



УДК 004.942, 004.042, 338.46

**А.А. Петренко**, аспирант, **А.И. Петренко**, д-р техн. наук  
Ин-т прикладного системного анализа Национального технического  
университета Украины «Киевский политехнический ин-т»  
(Украина, 03056, Киев, пр-т Победы 37, корп. 35,  
тел. +38 (067) 5972077, e-mail: tolja.petrenko@gmail.com)

### **Семантическое модельно-управляемое моделирование архитектуры системы сервисов на основе доменных онтологий**

Исследована методология моделирования системы сервисов с современной сервис-ориентированной архитектурой и ее модификациями, основанная на совместном использовании онтолого-управляемого (Ontology-Driven Development) и модельно-управляемого (Model-Driven Development) подходов к моделированию архитектуры бизнес-процессов. Рассмотрены взаимодействие и интеграция различных моделей, основанных на использовании семантических онтологий бизнес-процессов. Показано применение семантического модельно-управляемого подхода к разработке сервис-ориентированных программных систем.

Досліджено методологію моделювання системи сервісів із сучасною сервіс-орієнтованою архітектурою і її модифікаціями, базовану на спільному використанні онтолого-керованого (Ontology-Driven Development) і модельно-керованого (Model-Driven Development) підходів до моделювання архітектури бізнес-процесів. Розглянуто взаємодію та інтеграцію різних моделей, основаних на використанні семантичних онтологій бізнес-процесів. Показано застосування семантичного модельно-керованого підходу до розробки сервіс-орієнтованих програмних систем.

*К л ю ч е в ы е с л о в а: сервис-ориентированная архитектура, семантические сервисы, модельно-управляемое моделирование, онтология, онтолого-управляемое моделирование, бизнес-модель, универсальный язык моделирования UML.*

Мир становится сервис-доминирующим в связи с возрастающим значением сервисов в наиболее развитых экономиках мира [1, 2]. Сектор сервисов доминирует в экономической активности развитых стран, обеспечивая более 70 % валового внутреннего продукта. Системы сервисов являются динамическими конфигурациями людей, технологий, организаций и средств обмена информацией, которые создают и обеспечивают ценности для клиентов, поставщиков и других заинтересованных сторон. Доминирова-

© А.А. Петренко, А.И. Петренко, 2016

ние сервисов в хозяйственной деятельности предприятий известно как индустрия сервисов.

Индустрию сервисов можно рассматривать как результат развития информационно-коммуникационных технологий, обеспечивающих решения, которые не были возможны ранее. В этом случае производство промышленных товаров и сельскохозяйственной продукции остается, как и ранее, актуальным, однако внимание фокусируется на использовании преимуществ и выгод, обеспечиваемых сервисами. Термин «сервис» имеет два основных общепринятых значения:

1) экономическое, бизнес-ориентированное, как, например, в выражении «сервисный сектор экономики»;

2) информационное, ИТ-ориентированное, как в выражениях «веб-сервис» или «сервис-ориентированная архитектура (СОА)».

В первом случае внимание сосредоточено на взаимодействии при обмене и нематериальном характере сервиса, во втором — на технологии программного обеспечения, которая позволяет поддерживать совместимость различных программных продуктов, созданных различными производителями. В современных системах сервисов для экономики эти два значения объединяются, так как внедрение современной системы сервисов предусматривает интеграцию информационных систем как подсистем общей организационной системы [3].

При поддержке компании IBM за короткий срок (приблизительно пять лет) новое направление — наука о сервисах, менеджменте и инжиниринге (SSME-Service science, Management and Engineering) получило широкое распространение в мире, что мотивирует проведение исследований в области науки о сервисах [4, 5]. Интенсивно развивающаяся наука о сервисах имеет по сути междисциплинарные признаки с долгосрочной целью стать действительно междисциплинарной отраслью. Основной движущей силой науки о сервисах есть желание промышленности получить (в потенциально огромных количествах) новый тип профессионалов, которые должны иметь глубокие знания в нескольких существующих дисциплинах, а также обладать навыками интегрирования науки и искусства дизайна сервисов и реализации ценностей путем объединения технологий, бизнес-моделей и социально-организационных инноваций для улучшения бизнес- и социальных систем.

При моделировании системы сервисов как бизнес-процессов следует рассматривать три аспекта:

объекты, с которыми оперирует бизнес;

процессы, которые он выполняет;

события, управляющие изменениями процессов и объектов.

Соответственно можно определить три типа моделирования: информационное, функциональное и событийное. Это распределение зависит от применяемого подхода, но в любом из них всегда присутствуют два действия: первое — сбор информации и моделирование бизнеса, второе — построение архитектуры будущей системы, что является важным шагом на пути к ее созданию. Указанные три типа моделирования объединяются в одной процедуре, которая основана на совместном использовании онтолого-управляемого (Ontology-Driven Development, ODD) и модельно-управляемого (Model-Driven Development, MDD) подходов к моделированию архитектуры бизнес-процессов.

Рассмотрим базовую методологию, которая объединяет подходы MDD и ODD, в результате чего формируются формализованные концептуальные модели, используемые как во время выполнения процесса, так и при его разработке с возможностью контекстного формирования выводов и автоматизированной генерации кодов исходных программ.

**Моделирование бизнес-процессов на основе ODD.** В самом общем случае под онтологией понимают формальное и явное описание общей концептуализации домена. Такое определение подчеркивает главные отличительные черты онтологии: 1) необходимость определенного языка для описания онтологии; 2) использование абстракций отрасли, которые включают в себя соответствующие понятия; 3) согласованную точку зрения на представление основ знаний домена.

Таким образом, онтологию следует трактовать как структурную спецификацию некоторой предметной области, ее формализованное представление, которое включает словарь (или имена) указателей на термины этой области и логические выражения, описывающие, как они соотносятся между собой. Такой словарь поддерживает обмен знаниями о некоторой предметной области и множество связей, установленных между терминами в этом словаре. Семантика поддерживает и обеспечивает общее понимание знания домена, которое может быть передано между людьми и программным обеспечением.

Консорциумом W3C (World Wide Web Consortium) разработаны два основных стандарта для представления онтологии, а именно RDF Schema (RDFS) и язык OWL (Web Ontology Language) [6]. Онтологии, описанные языком OWL, состоят из:

- классов как наборов отдельных лиц;
- частных лиц как экземпляров классов (или объектов домена);
- свойств как бинарных отношений между частными лицами.

Можно определить доменные особенности, диапазоны изменений и правила логических выводов в онтологии. Кроме того, некоторые логи-

ческие выводы можно использовать, чтобы вывести дополнительные сведения о знаниях, которые были явно указаны в OWL онтологии. Такие логические выводы можно реализовать на уровне класса, свойства или уровня экземпляра. Они включают членство класса, эквивалентность классов, последовательности, классификацию информации, получение дополнительных свойств с использованием транзитивности или эквивалентности.

В работе [7] приведена онтология высокого уровня Onto-ServSys для системы сервисов, которая имеет вид семантической сети и не зависит от конкретной проблемы или области. Семантическая сеть — это граф, содержащий узлы (понятия) и ребра (бинарные отношения между понятиями). Понятиями Onto-ServSys являются системные понятия, бизнес-организационные концепции и концепции системы сервисов. Приведенная онтология поддерживает трехмерное представление о том, что такое сервис. С точки зрения человека сервис является свойством системы, связанным с эффективностью, этическими и эстетическими показателями. С точки зрения объективного результата сервис является эффективным системным фактором или событием, а с точки зрения потока событий сервис является выходом системы. Сервис может генерировать человеко-значимые результаты, которые являются эффективными, этическими или эстетическими.

Сервис имеет несколько основных атрибутов, одним из которых является управляемость. Этот атрибут состоит из трех концепций: объективной метрики оценки, входов и выходов. Входами и выходами могут быть энергия, материалы, данные-информация-знание или поток событий.

Кроме онтологии высокого уровня существуют отраслевые онтологии, которые описывают лексику, связанную с выбранной областью, или ее бизнес-процессы, онтологии заданий, описывающие общие задачи или этапы деятельности (например, мониторинг или измерения) и онтологии приложений, описывающие понятия, зависящие как от определенного домена, так и от конкретной задачи, поэтому они часто базируются на обеих предыдущих онтологиях.

Определим бизнес-процесс, реализуемый приложением, как способ организации работы и ресурсов, позволяющий достичь цели с помощью набора мероприятий, которые проводятся в порядке, определенном потоком задач. Этот поток фактически определяет основные характеристики бизнес-процесса: цель процесса (что именно?), деятельность (как?), ресурсы (каким образом?) и порядок деятельности (когда?).

**Онтология бизнес-модели** ВМО (Business Model Ontology) является инструментом для разработки и анализа бизнес-процессов предприятия или мероприятия. ВМО описывает бизнес-модель с помощью девяти основных строительных блоков:

*сегмент потребителей* (Customer segments) — определяет различные группы людей или организаций (предприятий), стремящихся получить сервисы;

*стоимость предложения* (Value propositions) — описывает распределение сервисов, создающих ценность для конкретного сегмента потребителей;

*каналы* (Channels) — описывают взаимодействие компании с потребительским сегментом, чтобы доставить ценностное предложение;

*взаимоотношения с клиентами* (Customer relationships) — описаны типы отношений, которые организация устанавливает с конкретными сегментами клиентов;

*потоки доходов* (Revenue streams) — денежные средства, получаемые компанией от каждого сегмента клиентов;

*основные ресурсы* (Key resources) — наиболее важные активы, необходимые для того, чтобы бизнес-модель сделать рабочей;

*основные направления деятельности* (Key activities) — описаны наиболее важные мероприятия, которые организация должна выполнить, чтобы ее бизнес-модель работала;

*основные партнеры* (Key partnerships) — сеть поставщиков и партнеров, делающих бизнес-модель рабочей;

*структура расходов* (Cost structure) — описывает все расходы, понесенные для обеспечения работы бизнес-модели.

Свойства элементов бизнес-модели приведены в табл. 1.

Из базовых бизнес-элементов составим онтологию кросс-организационных бизнес-процессов, которые описывают задачу наблюдения и мо-

Таблица 1. Элементы онтологии бизнес-модели

Элемент	Пояснение
Определение	Точное описание элемента бизнес-модели
Часть чего-то	Определяет отношения элемента с четырьмя базовыми классами бизнес-модели: Service (сервис), Management (управление), Actors (роли), Cost-Benefit (цена-выгода) [8]
Связь	Описывает, с какими другими элементами онтологии связан элемент
Набор	Указывает, какие субэлементы могут входить в элемент
Вхождение элемента	Определяет число разрешенных вхождений элемента или субэлемента внутри онтологии
Атрибуты	Списки атрибутов элемента или субэлемента. Допустимые значения атрибута в пределах {значение 1, значение 2}. Частота их появления указывается в скобках (например, 1–n). Каждый элемент и субэлемент имеет два стандартных атрибута, которыми являются название (NAME) и описание (DESCRIPTION), содержащие цепочку символов {abc}
Ссылки	Указывает основные ссылки, связанные с элементом бизнес-модели

бильной медицинской помощи пожилым людям с хроническими болезнями (MobiMed). Рассмотрим следующий возможный сценарий организации такой помощи. Предположим, что в результате проведения долгожданной медицинской реформы в Украине в 2018 г. в Киеве, как и других областных центрах, будет создан Центр общественного здоровья (ЦОС) с сетью районных Сервисных центров здоровья (СЦЗ), связанных с местными больницами, поликлиниками, семейными врачами, службами скорой медицинской помощи и службой ликвидации последствий чрезвычайных ситуаций. Заинтересованные жители Киева (например, пожилые люди с хроническими болезнями) смогут заключить с ЦОС договоры на предоставление сервисов постоянного дистанционного наблюдения за состоянием здоровья, уходу и мобильной медицинской помощи. Они будут приписаны к одному из СЦЗ, где на сервере будут накапливаться данные об истории их болезни в виде персональной электронной медицинской карты. В договоре четко оговариваются меры, которые будут предоставлены в виде помощи пациенту, при каких симптомах и в какие сроки (например, при тяжелом сердечном приступе необходимая операция должна быть осуществлена в течение двух часов после получения сигнала тревоги).

Для организации мониторинга состояния здоровья пациент оснащается беспроводными датчиками, которые крепятся на его теле и подключены к его смартфону, используемому как персональная система мониторинга здоровья (ПСМЗ). Его дом также оснащается различными датчиками, подключенными беспроводно к смартфону, которые способны собирать контекстную информацию, такую как, например, время, место нахождения пациента (через GPS смартфона), положение пациента (стоит, сидит, лежит, упал), активность пациента в настоящее время (ходьба, физические упражнения, вождение, прием лекарств, сон), с кем пациент взаимодействует и есть ли кто-то рядом, параметры среды (температура, атмосферное давление, наличие аллергенных раздражителей). ПСМЗ настраивается на персональное медицинское состояние пациента; при этом параметры мониторинга однозначно оптимизируются под пациента с помощью алгоритма машинного обучения, реализованного в устройстве.

Если жизненные показатели, по которым ведется мониторинг, приблизятся к опасной черте, дежурный врач СЦЗ, с которым пациент заключил договор, незамедлительно информируется об этом через ПСМЗ и Портал наблюдения СЦЗ. Он немедленно реагирует на тревогу следующим образом:

- дистанционно настраивает ПСМЗ и датчики пациента на максимальную пропускную способность и повышенное качество информации, снимаемой с датчиков, чтобы перепроверять в возможном диагнозе при-



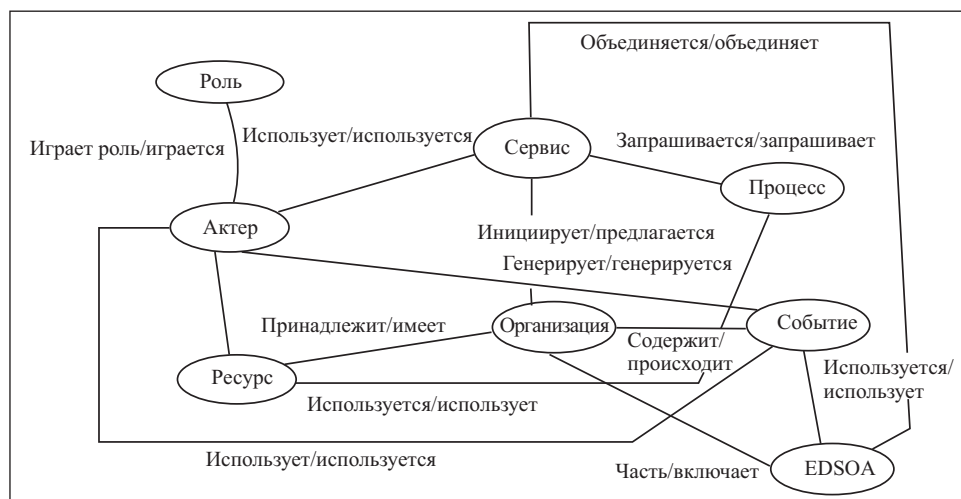


Рис. 1. Пример онтологии кросс-организационных бизнес-процессов

чины сигнала тревоги; затем переключает смартфон пациента на голосовую связь, чтобы, не беспокоя пациента, дать ему первые нужные советы (типа «остановиться на месте», «сесть», «лечь» и др.), исходя из полученной контекстной информации о местоположении пациента, активной его деятельности, истории приема лекарств, нахождения кого-либо рядом и др.;

- сообщает о ситуации в диспетчерскую ближайшей скорой помощи, семейному врачу и больнице, куда скорая помощь доставит пациента, а также родственникам пациента (если он был один во время кризиса); при этом во время обратной дороги скорой помощи ПСМЗ пациента автоматически повторно перестраивается на непрерывную отправку жизненно важных данных в больницу, куда отвезли пациента.

Этот сценарий свидетельствует о том, что скоординированная стратегия электронного здравоохранения, используя персональные системы мониторинга здоровья, может помочь реализации адресных приоритетов ЦОС в улучшении доступа к здравоохранению, профилактике хронических болезней, мониторингу и уходу за пожилыми людьми в сложных условиях. Эта стратегия может также способствовать уменьшению времени ожидания пациентов в поликлиниках и их пребывания в больницах.

На рис. 1. представлена возможная онтология кросс-организационных бизнес-процессов, описывающих задачу ухода и мобильной медицинской помощи пожилым людям с хроническими болезнями [9]. В данной онтологии использовано восемь базовых понятий: роль, актер, сервис, процесс, ресурс, организация, событие и EDSOA. Роль в концепции онто-

логии позволяет определять правила актерского поведения (пациента, семейного врача, дежурного врача СЦЗ или скорой помощи). Актер является организационным ресурсом, который играет роль как в инициировании процесса, так и на более детальном уровне выполнения задания. Дежурный врач СЦЗ, который играет роль регистратора событий, является примером такого актера в общественной организации. Пациент является другим примером актера в общественном секторе. Актер может осуществлять бизнес-процесс и использовать сервисы, необходимые для выполнения процесса. Например, регистратор, регистрирующий сигнал тревоги на портале наблюдения от ПСМЗ пациента, может инициировать процесс, называемый «реакция на тревогу». Во время своей работы регистратор использует сервисы для организации помощи пациенту. Из рис. 1 видно, что СЦЗ, скорая помощь, семейный врач и другие организации имеют ресурсы, которые они используют для выполнения процессов.

На рис. 1 указана также объединенная архитектура сервисов EDSOA (Even-Dreven SOA). События происходят во время выполнения процесса и передаются между интегрированными интерактивными сервисами, которые оркестрованы в EDSOA. Учет организационных событий имеет важное значение для успешной реализации предоставления интегрированных сервисов, так как координация между сервисами дает представление о том, как сервисы зависят друг от друга и какие актеры требуют какие услуги. Ограничения важны, так как показывают, какие отношения возможны. Например, дежурный врач СЦЗ может не вызвать скорую помощь, пока на портале наблюдения не увидит сигнал тревоги и не классифицирует его.

Реализация EDSOA на онтологической основе может обеспечить предоставление интегрированных сервисов и оркестровку событий на практике созданием базы данных (БД), в которую участники кросс-организационных медицинских бизнес-процессов могут добавить информацию о своих сервисах, ресурсах, процессах и событиях. Система может затем использовать эту информацию, чтобы исследовать, имеет ли отношение какой-либо актер к новому событию, и проконтролировать процесс предоставления необходимых сервисов.

Онтологии обычно поддерживают высокий уровень взаимодействия между пользователями (актерами) и «умными» приложениями, используя механизмы логического вывода в формализованных моделях для объяснения оснований принятия решений. Онтологии необходимо учитывать при разработке приложений для того, чтобы обеспечить их возрастающую сложность. Например, приложения могут иметь сервис-ориентированный характер и создаваться путем композиции и оркестровки веб-сервисов из



некоторого репозитория, наполненного разработками многих поставщиков сервисов. Семантические онтологии более богаты и могут описывать адекватно функциональности приложений, способствуют более эффективному повторному использованию знаний и управления логическими выводами. Хорошо определенную онтологию можно непосредственно преобразовать в код приложения, используя универсальный язык моделирования UML (Unified Modeling Language), в схему БД, в абстрактный интерфейс и др.

**Моделирование бизнес-модели сервисов на основе MDD.** Конфигурация и координация сервисов в выбранной архитектуре (сервис-ориентированной SOA, событийно-управляемой EDA и мультиагентной MAS) [10] и композиция самых сервисов и процессов одинаково важны в современной технологии обслуживания. Существующие сервисы могут быть использованы повторно и скомпонованы для того, чтобы сформировать бизнес-процесс или рабочий поток (workflow) [8, 11].

MDD подход основан на увеличении уровня абстракции для преодоления сложности при разработке приложений, которая связана с применением только ODD подхода. MDD подход (в сочетании с онтологией) поддерживает абстрактные семантические описания и композиции сервисов при моделировании систем сервисов. На основе трехслойного MDD подхода возможно создание следующих моделей [12—14].

*Вычислительная независимая модель CIM (Computing Independent Model),* которая описывает систему с вычислительной независимой точки зрения, освещая структурные аспекты системы. При этом процессно-ориентированная онтология описывает два типа сущностей домена: объекты, являющиеся статическими сущностями, и процессы, являющиеся динамическими сущностями, а также три типа отношений между ними: «is a» (подкласс отношения), «has part» (компонентные отношения) и «depends» (отношение зависимости).

*Независимая от платформы модель PIM (Platform Independent Model),* которая может рассматриваться как обоснование системы сервисов с точки зрения технологии нейтральной виртуальной машины, или вычислительной абстракции. Архитектура и процессы являются ключевыми аспектами на этом уровне моделирования сервисов. Архитектура влияет на сервисы, их архитектурные конфигурации и типы взаимодействия (удаленный вызов и активация сервисов) между различными сервисами, или компонентами системы программного обеспечения. В данном случае онтология используется для определения этих аспектов.

*Модель для конкретной платформы PSM (Platform Specific Model),* которая направлена на конкретную технологию реализации системы сер-

висов. При создании этой модели осуществляется поиск необходимых сервисов через сеть. При этом используются онтологии для повышения точности поиска сервиса и открытие такого сервиса, в котором упоминается в точности искомое понятие, в отличие от произвольных сервисов, в описаниях которых встретилось заданное ключевое слово. Общеизвестно, что в различных предметных областях одни и те же понятия могут быть представлены различными терминами. Механизм онтологии в этих случаях позволяет формировать осмысленные, иерархические взаимосвязи между сервисами, т.е. реализовать композицию сервисов, способную удовлетворить запрос пользователя, хотя в ее описании может не быть некоторых ключевых слов из входного запроса.

Как известно, для комбинирования сервисов в необходимые приложения используются процедуры «хореографии» и «оркестровки». Хореография веб-сервисов определяет взаимодействие между сервисами с помощью обменных уведомлений, а оркестровка описывает взаимодействие сервисов в рамках одного бизнес-процесса.

Технология веб-сервисов в настоящее время является наиболее распространенным средством реализации сервис-ориентированных концепций и приводит к становлению «Интернет сервисов», где концентрируются сервисы для всех областей жизни и бизнеса. Основой этой технологии является описание сервисов с абстрактным синтаксисом (Web Services Description Language, WSDL), семантических сервисов (например, Web Service Modeling Ontology, WSMO), бизнес-процессов приложения (Web Service Process Ontology, WSPO) и определение сервисного процесса (например, WS-BPEL, Business Process Execution Language).

Интероперабельность сервисов является основной задачей платформы веб-сервисов. Два метода используются на этом уровне: абстрактное описание сервисов для поддержки их открытия и удаленного вызова и стандартизированный монтаж (композиция) сервисов и процессов.

Следует признать, что веб-сервисы, описанные с использованием языка WSDL, содержат преимущественно синтаксическую информацию, и это затрудняет организацию автоматического поиска веб-сервиса и выполнения композиции сервисов. Поэтому предложено использовать также семантическую информацию в веб-сервисах, в результате чего появилось понятие семантических веб-сервисов (SWS) со своим языком описания OWL-S (Web Ontology Language for Web Services). Методы, инструменты и технологии семантического веб (Semantic Web), который является продолжением современного Интернета, усилиями мирового веб-консорциума W3C (World Wide Web Consortium) направлены на согласование информации в Интернете с ее семантикой и обеспечение интерпретации данных компьютерами наряду с людьми [15]. Поиск сервисов при этом

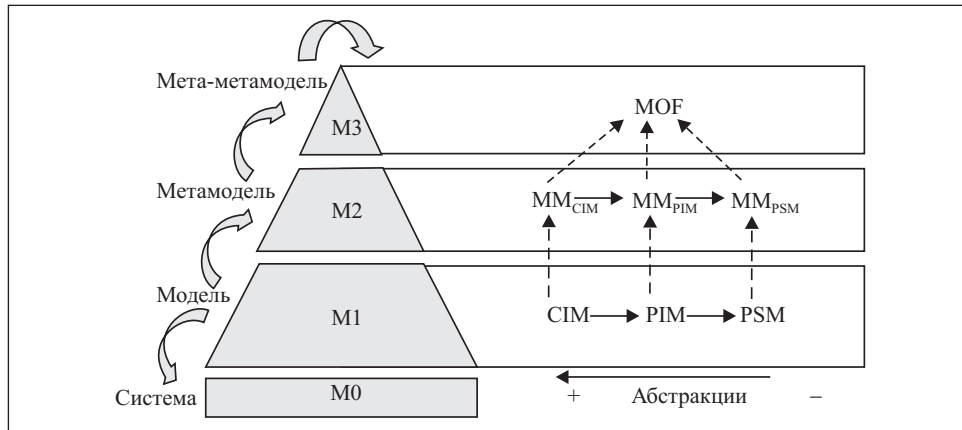


Рис. 2. Связь между моделями MDD и метамоделями

осуществляется с помощью оценки семантической близости между элементами онтологии сервисов, находящихся в репозитории, и семантическим запросом на нужный сервис [16].

При проектировании систем сервисов используется иерархическая система уровней описания семантики системы сервисов (рис. 2), предложенная OMG (Object Management Group) [17]:

M0: информационный уровень (фактические данные);

M1: уровень моделей (метаданные, или семантика данных);

M2: уровень метамodelей (мета-метаданные);

M3: уровень мета-метамodelей (семантика метамodelей), который часто называют Meta Object Facility (MOF).

Уровень M2 определяет модель языка моделирования, а M3 — логические выводы из понятий уровня M2.

Процесс проектирования системы сервисов можно смоделировать как процесс преобразований моделей, начиная с метамodelи в качестве входных данных и используя набор правил преобразования. При этом само преобразование является также моделью. Модели преобразования могут иметь различные области применения:

создание моделей нижнего уровня, и в итоге программного кода, начиная с моделей более высокого уровня;

отображение и синхронизация между моделями на том же уровне или на различных уровнях абстракции;

создание представления о системе на основании запросов;

изменение внутренней структуры системы для упрощения ее восприятия и внесения дальнейших изменений, не изменяя при этом существующей функциональности;

обратное проектирование от моделей нижнего уровня или кода до модели более высокого уровня.

Преобразование моделей — основной элемент в проектировании систем сервисов, обеспечивающий плавную обработку входящих онтологических моделей для дальнейшего генерирования, фильтрации и обновления целевых моделей. С помощью языков преобразования моделей реализуются различные типы преобразования, такие как модель-модель (ММ) или модель-код (МК). Особенность данного подхода состоит в том, что разработчик большую часть времени работает не с кодом, а с моделями. Например, отображение описания WSPO в описания WSMO переносит функциональные свойства (как синтаксические, так и семантические) в модель PSM. Это отображение превращает описание процессов в BPEL-скелет бизнес-процессов. WS-BPEL является языком реализации для выполнения процесса в виде оркестровки сервисов. Реализация WS-BPEL поддерживается сервисными механизмами от разных поставщиков. Поскольку язык WSPO включает описания процессов, похожие на WS-BPEL, модель WSPO может быть полностью переведена в модель WS-BPEL.

При объединении ODD и MDD подходов для рассматриваемого случая использования описаний WSPO, WSMO и WS-BPEL предложен следующий алгоритм разработки программного обеспечения [18]:

1. Построить независимые общую (GO) и доменную (DO) онтологии, при этом часть DO превращается в вычислительную независимую модель (CIM).

2. Преобразовать модель CIM в независимую от платформы модель PIM согласно правилам, приведенным в табл. 2.

Таблица 2. Правила преобразования моделей CIM в PIM

Правило	Аспект	Описание
CP0	Шаблон	Для каждого элемента модели CIM создать шаблон в модели PIM
CP1	Элемент процесса	Элемент модели PIM является элементом модели CIM
CP2	Состояние	Создание концепции по умолчанию для пред- и пост-состояний
CP3	Синтаксис	Для каждого in- и out-параметров процесса создать отдельный элемент объекта
CP4	Семантика	Создать пре- и постусловия в зависимости от наличия внешней дополнительной информации в виде ограничений
CP5	Процесс	Если выражения для процесса доступны в виде ограничений, то они создают сложный процесс, используя описания отношений в WSPO

3. Преобразовать модель PIM в модель конкретной платформы PSM согласно правилам, приведенным в табл. 3 (справедливым при выборе языков WSMO для описания онтологии и WS-BPEL — для оркестровки сервисов), и в сопутствующие модели (например, в схему БД), определить методику сравнения и аннотации сервисов.

4. Преобразовать части онтологий DO и GO в онтологию контекста (CO), используемую внешним приложением.

5. Применить преобразования PSM и сопутствующих моделей в коды приложений и программных артефактов.

Основное преимущество MDD — обеспечение нескольких преобразований заданной CIM для поддержки различных платформ. Например, можно обеспечить преобразование описаний OWL-S в код Java [19] в качестве альтернативы для WSMO и WS-BPEL, позволяя пользователю легко переключаться между языками платформы.

Взаимодействие и интеграция моделей — главный вопрос в модельно-управляемой методологии моделирования систем сервисов. В настоящее время перспективной является разработка общей метамодели ODM (Ontology Definition Metamodel) с отражениями в UML и OWL [20], а также разработка нового стандарта языка описания онтологии этой метамодели, проводимая международным консорциумом OMG, который поддержи-

Таблица 3. Правила преобразования моделей PIM в PSM

Правило	Аспект	Описание
PP1	WSMO	Из модели PIM, основанной на языке WSPO, отобразить отношения процессов в сервисные концепты WSMO и заполнить свойства messageExchange (обмен сообщениями) и Pre / postCond (пре- и постусловия)
PP1.1	WSMO обмен сообщениями	Отобразить WSPO in- и out- объекты в WSMO сообщения описаний обмена
PP1.2	WSMO пре- и постусловия	Отобразить WSPO пре- и постусловия в WSMO пре- и постусловия
PP2	WS-BPEL	Отношения композиционного процесса WSPO отобразить в BPEL-процессах
PP2.1	WS-BPEL партнерский	Для каждого процесса создать BPEL партнерский процесс
PP2.2	WS-BPEL оркестровка	Преобразовать выражение для каждого процесса в BPEL-вызове и перенести на сторону сервера клиентские BPEL-receive и -reply
PP2.3	WS-BPEL управление процессом	Преобразовать процессорные комбинаторы « ; », « + », « ! », «    » в соответствующие BPEL комбинаторы: sequence, pick, while, and flow (последовательность, выбор, в то же время, поток)

вает преобразование моделей на языках описания онтологии в модели на языках моделирования (например, UML).

## **Выводы**

Настало время перехода на новую парадигму программирования, связанную не с объектами, а с бизнес-процессами и их составной частью — бизнес-функциями. Ее идея — компоновка приложений путем обнаружения и вызова сервисов, доступных в сети, для выполнения некоторой задачи. Этот подход не зависит от конкретных языков программирования и операционных систем.

Объединенная архитектура SOA — это архитектура приложений, построенная на основе формализованных бизнес-процессов, функции которых представлены в виде многократно используемых сервисов с прозрачными описанными интерфейсами. Применение подхода MDD в сочетании с онтологией может помочь инженерам программного обеспечения разрабатывать и управлять сложными системами. Онтологии рассматриваются как CIM модели, поэтому концептуальные модели PIM и PSM также отражают семантику данного домена приложений, соответствующую общей доменной семантике независимо от конкретных потребностей приложений.

Таким образом, онтология является основой метамодели, позволяющей генерировать концептуальные модели для реализации конкретных информационных систем. Становление онтолого-ориентированной программной инженерии создает возможность определять программные проекты не только требованиями ТЗ и моделями, но и онтологией, образующей базу знаний предметной области и используемой в качестве платформы моделирования и логических выводов для управления моделями.

Предложенный подход к проектированию систем сервисов хорошо адаптируется к особенностям современных распределенных вычислительных сред (мультиоблака или грид), когда веб-сервисы могут находиться в различных географически разделенных репозиториях (возможно, в нескольких эквивалентных вариантах реализации), а их композиции в виде созданных приложений выполняются на различных вычислительных узлах среды, ресурсы которых освободились к моменту начала выполнения очередного сервиса композиции. В результате реализованное приложение имеет динамически изменяемую архитектуру и переменный компонентный состав.



## СПИСОК ЛИТЕРАТУРЫ

1. *Handbook of Services. Service Science: Research and Innovations in the Service Economy* / Maglio, P., Kieliszewski, C., Spohrer, J. (Eds). — N.Y.: Springer, 2010. — 480 p.
2. *Service Systems Implementation. Service Science: Research and Innovations in the Service Economy* / Demirkan, H., Spohrer, J., Krishna, V. (Eds). — N.Y.: Springer, 2011. — 310 p.
3. *Services Science: Fundamentals, Challenges and Future Developments* / Stauss B., Engelmann, K. Kremer, A., Luhn, A. (Eds). — Berlin: Springer, 2007. — 380 p.
4. *Петренко А.А.* Объекты и методы науки о сервисах // Системні дослідження і інформаційні технології. — 2015. — № 2. — С. 75—82.
5. *Кисельов Г.Д., Петренко О.О.* Наука про сервіси, менеджмент та інжиніринг як основа інноваційної діяльності // Вісн. ун-ту «Україна», Серія «Інформатика, обчислювальна техніка та кібернетика». — 2015. — №2(18). — С. 28—36.
6. *World Wide Web Consortium. Web Services Architecture.* — Режим доступа: <http://www.w3.org/TR/ws-arch>
7. *Onto-ServSys : a service system ontology* , Mora Manuel, Raisinghani Mahesh, Gelman Ovsei, Sicilia Miguel Angel (Contribs). — Режим доступа: <https://www.econbiz.de/Record/onto-servsys-a-service-system-ontology-morammanuel/10009127309>
8. *Business model generation: a handbook for visionaries, game changers and challengers*, Osterwalder, A. and Pigneur, I. (Eds). — New Jersey : Willey, 2010. — 278 p.
9. *Petrenko A.I.* Mobile health applications to support the diabetic patient and the doctor // Design & Test Symposium (EWDTS), 2014 East-West. — P. 1—6. — DOI:10.1109/ EWDTS. 2014.7027105
10. *Петренко О.О.* Порівняння типів архітектури систем сервісів // Системні дослідження і інформаційні технології. — 2015. — № 4. — С. 48—62.
11. *Egon L., Comuzzi M., Grefen P., Weisleder P.* The Service Dominant Business Model: A Service Focused Conceptualization // BETA publicatie. — WP 402 (working paper). — ISBN NUR982. — Eindhoven. — 2013. — 23 p.
12. *A Model-Driven Architecture Approach to the Efficient Identification of Services on Service-oriented Enterprise Architecture* / Alahmari S., De Roure D., Zaluska (Eds). // Proc. of the Second Workshop on Service oriented Enterprise Architecture for Enterprise Engineering in conjunction with the 14th IEEE International Enterprise Distributed Object Computing Conference. — Brazil: Vityria, 2010.
13. *Pahl Claus.* Semantic Model-Driven Architecting of Service-based Software Systems. — Режим доступа: <http://ceur-ws.org/Vol-244/paper3.pdf>
14. *Pahl Claus.* An Ontology for Software Component Matching // Intern. J. on Software Tools for Technology Transfer, Special Edition on Component-based Systems Engineering. — 2006. — № 7. — P. 1—10.
15. *Петренко І.А., Петренко О.О.* Автоматизовані методи пошуку і відкриття необхідних сервісів // Вісн. ун-ту «Україна», Серія «Інформатика, обчислювальна техніка та кібернетика». — 2015. — №1(17). — С. 55—64.
16. *Нгуен Ба Нгок, Тузовский А.Ф.* Модель информационного поиска на основе семантических метаописаний // Управление большими системами. — 2014. — Вып. 41. — С. 51—92.
17. *The Open Group.* Service-Oriented Architecture Ontology. Draft Technical Standard. — 2008. — P. 1—112.
18. *Czarnecki K., Helsen S.* Feature-Based Survey of Model Transformation Approaches // IBM Systems Journal. — 2006. — № 45(3). — P. 621—645.
19. *Kalyanpur A., Pastor D., Battle S., Padget J.* Automatic mapping of OWL ontologies into Java. — Режим доступа: <https://pdfs.semanticscholar.org/5af1/38779ab343a802aa29e93ca96d347f393f7f.pdf>
20. *The OMG Ontology Definition Metamodel.* — Режим доступа: <http://www.omg.org/spec/ODM/1.0/>

A.A.Petrenko, A.I.Petrenko

## SEMANTIC MODEL-DRIVEN SERVICE SYSTEMS ARCHITECTURE MODELING BASED ON DOMAIN ONTOLOGIES

The methodology of service systems modeling with the modern service-oriented architecture (SOA) and its modifications is considered. It is based on the sharing Ontology-Driven Development (ODD) and Model-Driven Development (MDD) approaches to business process architecture modeling. The central issue in this case is the interaction and integration of different models based on the use of business processes semantic ontologies. The implementation of the semantic model-driven approach to the development of service-oriented software systems is also shown.

*Keywords:* service-oriented architecture, semantic services, model-driven development, ontology, ontology driven development, business-model, universal language of modeling UML.

### REFERENCES

1. Handbook of services. Service science: Research and innovations in the service economy (2010), Maglio, P., Kieliszewski, C., Spohrer, J. (Eds), Springer, New York, USA.
2. Service systems implementation. Service science: Research and innovations in the service economy (2011), Demirkan, H., Spohrer, J., Krishna, V. (Eds), Springer, New York, USA.
3. Services science: Fundamentals, challenges and future developments (2007), Stauss, B., Engelmann, K. Kremer, A., Luhn, A. (Eds), Springer, Berlin, Germany.
4. Petrenko, A.A. (2015), "Objects and methods of science services", *Systemni doslidzhennya ta informatsiyini tekhnologii*, no. 2, pp. 75-82.
5. Kiselev, G. and Petrenko, O.O. (2015), "The science of services, management and engineering as the basis of innovation activity", *Visnyk Universitetu "Ukraina", Seria Informatsiya, kompyuterizatsiya i kibernetika*, no. 2(18), pp. 28-36.
6. Worldwide Web Consortium (2006), *Web Services Architecture*, available at: <http://www.w3.org/TR/ws-arch>
7. Onto-ServSys : a service system ontology (2011), Mora, M., Raisinghani, M., Gelman, O. and Sicilia, M.A. (Contribs), available at: <https://www.econbiz.de/Record/onto-servsys-a-service-system-ontology-mora-manuel/10009127309>
8. Business model generation: a handbook for visionaries, game changers and challengers (2010), Osterwalder, A. and Pigneur, I. (Eds), Willey, New Jersey, USA.
9. Petrenko, A.I. (2014), Mobile health applications to support the diabetic patient and the doctor, *Proceedings of Design & Test Symposium (EWDTS)*, 2014 East-West, pp. 1-6, DOI: 10.1109/EWDTS.2014.7027105
10. Petrenko, A.A. (2015), "Comparing the types of service systems architectures", *Systemni doslidzhennya ta informatsiyini tekhnologii*, no. 4, pp. 48-62.
11. Lüftenegger, E., Comizzi, M., Grefen, P. and Weisleder, C. (2013), The service dominant business model: A service focused conceptualization, BETA publicatie, WP 402, NUR982, Eindhoven, the Netherlands.
12. Alahmari, S. (2010), A model-driven architecture approach to the efficient identification of services on service-oriented enterprise architecture, Alahmari, S., De Roure, D.Z. (Eds), *Proceedings of the Second Workshop on Service Oriented Enterprise Architecture for Enterprise Engineering in Conjunction with the 14th IEEE International Enterprise Distributed Object Computing Conference*, 2010, Vityria, Brazil.
13. Pahl, C. (2008), Semantic model-driven architecting of service-based software systems, available at: <http://ceur-ws.org/Vol-244/paper3.pdf>

14. Pahl, C. (2006), An ontology for software component matching, *International Journal on Software Tools for Technology Transfer*, Special Edition on Component-based Systems Engineering, no. 7, pp. 1-10.
15. Petrenko, I.A. and Petrenko, O.O. (2015), “Automated methods of search and discovery needed services”, *Visnyk Universitetu “Ukraina”, Serija Informatsiya, kompyuterizatsiya i kibernetika*, no. 1(17), pp. 55-64.
16. Nguyen Ba Ngok and Tuzovsky, A.F. (2014), “Information search model based on semantic metadescriptions”, *Upravleniye bolshimi sistemami*, Iss. 41, pp. 51-92.
17. The open group (2008), Service-oriented architecture ontology. Draft Technical Standard, 208, pp. 1-112.
18. Czarnecki, K. and Helsen, S. (2006), Feature-based survey of model transformation approaches, *IBM Systems Journal*, no. 45(3), pp. 621-645.
19. Kalyanpur, A., Pastor, D., Battle, S. and Padget, J. (2004), Automatic mapping of OWL ontologies into Java, available at: <https://pdfs.semanticscholar.org/5af1/38779ab343a802a29e93ca96d347f393f7f.pdf>
20. The OMG ontology definition metamodel, available at: [http://www.lirmm.fr/~mougenot/Enseignement/FMIN321/OMG\\_ODM\\_Metamodel.pdf](http://www.lirmm.fr/~mougenot/Enseignement/FMIN321/OMG_ODM_Metamodel.pdf)

Поступила 05.05.16

*ПЕТРЕНКО Алексей Алексеевич, аспирант Учебно-научного комплекса «Ин-т прикладного системного анализа» Национального технического университета Украины «Киевский политехнический ин-т» МОН и НАН Украины, который окончил в 2014 г. Область научных исследований — сервис-ориентированные архитектуры, их моделирование и реализация.*

*ПЕТРЕНКО Анатолий Иванович, д-р техн. наук, профессор, зав. отделом информационных ресурсов и кафедрой системного проектирования Учебно-научного комплекса «Ин-т прикладного системного анализа» Национального технического университета Украины «Киевский политехнический ин-т» МОН и НАН Украины. В 1957 г. окончил Киевский политехнический ин-т. Область научных исследований — сервис-ориентированные вычисления для решения практических задач междисциплинарного характера в интеллектуальной среде; организация распределенных программно-технических сетевых комплексов коллективного проектирования на базе грид/облачных технологий.*

