрон. ресурс] / The World Wide Web Consortium (W3C). — Режим доступа: <http://www.w3.org/TR/owl-features/> (дата обращения: 10.03.12).
7. Gruber, T. R. A Translation Approach to portable ontology specification: Technical Report KSL 92-71 / T. R. Gruber. — Stanford: Stanford University; Knowledge Systems Laboratory, 1993. — 23 p.
8. Noy, N. F. Ontology development 101: A Guide to Creating Your First Ontology: Technical Report KSL 01-05 / N. F. Noy, D. L. McGuinness. — Stanford: Stanford University; Knowledge Systems Laboratory, 2001. — 25 p.
9. Hao, H. What is Service-oriented Architecture [Электрон. ресурс] / H. Hao. — Режим доступа: <http://www.xml.com/pub/a/ws/2003/09/30/soa.html> (дата обращения: 12.03.12).

Материал поступил в редакцию 30.08.2012.

References

1. Laudon, J., Laudon, K. *Upravleniye informatsionnymi sistemami*. [Management information systems.] 7th edition. St. Petersburg: Piter, 2005, 912 p. (in Russian).
2. Levenets, V.V. *Odin podkhod k proektirovaniyu i vnedreniyu informatsionnykh sistem (s primeneniym metodov KA i P)*. [One approach to information systems design and implementation (using concept analysis and design techniques).] *Teoreticheskiye i prikladnyye voprosy sovremennykh informatsionnykh tekhnologiy (TPVSIT): trudy VIII Vserossiyskoy nauchno-tekhnicheskoy konferentsii*. [Proc. VIII All-Russian sci. and tech. conf.: "Theory and application of information technologies".] Ulan-Ude: VSGTU, 2005, pp. 274—276 (in Russian).
3. *Trebovaniya k otraslevoy informatsionnoy sisteme sfery obrazovaniya Rossiyskoy Federatsii*. [Requirements for branch information system of education sphere in the Russian Federation.] *Telekommunikatsii i informatizatsiya obrazovaniya*, 2002, no. 1, pp. 18—28 (in Russian).
4. Bezduzhny, A.A., Bezduzhny, A.N., Serebryakov, V.A. *Skhemy metadannykh YeNIP: praktika primeneniya OWL v YeNIP*. [Metadata diagrams of CRIA (Common Research Information Area): practical application of OWL in CRIA.] *Informatsionnoye obespecheniye nauki. Novyye tekhnologii*. Moscow, 2005, pp. 155—182 (in Russian).

5. Lande, D.V. *Osnovy integratsii informatsionnykh potokov*. [Principles of information flows integration.] Monograph. Kiev: Inzhiniring, 2006, 240 p. (in Russian).
6. OWL Web Ontology Language. Overview. The World Wide Web Consortium (W3C). Available at: <http://www.w3.org/TR/owl-features/> (accessed: 10.03.2012).
7. Gruber, T.R. A Translation Approach to portable ontology specification: Technical Report KSL 92-71. Stanford: Stanford University; Knowledge Systems Laboratory, 1993, 23 p.
8. Noy, N.F., McGuinness, D.L. Ontology development 101: A Guide to Creating Your First Ontology: Technical Report KSL 01-05. Stanford: Stanford University; Knowledge Systems Laboratory, 2001, 25 p.
9. Hao, H. What is Service-oriented Architecture. Available at: <http://www.xml.com/pub/a/ws/2003/09/30/soa.html> (accessed: 12.03.2012).

EDUCATIONAL INSTITUTION ADMINISTRATION AUTOMATION BASED ON ONTOLOGICAL MODELING AND SERVICE-ORIENTED ARCHITECTURE

A. A. Petrikin, I. V. Boguslavskiy
(Don State Technical University)

The current state of the national universities automation is touched upon. Some aspects of the automation being the key ones in building the information control system (ICS) are defined. The tasks solved by the introduction of ICS in the university are formulated. It is concluded that the service-oriented architecture (SOA) is best suited for the ICS. The formal definition of the service-oriented architecture is given, the benefits of the service-oriented technology introduction into the university information space are disclosed. For illustrative purposes, the multilevel university ICS diagram is presented in the graphical format. The service redundancy problem specific to the most information systems when in long-life operation, is also considered. As a solution to this problem, the use of the structural meta-service that coordinates the interaction between ICS structural components is proposed. The ICS mathematical model is suggested. Generalizing the ICS management policy based on SOA, the authors draw the conclusion on the necessity for the unified description of the domain. The benefits of the ontological approach for the domain conceptualization are considered. It is concluded that to develop the ontological description of the concepts and relations between concepts, the semantic kernel, and the mechanism of the ontological description transformation is essential. Both the university conception ICS diagram and the structure of the domain ontological model as the ontological model symbiosis of individual ICS services are presented in the graphical view.

Keywords: automation, information management system, service, service-oriented architecture, structure, structural component, ontology, ontological modeling, domain, semantic kernel, mathematical model, unification, higher educational institution.