

СТРУКТУРА И СТЕРЕОТИПЫ UML-ПРОФИЛЯ ДЛЯ МОДЕЛИРОВАНИЯ БИЗНЕСА

Ключевая проблема, возникающая при разработке программного обеспечения, связана с определением требований. Бизнес-моделирование дает возможность понять, где приложения программного обеспечения могут улучшить производительность организации, и помогает определить требования к этим приложениям. С этой целью определяется UML-профиль для моделирования бизнеса, в основе которого лежит системологический подход «Узел – Функция – Объект». Одной из ключевых задач при разработке профиля является определение его структуры и стереотипов – главная цель данной статьи. Профиль позволяет создавать UML-модели бизнес-процессов и обеспечить простоту описания зависимостей между ними и UML-моделями информационной программной системы.

Ключевые слова: инженерия требований, метод анализа требований, модель бизнес-процесса, модель спецификации системы, унифицированный язык моделирования UML, системологический подход «Узел – Функция – Объект», триединая парадигма континуума, UML-профиль.

Введение

Постановка проблемы. Исследовательская группа Стендиша в 1994 году проводила опрос среди американских компаний разработчиков программного обеспечения и правительственных учреждений о наиболее значительных факторах, повлиявших на «проблемные» и «потерпевшие неудачу» программные проекты [1]. В отчете о результатах исследования указано три наиболее часто встречающихся ключевых фактора, которые создают «проблемы» в проектах:

- недостаток исходной информации от клиента: 13% всех проектов;
- неполные требования и спецификации: 12% всех проектов;
- изменение требований и спецификаций: 12% всех проектов.

Таким образом, более трети всех проблем, возникающих при разработке программного обеспечения, связаны с инженерией требований, то есть возникают в процессе выявления и анализа требований, документирования результатов наблюдений в различных форматах представления и проверки точности полученного понимания.

Другие источники приводят даже более впечатляющие результаты. Например, организация ESPITI (European Software Process Improvement Training Initiative – Европейская инициатива по обучению совершенствованию процесса программирования) провела в 1995 году опрос с целью определить относительную важность различных типов существующих проблем в отрасли разработки программных систем [1]. Результаты этого широкого опроса, в котором приняли участие 3800 респондентов, показали, что двумя самыми главными проблемами, упоминавшимися почти в половине ответов, оказались:

- спецификации требований;
- управление требованиями клиентов.

В проведенных в компаниях GTE, TRW, IBM и HP исследованиях была оценена стоимость ошибок, возникающих на различных стадиях жизненного цикла проекта [1]. Результаты следующие: если стоимость усилий, необходимых для обнаружения и устранения ошибки на стадии написания кода, принять за единицу, то стоимость выявления и устранения ошибки на стадии выработки требований будет в пять – десять раз меньше. А стоимость обнаружения и устранения ошибки на стадии сопровождения – в 20 раз больше.

В исследовании Снайдера (Snyder) из компании Hughes Aircraft в 1992 году было обнаружено явление «просачивания» дефектов на следующие стадии разработки проектов [1]. В том числе и ошибок в определении требований вплоть до этапа сопровождения, когда стоимость устранения этих ошибок уже в 200 раз выше, чем в начале разработки. Учитывая эффект умножения стоимостей устранения и частоту возникновения ошибок в определении требований исследователи Бом (Boehm) и Папаччо (Parrasio) подсчитали, что на устранение таких ошибок может быть истратчено 25 – 40 % всего бюджета проекта [1].

Таким образом, ошибки в определении требований являются одной из *основных причин* возникновения проблем при разработке программного обеспечения, а *стоимость их устранения наиболее высокая* и значительно может повлиять на бюджет всего проекта разработки программного продукта.

Различают два довольно различных аспекта инженерии требований: выявление (формулировка) необходимых требований и их анализ [2]. Оба процесса оказывают значительное влияние на появление вышеупомянутых проблем требований, но в данном исследовании уделяется внимание именно методам анализа требований.

Эволюция методов анализа требований (табл. 1) началась с попыток распространить фор-

мальные теории на объектно-ориентированный анализ [2]. Использование формальных и полужформальных методов для анализа требований показало их несостоятельность, так как они не подразумевали процесс получения знаний и не гарантировали, что спецификация системы соответствует истинным требованиям. В противоположность формальным методам, в таких подходах, как ETHICS, особое значение придавалось социальным и техническим аспектам системы. При этом ETHICS не интегрирует идеи перестройки бизнес-процессов, быстрой разработки или объектной технологии. Появление социально-ориентированных методов в значительной степени улучшило ситуацию в отрасли, так как их

основной особенностью было привлечение пользователей к определению технических требований и проектированию. Несмотря на это они не разграничивали модель мира и модель системы (системную спецификацию). Метод ORCA представляет одну из попыток разграничить эти две модели, но не обеспечивает формальную связь между его двумя языками моделирования. Этот недостаток устранен в таких методах как SSM и SOMA. Однако метод SSM не обеспечивает доказательство, что системная спецификация служит представлением требований. А метод SOMA не имеет формального синтаксиса, и не гарантирует, что требования адекватны потребностям пользователей.

Таблица 1

Эволюция методов анализа требований

Название метода	Основные особенности и преимущества метода	Основные недостатки метода
1) Формальные и полужформальные методы (JSD, VDM, Z, Object Z, Catalysis, Syntropy, метод проблемных фреймов)	Позволяют проводить соответствие кода спецификации системы, что может быть оправдано для систем безопасности и ключевых подсистем	Настоящее требование может быть проигнорировано как не формальное Нет гарантии, что спецификация соответствует истинным требованиям Все методы относятся к анализу уже известных и понятых требований и не подразумевают процесс получения знаний
2) ETHICS (Effective Technical and Human Implementation of Computer-based Systems – эффективная техническая и человеческая реализация компьютерных систем)	Предоставляет современные средства перепроектирования бизнес-процессов и анализа последовательности выполняемых работ и меньше связан со строгими бюрократическими системами	Не дает никаких рекомендаций по моделированию Не интегрирует идеи перестройки бизнес-процессов, быстрой разработки или объектной технологии
3) Социально-ориентированные методы (методы проектирования, направленные на пользователя и задачу; метод контекстных запросов; проектирование, ориентированное на использование; метод функции качества (дом качества); метод CREWS; метод анализа последовательности выполняемых действий)	Привлечение пользователей к определению технических требований и проектированию Создание системы не приведет к деградации или снижению профессионализма пользователей	Возникают случаи, ведущие к ухудшению или невозможности совместной работы мужчин и женщин Не разграничивают модель мира и модель спецификации системы
4) ORCA (Object-Oriented Requirements Capture and Analysis – объектно-ориентированный сбор требований и анализ)	Основан на разграничении целевых и поведенческих сущностей и обеспечивает две отдельные системы обозначений для представления каждого аспекта модели. Ориентирован на перепроектирование бизнес-процессов и среду внедрения системы	Неспособность ORCA обеспечить формальную связь между его двумя языками моделирования Не позволяет, несмотря на строгость, доказать, что модель системы отвечает требованиям, установленным в модели мира
5) SSM (Soft Systems Method – метод программных систем)	Подход обычно начинается с создания так называемой «детальной картины мира», или модели бизнес-процесса, которая разграничивается с моделью спецификации системы	Не обеспечено доказательство, что системная спецификация служит представлением требований
6) SOMA (Semantic Object Modeling Approach – семантический подход к объектному моделированию)	Обеспечивает построение «модели мира» в форме моделей бизнес-процессов и задачи, «модель системы» – в виде модели спецификации Catalysis. Предоставляет доказательство, что системная спецификация служит представлением требований	Не имеет формального синтаксиса Не гарантирует, что требования адекватны потребностям пользователей

Таким образом, современный метод анализа требований должен обеспечивать: современные средства перепроектирования бизнес-процессов; привлечение пользователей к определению технических требований и проектированию бизнес-процессов; раздельное построение «модели мира» (модели бизнес-процессов) и «модель системы» (модель спецификации системы); различный формаль-

ный синтаксис для модели бизнес-процессов и модели спецификации системы; доказательство, что системная спецификация служит представлением требований; доказательство, что требования адекватны потребностям пользователей.

Таким образом, *общий подход* к анализу требований заключается в раздельном, но согласованном между собой, моделировании бизнес-процессов и

спецификации системы с привлечением к этому процессу пользователя (рис. 1).

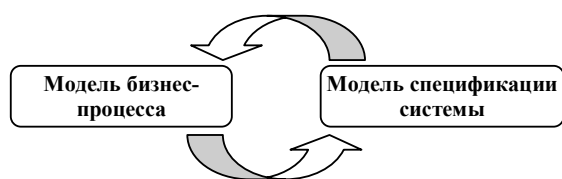


Рис. 1. Общий подход к анализу требований

Это объясняется тем, что в среде информационных систем и технологий, изобилующей сложностями бизнес-процесса, прежде чем попытаться определить конкретные проблемы, достойные решения, необходимо понять некоторые моменты самого бизнес-процесса. Кроме того, даже при создании отдельного приложения необходимо помнить о многогранности среды, в которой оно будет функционировать. Бизнес-моделирование дает возможность понять, где информационные системы могут улучшить производительность данной организации, и помогает определить требования к этим системам [1]. А формальная связь между моделями мира и системы служит доказательством, что системная спецификация служит представлением требований пользователей и заинтересованных лиц.

Для моделирования бизнес-процессов, безусловно, можно применить множество методов. Поскольку методы объектно-ориентированного визуального моделирования, которые объединены в унифицированном языке моделирования UML [3], применяются практически во всех новых проектах разработки программных систем для описания системной спецификации, естественно попытаться использовать аналогичные методы в области моделирования бизнес-процессов. Это обеспечит доступность метода широкому кругу не специалистов, а это очень важно в данном случае, так как бизнес-модель должна быть понятна для многих заинтересованных в информационной системе лиц и ее пользователей.

Одно из преимуществ данного подхода к моделированию бизнес-процесса состоит в простоте описания зависимостей между моделями бизнес-процесса и информационной системы. Это повышает производительность процесса разработки системы (автоматизированный переход между моделями и их согласование на основе концепции MDD), а также помогает удостовериться в том, что разрабатываемая информационная система решает реальные бизнес-потребности (бизнес-ориентированный подход к ИТ).

Важно также отметить, что в UML есть возможности специализации языка для конкретных предметных областей в виде UML-профилей [3], то есть и для бизнес-моделирования. Это дает возможность технологически реализовать предложенный подход к решению задачи.

Обычной практикой является применение метода прецедентов для моделирования бизнес-процессов, но такой подход вызывает ряд проблем [2]. Преце-

денты, несомненно, полезны для определения систем, но не позволяют полностью описать требования. Прецеденты связывают исполнителей, которые представляют роли пользователей в системе, с самой системой. С другой стороны, требования могут выдвигать люди и организации, которые никогда не окажутся вблизи системы. Игнорирование взаимодействия с не пользователями может привести к потере важных технических возможностей. То есть, *необходим особый объектно-ориентированный метод моделирования бизнес-процессов*, совместимый с моделями прецедентов из системной спецификации.

Анализ последних исследований и публикаций. Описанная задача и пути ее решения уже рассматривались ранее автором в работах [4, 5]. В ходе ее исследования были получены результаты, свидетельствующие о том, что целесообразно представлять модели бизнес-процесса непосредственно средствами разработки программных моделей – на языке UML, что можно реализовать в виде UML-профиля для моделирования бизнеса. В таком случае описание зависимостей между моделями бизнеса и программной системы значительно проще и можно обеспечить их совместимость, согласованность и автоматизированное преобразование.

В работе [6] исследуемая задача рассматривается в контексте проблемы контроля инвестиций в информационные системы и технологии, для решения которой предлагается использовать бизнес-ориентированный подход. Реализацию бизнес-ориентированного подхода к информационным технологиям предлагается обеспечить на основе использования системологического подхода «Узел – Функция – Объект» (УФО-подход) в качестве метода моделирования бизнеса, который может создать контекст для разработки программных моделей на унифицированном языке моделирования UML.

Выбор средств решения рассматриваемой задачи и его обоснование – одна из главных задач всего исследования – представлено в работе [7] в виде математической модели согласования объектно-ориентированной методологии и системологического подхода к моделированию бизнеса.

Целью исследования является создание метода анализа требований, который обеспечивает раздельное, но совместимое и согласованное между собой, объектно-ориентированное моделирование бизнес-процессов и спецификации программной информационной системы на языке UML с привлечением к этому процессу пользователей и заинтересованных лиц.

Объектом исследования является процесс разработки требований с использованием процесса анализа проблемы, документирования результатов наблюдений в различных форматах представления и проверки точности полученного понимания. В пределах объекта можно выделить объектно-ориентированные метод, модели и технологию анализа требований к информационным системам, что является **предметом данного исследования**.

Для достижения поставленной цели необходимо решить определенные ниже задачи:

- выбрать и обеспечить использование в качестве методологического основания для моделирования бизнес-процессов, строгой и адекватной теории объектно-ориентированного анализа и проектирования бизнес-процессов;

- разработать в виде UML-профиля модель бизнес-процессов и ее формальный синтаксис, такой, что будет обеспечивать ее совместимость с UML-моделью спецификации системы – задача, частичное решение которой является целью статьи;

- разработать метод анализа требований, в основе которого лежит согласованность UML-моделей бизнес-процессов и спецификации системы, обеспечивающая строгое доказательство того, что системная спецификация служит представлением требований, адекватных потребностям пользователей и заинтересованных лиц;

- разработать и реализовать технологические, алгоритмические и программные средства поддержки процесса анализа требований.

Визуальную объектно-ориентированную модель можно представить в виде графа или одномерного комплекса Γ [8], который состоит из множества вершин $V(\Gamma) \neq \emptyset$ – понятий модели, ребер $E(\Gamma)$ – концептуальных отношений между понятиями модели, пары отображений $\alpha, \omega: E(\Gamma) \rightarrow V(\Gamma)$, где $\alpha(e)$ – начало ребра e , $\omega(e)$ – конец ребра e .

Если рассматривать визуальные объектно-ориентированные модели бизнес-процессов и спецификации системы в виде графов, то вторая и третья сформулированные выше задачи приобретают формальное представление.

Пусть граф Γ_1 – оригинальная модель бизнес-процессов, Γ_2 – модель бизнес-процессов в формальном синтаксисе языка UML, Γ_3 – модель спецификации системы. Тогда, задача разработки в виде UML-профиля модели бизнес-процессов и ее формального синтаксиса будет представлять собой построение отображения $f^{1,2}: \Gamma_1 \rightarrow \Gamma_2$ графа Γ_1 в граф Γ_2 , которое состоит из согласованных отображений $f^{1,2}_V: V(\Gamma_1) \rightarrow V(\Gamma_2)$, $f^{1,2}_E: E(\Gamma_1) \rightarrow E(\Gamma_2)$, то есть:

$$f^{1,2}_V(\alpha_1(e)) = \alpha_2(f^{1,2}_E(e));$$

$$f^{1,2}_V(\omega_1(e)) = \omega_2(f^{1,2}_E(e)).$$

Соответственно задача согласования моделей бизнес-процессов и спецификации системы будет представлять собой построение взаимно однозначного отображения (изоморфизма) $f^{2,3}: \Gamma_2 \leftrightarrow \Gamma_3$ графа Γ_2 и графа Γ_3 , которое состоит из согласованных взаимно однозначных отображений $f^{2,3}_V: V(\Gamma_2) \leftrightarrow V(\Gamma_3)$, $f^{2,3}_E: E(\Gamma_2) \leftrightarrow E(\Gamma_3)$, то есть

$$f^{2,3}_V(\alpha_2(e)) = \alpha_3(f^{2,3}_E(e));$$

$$f^{2,3}_V(\omega_2(e)) = \omega_3(f^{2,3}_E(e)).$$

Выбор средств решения задачи

Для достижения цели исследования, как указывалось выше, необходимо решить ряд задач, первая из которых заключается в обеспечении использования в качестве методологического основания для моделирования бизнес-процессов, строгой и адекватной теории объектно-ориентированного анализа и проектирования бизнес-процессов. Решение этой задачи автором рассматривалось в трудах [6, 7], где предложен обоснованный выбор в качестве такого методологического основания моделирования бизнес-процессов системологии и системологического подхода «Узел – Функция – Объект» [9].

Задачу разработки в виде UML-профиля модели бизнес-процессов и ее формального синтаксиса можно разбить на несколько этапов:

- определение структуры и стереотипов профиля – объект внимания данной статьи;

- разработка концептуальной модели профиля;

- формализация внутренней согласованности элементов модели профиля.

Для разработки UML-профиля для моделирования бизнеса предлагается использовать средства триединой парадигмы континуума [10], математический аппарат теории полугрупп и графов [8].

Рассмотрим последовательно решение задачи определения структуры и стереотипов профиля и выбор указанных средств для ее разрешения.

Выбор и обоснование структуры UML-профиля для моделирования бизнеса

В модели произвольной бизнес-системы (организации), как правило, необходимо показать организационную и функциональную структуру, материальные и информационные потоки моделируемой организации [11]. UML-профиль для моделирования бизнеса должен обеспечивать представление всех указанных аспектов бизнес-системы для обеспечения полноты модели. Таким образом, первоочередной задачей при разработке профиля является определение набора представлений профиля и идентификация представлений языка UML, на основе которых можно промоделировать все аспекты бизнес-системы, для чего необходимо рассмотреть логическую структуру UML. Логическая структура языка UML определена его метамоделью [12] таким образом, что элементы каждого из логических представлений языка собраны в одноименных пакетах верхнего уровня. Анализируя пакеты верхнего уровня UML [12], целесообразно выделить среди них «Прецеденты (UseCases)», так как разработка произвольной модели на UML, как правило, начинается с ключевого представления прецедентов. Оно описывает функциональное назначение системы в самом общем виде с точки зрения всех ее пользователей и заинтересованных лиц [3]. Таким образом, модель прецедентов может служить базовым представлением функциональной структуры бизнес-системы.

Для того чтобы раскрыть внутреннее представление бизнес-процессов организации целесообразно строить диаграммы деятельности UML, так как представление деятельности традиционно рекомендуется использовать для этой цели, как создателями UML [3], так и рационального унифицированного процесса RUP [13]. Элементы данного представления объединены в пакет с одноименным названием – «Деятельности (Activities)», среди пакетов верхнего уровня метамодели языка UML [12]. Таким образом, модель деятельности, как один из аспектов поведения системы, может обеспечить в первую очередь представление функциональной структуры бизнес-системы.

Для отображения структурного аспекта системы в UML используются несколько представлений, но базовым среди них традиционно является статическое представление классов [3]. Диаграммы классов целесообразно использовать также и для представления структуры бизнес-системы. Элементы данного представления собраны в пакете «Классы (Classes)» [12] – пакете верхнего уровня метамодели языка UML.

В итоге можно констатировать, что модель произвольной бизнес-системы, а именно ее организационную и функциональную структуру, материальные и информационные потоки, можно отобразить на представлениях прецедентов, деятельности и классов языка UML. Причем функциональный аспект организации в большей степени представлен диаграммами прецедентов и деятельности, а структурный – диаграммами классов. Материальные и информационные потоки организации проявляются повсеместно на всех вышеуказанных представлениях.

Целесообразно поименовать идентифицированные представления (пакеты верхнего уровня) UML-профиля для моделирования бизнеса в соответствии с контекстом объекта моделирования и принятой нотацией UML следующим образом: модель бизнес-прецедентов (BusinessUseCaseModel), модель бизнес-деятельности (BusinessActivityModel), модель предметной области бизнеса (BusinessDomainModel). В таком случае структура профиля приобретает вид, представленный на рис. 2.

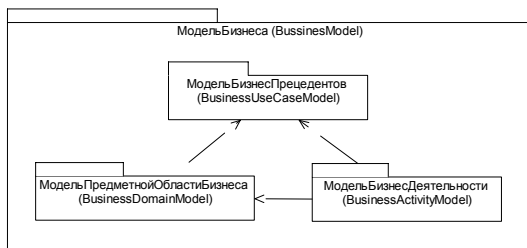


Рис. 2. Пакеты верхнего уровня UML-профиля для моделирования бизнеса

Модель бизнес-прецедентов является основной отправной точкой для моделирования бизнес-процессов (функциональная структура бизнес-системы). Бизнес-единицы, обеспечивающие выполнение функций системы, представляются на модели предметной области бизнеса (организационная структура

бизнес-системы). Детали бизнес-процессов в виде отдельных бизнес-операций и ответственные за их выполнение бизнес-единицы представляются на модели бизнес-деятельности.

Идентификация UML-метаклассов для определения стереотипов UML-профиля моделирования бизнеса

Использование UFO-подхода в качестве основы UML-профиля для моделирования бизнес-процессов предполагает отображение системологических понятий и правил построения модели бизнес-процесса в представлениях профиля. Рассмотрим задачу отображения понятий. Задача представляет собой идентификацию UML-метаклассов соответствующих понятиям UFO-подхода и обеспечивающих определение стереотипов UML-профиля.

Решение задачи можно осуществить за счет введения общего формального основания. Таковым может служить триединая парадигма континуума (ТПК) [10], которая предоставляет согласованную структуру метамодели и необходимый и достаточный набор базовых понятий моделирования («Точка в пространстве», «Точка во времени», «Временной интервал», «Пространственный интервал», «Объект», «Среда», «Состояние», «Действие»). Таким образом, для выявления семантического соответствия между понятиями UFO-подхода и метаклассами UML необходимо их представить в метамодели ТПК.

Как указано в [10] каждое понятие любой объектно-ориентированной структуры моделирования является либо одним из базовых понятий моделирования, либо понятием спецификации ТПК, то есть их специализацией. К основным понятиям системологического подхода «Узел – Функция – Объект» [9] можно отнести: модель бизнеса (Business Model), UFO-элемент (UFO-element), узел (Unit), функция (Function), объект (Object), связь (Link). Проанализируем UFO-элемент – основное понятие UFO-подхода, и его составляющие с целью представления системологических понятий в метамодели ТПК.

«UFO-элемент – это представление любой системы в виде трехэлементной конструкции, то есть одновременно в виде «Узла», «Функции» и «Объекта». «Узел» – это представление системы как структурного элемента надсистемы в виде перекрестка связей с другими системами» [9]. Как видно из определения «Узел» является структурным элементом, т.е. непространственно-временным понятием, а именно понятием «Континуума конституции модели» в терминах ТПК (рис. 3). В ТПК введено два базовых понятия моделирования «Континуума конституции модели»: «Объект» и «Среда» [10]. Очевидно, что «Узел» не является «Объектом» ТПК или его специализацией, так как «Узел» задает условия для существования конкретного «Объекта» за счет введения конкретных входных и выходных системных связей – «перекрестка связей с другими системами» [9].

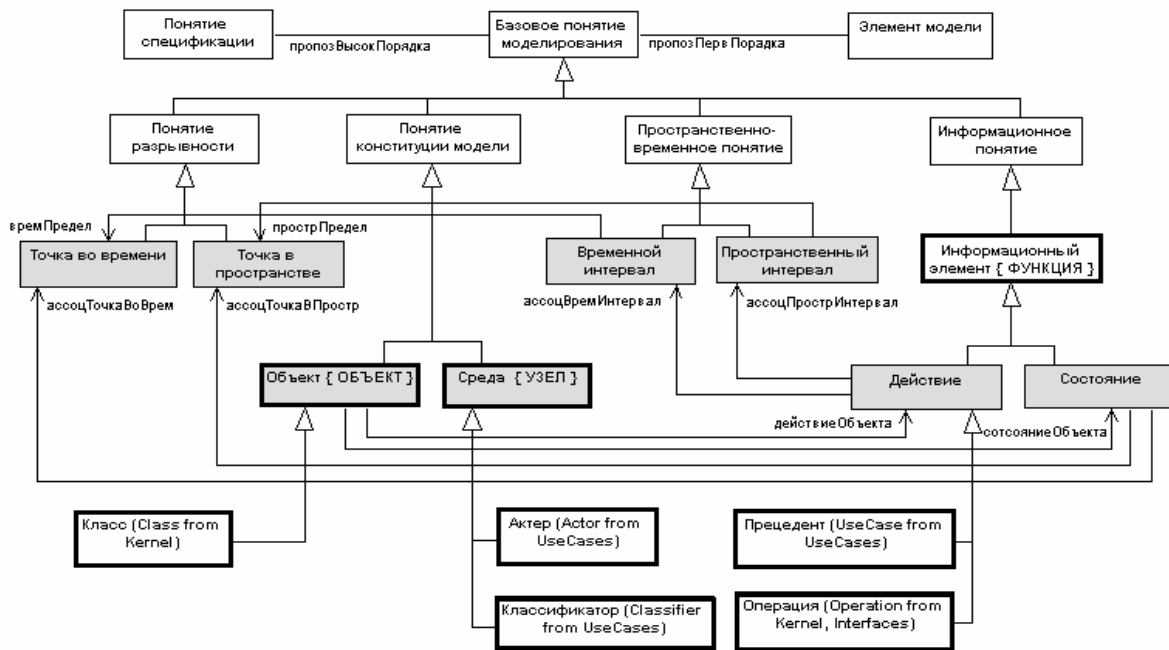


Рис. 3. Фрагмент представления системологических понятий и метаклассов UML в структуре триединой парадигмы континуума

Таким образом, «Узел» может соответствовать только «Среде» ТПК или ее специализации. Действительно семантика «Узла» хорошо согласуется с понятием «Среды» в ТПК, которая представляет собой границы существования соответствующего «Объекта» [10]. «Узел» специализирует понятие «Среды» ТПК за счет введения конкретных входных и выходных системных связей, то есть соответствует понятию спецификации ТПК, специализирующему «Среду» – базовое понятие моделирования ТПК.

Теперь рассмотрим «Функцию», вторую составляющую UFO-элемента. «Функция» – это представление системы как функционального элемента, выполняющего определенную роль с точки зрения поддержания надсистемы путем балансирования соответствующего узла». Важно также отметить, что в UFO-элементе «Функцию» реализует соответствующий «Объект» данного «Узла» [9].

Как видно из определения «Функция» в терминах ТПК представляет собой некую пространственно-временную информацию об не пространственно-временном элементе. Это очевидно из семантики понятия функционального элемента, который представляет динамику поведения некоторого структурного элемента. Как видно из второй части указанного определения этим структурным элементом является «Объект». Это полностью соответствует определению «Информационного элемента» в ТПК [10].

Таким образом, «Функция» соответствует «Информационному элементу» – базовое понятие моделирования ТПК. Важно отметить, что нет возможности специализировать «Функцию», так как в UFO-подходе формально не введены пространственно-временные понятия «Точки в пространство-времени» и «Интервала в пространство-времени». ТПК дает нам возможность формально определить

эти понятия. После чего можно говорить о значении «Функции» в «Точке в пространство-времени» и на «Интервале в пространство-времени», что соответствует базовым понятиям моделирования ТПК «Состояние» и «Действие» соответственно.

Рассмотрим «Объект» – последнюю составляющую UFO-элемента. «Объект» – это представление системы как субстанциального элемента, реализующего данную функцию в виде некоторого материального образования, обладающего конструктивными, эксплуатационными и т.д. характеристиками» [9]. В ТПК существует аналогичное понятие «Объекта», как конституционного элемента. Следовательно, «Объект» UFO-элемента соответствует «Объекту» ТПК – базовому понятию моделирования в ТПК.

Аналогичным образом можно на основании ТПК проанализировать и другие понятия UFO-подхода. С другой стороны метаклассы UML также можно соотнести с базовыми понятиями моделирования ТПК. Тогда можно говорить о семантическом соответствии UFO-понятий и UML-метаклассов, которые соотнесены с одними и теми же базовыми понятиями моделирования ТПК. Фрагмент поиска семантических соответствий UFO-понятий и UML-метаклассов модели бизнес-прецедентов представлен на рис. 3. Серым цветом на рисунке закрашены классы базовых понятий моделирования ТПК, жирной черной рамкой обведены классы UFO-понятий и UML-метаклассы.

Результаты сопоставления UFO-понятий и метаклассов UML на основе ТПК с определением стереотипов моделей бизнес-прецедентов, бизнес-деятельности, предметной области бизнеса UML-профиля для моделирования бизнеса представлены соответственно в табл. 2 – 4.

Выводы

В статье рассмотрена актуальная проблема определения требований к программным системам. Основной задачей является анализ требований к бизнес-моделированию, которое дает возможность понять, где приложения программного обеспечения могут улучшить производительность организации, и помогает определить требования к этим приложениям. С этой целью определяется UML-профиль

для моделирования бизнеса и его формальный синтаксис, в основе которого лежит системологический подход «Узел – Функция – Объект».

Задачу разработки UML-профиля для моделирования бизнеса и его формального синтаксиса можно разбить на несколько этапов. Одной из ключевых задач при разработке профиля является определение его структуры и стереотипов. В данной статье было представлено решение этой задачи.

Таблица 2

Модель бизнес-прецедентов

УФО-понятие		Понятие / метакласс UML (UML Metaclass)	Стереотип UML (UML Stereotype)
УФО-модель (UFO-Model)		Модель (Model from Models)	Модель бизнес-прецедентов (BusinessUseCaseModel <i>Attribute</i> BusinessUseCaseModel :: viewpoint = "BusinessUseCaseView")
УФО-элемент (UFO-Element)	Узел (Unit)	1) Классификатор (Classifier from UseCases) 2) Актер (Actor from UseCases)	1) Бизнес-система (BusinessSystem) 2) Бизнес-актер (BusinessActor)
	Функция (Function)	1) Прецедент (UseCase from UseCases) 2) Операция (Operation from Kernel, Interfaces)	1) Бизнес-прецедент (BusinessUseCase) 2) Потребность бизнеса (BusinessRequirement)
	Объект (Object)	1) Класс (Class from Kernel) 2) Класс (Class from Kernel)	1) Исполнитель бизнеса (BusinessWorker) 2) Заказчик бизнеса (BusinessCustomer)
Связь (Link)		Ассоциация-класс (AssociationClass from AssociationClasses)	Бизнес-сущность (BusinessEntity)

Таблица 3

Модель бизнес-деятельности

УФО-понятие		Понятие / метакласс UML (UML Metaclass)	Стереотип UML (UML Stereotype)
УФО-модель (UFO-Model)		Модель (Model from Models)	Модель бизнес-деятельности (BusinessActivityModel <i>Attribute</i> BusinessActivityModel :: viewpoint = "BusinessActivityView")
УФО-элемент (UFO-Element)	Узел (Unit)	Деятельность (Activity from BasicActivities, CompleteActivities, FundamentalActivities, StructuredActivities)	Бизнес-деятельность (BusinessActivity)
	Функция (Function)	Действие (Action from CompleteActivities, FundamentalActivities, StructuredActivities)	Бизнес-действие (BusinessAction)
	Объект (Object)	Класс (Class from Kernel) Разбиение деятельности (ActivityPartition from IntermediateActivities)	Исполнитель бизнес-действия (BusinessActionWorker)
Связь (Link)		Поток объектов (ObjectFlow from BasicActivities, CompleteActivities) Узел объекта (ObjectNode from BasicActivities, CompleteActivities)	Поток бизнес-сущностей (BusinessEntityFlow)

Таблица 4

Модель предметной области бизнеса

УФО-понятие		Понятие / метакласс UML (UML Metaclass)	Стереотип UML (UML Stereotype)
УФО-модель (UFO-Model)		Модель (Model from Models)	Модель предметной области бизнеса (BusinessDomainModel <i>Attribute</i> BusinessDomainModel :: viewpoint = "BusinessDomainView")
УФО-элемент (UFO-Element)	Узел (Unit)	Пакет (Package from Kernel) Бизнес-система (BusinessSystem from BusinessUseCaseModel)	Бизнес-система (BusinessSystem)
	Функция (Function)	Операция (Operation from Kernel, Interfaces)	Операция исполнителя бизнеса (BusinessWorkerOperation)
	Объект (Object)	Класс (Class from Kernel)	Исполнитель бизнеса (BusinessWorker)
Связь (Link)		Класс (Class from Kernel)	Бизнес-сущность (BusinessEntity)

Для определения профиля были определены системологические понятия модели бизнес-процесса. На основе средств, предоставляемых триединой парадигмой континуума, найдены для системологических понятий семантические аналоги в метамодели UML – метаклассы языка, на основе которых введены стереотипы профиля.

Таким образом, формальный синтаксис UML-профиля для моделирования бизнеса имеет твердые методологические основания в виде обоснованного и согласованного набора понятий и концепций бизнес-моделирования, определенного в системологии и системологическом подходе «Узел – Функция – Объект», при этом сохраняет привычную нотацию языка UML.

Это должно позволить создавать с помощью профиля строгие, адекватные потребностям бизнеса и согласованные модели бизнес-процессов и обеспечить простоту описания зависимостей между ними и UML-моделями программной системы, что существенно для достижения главной цели исследования – создания метода анализа требований.

Список литературы

1. Леффингуэлл Д, Уидриг Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Пер. с англ. и ред. Н.А. Ореховой. – М.: Издательский дом «Вильямс», 2002. – 448 с.
2. Грэхем И. Объектно-ориентированные методы. Принципы и практика. – 3-е издание / Пер. с англ. С.В. Белковой и др.; Под ред. Н.Н. Кукуль. – М.: Издательский дом «Вильямс», 2004. – 880 с.
3. UML. Классика CS: Пер. с англ. / Г. Буч, А. Якобсон, Дж. Рамбо; С.А. Орлов. – С.-Пб.: Питер, 2006. – 736 с.
4. Українець А.Г. Преобразование диаграммы классов UML в модель UFO-toolkit // Радиоэлектроника и ин-

форматика. – 2004. – № 2. – С. 55-60.

5. Взаимное преобразование UFO- и UML-моделей. / М.Ф. Бондаренко, С.С. Пирог, Е.А. Соловьева, А.Г. Українець // Научный журнал «Проблемы программирования». – 2004. – № 2-3. – С. 150-155.

6. Українець А.Г. Системологический подход к моделированию бизнеса как основа бизнес-ориентированного подхода к информационным технологиям // Радиоэлектроника и информатика. – 2007. – № 4. – С. 142-149.

7. Українець А.Г. Разработка модели согласования объектно-ориентированной методологии и системологического подхода к моделированию бизнеса // Вісник Міжнародного Слов'янського університету. Серія «Технічні науки». – 2007. – Т. X, № 2. – С. 29-37.

8. Общая алгебра / О.В. Мельников, В.Н. Ремесленников, В.А. Романьков и др.; Под общ. ред. Л.А. Скорняков. – М.: Наука, 1990. – Т. 1 – 592 с.

9. Бондаренко М.Ф., Маторин С.И., Соловьева Е.А. Моделирование и проектирование бизнес-систем: методы, стандарты, технологии: Уч. пособ. / Предисл. Э.В. Попова. – Х.: ООО «СМИТ», 2004. – 272 с.

10. Naumenko A. Triune Continuum Paradigm: a paradigm for General System Modelling and its applications for UML and RM-ODP. Ph.D thesis 2581, Swiss Federal Institute of Technology – Lausanne, June 2002.

11. Мамиконов А.Г. Проектирование АСУ: Учебник для спец. "АСУ" вузов. – М.: Высшая школа, 1987. – 303 с.

12. Unified Modeling Language: Superstructure. Version 2.1.1 [Электрон. ресурс]. – Режим доступа: <http://www.omg.org/cgi-bin/apps/doc?formal/07-02-05.pdf>.

13. Крамен, Ф. Введение в Rational Unified Process: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 240 с.

Поступила в редколлегию 21.08.2008

Рецензент: д-р техн. наук, проф. Е.А. Соловьева, Харьковский национальный университет радиоэлектроники, Харьков.

UML-ПРОФІЛЬ ДЛЯ МОДЕЛЮВАННЯ БІЗНЕСУ

О.Г. Українець

Ключова проблема, що виникає при розробці програмного забезпечення, пов'язана з визначенням вимог. Бізнес-моделювання дає можливість зрозуміти, де прикладне програмне забезпечення може поліпшити продуктивність організації, і допомагає визначити вимоги до нього. Із цією метою визначається UML-профіль для моделювання бізнесу, в основі якого лежить системологічний підхід "Вузол – Функция – Об'єкт". Однією із ключових задач при розробці профілю є визначення його структури й стереотипів – головна мета даної статті. Профіль дозволяє створювати UML-моделі бізнес-процесів та забезпечити простоту опису залежностей між ними й UML-моделями інформаційної програмної системи.

Ключові слова: інженерія вимог, метод аналізу вимог, модель бізнес-процесу, модель специфікації системи, уніфікована мова моделювання UML, системологічний підхід "Вузол – Функция – Об'єкт", триєдина парадигма континууму, UML-профіль.

UML PROFILE FOR BUSINESS MODELING

O.G. Ukrainets

Software development key problem connected with requirements definition. Business modeling to give a chance to understand where software applications can improve organization productivity and help to define requirements to these applications. Toward this end is defined UML Profile for business modeling based on the systemological approach "Unit – Function – Object". One of the key tasks of Profile development is definition of its structure and stereotypes – the main goal of the article. Profile allows creating business-process UML-models and providing description simplicity of dependencies between them and UML-models of information software system.

Keywords: Requirements Engineering, Requirements Analysis Method, Business-Process Model, System Specification Model, Unified Modeling Language – UML, Systemological Approach "Unit – Function – Object", Triune Continuum Paradigm, UML Profile.