

ОСОБЕННОСТИ ОТОБРАЖЕНИЯ ОНТОЛОГИЙ ПРЕДМЕТНОЙ ОБЛАСТИ В ОПИСАНИИ ЭЛЕМЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ

В. М. Левыкин, М. В. Евланов, М. А. Керносов, М. Э. Керносова

Харьковский национальный университет радиоэлектроники
просп. Ленина, 14, г. Харьков, 61166, Украина. E-mail: iyc@kture.kharkov.ua

Рассмотрены основные проблемы сокращения затрат времени на разработку современных информационных систем. Выделена задача анализа предметной области и портфеля выполненных проектов на предмет возможности применения полученных ранее решений в ходе создания новой информационной системы. Рассмотрены основные функции информационной технологии, которая позволяет решить эту задачу. Предложен подход к отображению сети фреймов, которая описывает онтологию предметной области, в описания классов программного обеспечения и описания сущностей базы данных информационной системы. Выделены основные этапы формирования сети фреймов, диаграмм классов и схемы данных разрабатываемой информационной системы. Предложен перечень работ, которые следует выполнять на каждом из выделенных этапов. Разработан подход к решению проблемы обеспечения целостности данных в генерируемой схеме данных информационной системы. Предложена модель формирования интерфейсов пользователя разрабатываемой информационной системы.

Ключевые слова: информационная система, онтология, класс, сущность.

ОСОБЛИВОСТІ ВІДОБРАЖЕННЯ ОНТОЛОГІЙ ПРЕДМЕТНОЇ ГАЛУЗІ В ОПИСИ ЕЛЕМЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ

В. М. Левикін, М. В. Євланов, М. А. Керносов, М. Е. Керносова

Харківський національний університет радіоелектроніки
просп. Леніна, 14, г. Харків, 61166, Україна. E-mail: iyc@kture.kharkov.ua

Розглянуто основні проблеми скорочення витрат часу на розробку сучасних інформаційних систем. Виділено задачу аналізу предметної галузі та портфелю виконаних проектів для подальшого вивчення можливості використання отриманих раніше рішень під час створення нової інформаційної системи. Розглянуто основні функції інформаційної технології, яка дозволяє вирішувати зазначену задачу. Запропоновано підхід до відображення мережі фреймів, яка описує онтології предметної галузі, в описи класів програмного забезпечення та описи сутностей бази даних інформаційної системи. Виділено основні етапи формування мережі фреймів, діаграми класів та схеми даних інформаційної системи, що розробляється. Запропоновано перелік робіт, які слід виконувати на кожному із визначених етапів. Розроблено підхід до вирішення проблеми забезпечення цілісності даних в схемі даних інформаційної системи, що генерується. Запропоновано модель формування інтерфейсів користувача інформаційної системи, що розробляється.

Ключові слова: інформаційна система, онтологія, клас, сутність.

АКТУАЛЬНОСТЬ РАБОТЫ. В настоящее время развитие информационных систем (ИС) для автоматизации различных бизнес-процессов (БП) требует решения наукоемких задач. В условиях жесткой конкуренции на ИТ-рынке время, которое тратится на создание ИС, является самым важным ресурсом проектов создания ИС и оказывает существенное влияние на планирование проектов развития ИС.

Одним из наиболее распространенных способов сокращения затрат времени на разработку ИС является повторное использование компонентов, полученных в результате выполнения предыдущих проектов создания ИС. Основным недостатком данного подхода является сложность анализа портфеля выполненных ранее проектов создания ИС на предмет возможности применения компонентов этих ИС при разработке новой системы [1].

Поэтому задача разработки методов, приемов, способов и информационных технологий (ИТ), позволяющих автоматизировать проведение подобного анализа, снизить сложность сопоставления терминов различных предметных областей (ПрО), устранить ошибки при проведении подобного анализа, является актуальной с теоретической и практической точек зрения.

МАТЕРИАЛ И РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ. Основные затраты времени на проведение анализа

портфеля выполненных ранее ИТ-проектов для повторного использования компонентов возникают в ходе формирования и анализа требований к создаваемой ИС. Существующие стандарты вообще не рассматривают проведение такого анализа как одну из работ, образующих типовые процессы создания ИС [2]. Следует отметить, что вопросам представления и анализа требований к программным продуктам и, в частности, требований к ИС посвящены труды целого ряда исследователей [3–5]. Однако подавляющее большинство этих исследований оставляют открытым вопрос применимости опыта, накопленного в ходе выполнения предыдущих ИТ-проектов создания ИС, при формировании и анализе требований к создаваемой ИС. Кроме того, задачи максимизации объема повторного использования готовых компонентов в ходе создания новой ИС и обеспечения максимального соответствия создаваемой ИС специфике конкретного БП объекта автоматизации во многом являются противоречивыми.

Поэтому как с теоретической, так и практической точек зрения актуальной является задача разработки интеллектуальной ИТ быстрого создания ИС, обеспечивающей автоматизацию проведения анализа ПрО и портфеля выполненных проектов на предмет возможности применения полученных ранее решений в ходе создания новой ИС, что позво-

лит повысить эффективность повторного использования компонентов ИС за счет снижения сложности сопоставления терминов различных ПрО и устранения ошибок при проведении подобного сопоставления.

Для решения данной задачи в рамках интеллектуальной ИТ ускоренной разработки ИС предлагается реализовать следующие ИТ-услуги [6]:

- а) выявление требований к ИС;
- б) формирование иерархий терминов ПрО создаваемой ИС;
- в) сравнение сформированных иерархий терминов ПрО ИС с разработанными ранее иерархиями терминов ПрО;
- г) выделение и уточнение паттернов требований к ИС;
- д) синтез архитектуры ИС на основе выделенных паттернов требований к ИС;
- е) осуществление взаимно-однозначного отображения паттернов ИС в сущности базы данных и классы программного обеспечения (ПО) разрабатываемой ИС.

ИТ-услуга «Выявление требований к ИС» представляет собой совокупность методов, приемов и способов сбора требований к создаваемой ИС вне зависимости от способа представления этих требований, а также их последующего анализа, обработки и формализованного описания. Основным результатом данной ИТ-услуги следует считать сформированные представления требований к создаваемой ИС на уровне информации и на уровне данных, необходимые для последующего их отображения в представлении требований на уровне знаний и создания онтологии анализируемой ПрО.

ИТ-услуга «Формирование иерархий терминов ПрО создаваемой ИС» представляет собой совокупность методов, приемов и способов непосредственного создания на основе различных описаний требований к ИС онтологии терминов ПрО создаваемой ИС. В общем случае онтологии терминов ПрО, выделенных из требований к создаваемой ИС, представляют собой иерархии структур данных, правил их обработки и интерфейсов взаимодействия (спецификаций наборов входных и выходных данных). При этом онтология одной ПрО, соответствующей БП предприятия, как правило, имеет несколько корневых узлов, то есть характеризуется несколькими иерархиями понятий, совместно используемыми для достижения целей ИС и имеющими между собой горизонтальные связи. Это утверждение наглядно демонстрируют иерархии бизнес-классов ПО ИС, в которых отдельные иерархии наследуемых классов, как правило, связаны отношениями ассоциации (агрегации или композиции) и характеризуются наличием ситуаций их совместного использования.

Аналогичным образом предлагается описывать и онтологии каждого элемента библиотеки готовых компонентов ИС (паттернов реализованных требований). При этом онтологии ПрО создаваемой ИС и онтологии элементов реализованных ИС имеют принципиальные отличия. В онтологии ПрО создаваемой ИС отражаются термины и требования, характерные для конкретного объекта автоматизации

и его БП. Эта онтология может быть избыточна, допускает синонимию и неоднозначность описаний отдельных терминов и структур и не предполагает обязательное отражение в ИС в неизменном виде. В то же время онтология элементов реализованной ИС является системоцентричным описанием реализованных паттернов и описывает структуры важных для ИС данных (храняемых и обрабатываемых), а также правила их обработки. В данном описании избыточность и неоднозначность не допускаются, поскольку любое его изменение требует модификации элементов разработанных решений по информационному и программному обеспечению ИС.

На основе концепции представления требований к ИС, рассмотренной авторами в [7], представление онтологий создаваемой ИС, отражающих определенную ПрО, можно описать как универсум понятий и терминов, используемых в данной ИС. Универсум реализованных требований к ИС, описывающий библиотеку реализованных элементов ИС, является универсумом понятий, терминов, структур данных, доступных разработчику при разработке новых ИС на базе существующих систем. Над этими универсумами возможно выполнение операций расширения (введения в онтологию новых терминов) и сужения (исключения неиспользуемых терминов). Критерием добавления (или удаления) в онтологию иерархии терминов ПрО является наличие (отсутствие) горизонтальных понятийных связей (аналог отношения ассоциации в диаграмме классов) между анализируемым термином и остальными элементами онтологии.

Таким образом, применение теории фреймов для описания иерархий онтологий ПрО и реализованных элементов ИС позволяет создать их формальное описание в виде знаний, обеспечить их легкую интерпретацию Потребителем и Поставщиком в виде визуальных диаграмм классов и определить формальное условие необходимости присутствия каждого отдельного элемента в онтологии требований к ИС. Данное условие применимо как в ходе формирования онтологии создаваемой ИС на основе онтологий реализованных ранее требований к ИС, так и в ходе расширения библиотеки реализованных требований к ИС по результатам выполнения очередного проекта разработки ИС.

Основным результатом данной ИТ-услуги следует считать множество сформированных представлений требований к создаваемой ИС на уровне знаний, выделение из этих представлений универсума требований к проектируемой ИС и универсума библиотеки реализованных требований, описанных с применением фреймового представления знаний. Выполнение данной ИТ-услуги завершает подготовку исходных данных для сопоставления различных ПрО.

ИТ-услуга «Сравнение сформированных иерархий терминов ПрО ИС с разработанными ранее иерархиями терминов ПрО» представляет собой совокупность методов, приемов и способов анализа и сопоставления сформированных иерархий терминов ПрО проектируемой ИС с разработанными ранее иерархиями терминов ПрО. Целями такого анализа

и сопоставления является поиск ответов на следующие вопросы:

а) какие существующие элементы ИС и в каком объеме могут быть применены;

б) в каком объеме необходима разработка новых или расширение (дополнение) существующих элементов ИС.

Основная сложность анализа онтологий различных ПрО связана с проблемой синонимии понятий, когда одно и то же, по сути, явление реального мира, характеризующееся схожим набором атрибутов, в разных ПрО (или даже в пределах одной ПрО) имеет несколько различных названий. Набор атрибутов и их названий в различных ПрО также может отличаться (синонимия атрибутов). В то же время, приведение названий атрибутов и объектов различных ПрО к единому глоссарию позволяет осуществлять операции пересечения множеств атрибутов, описывающих различные понятия ПрО, и оценивать степень их подобия.

Таким образом, основными результатами данной ИТ-услуги следует считать следующие группы решений, принимаемых Поставщиком и Потребителем:

а) решения о наследовании терминов (добавление терминов к существующим деревьям онтологий в роли ветвей или листьев) и, соответственно, о повторном использовании существующих элементов ИС;

б) решения о формировании новых узлов иерархий онтологий и, соответственно, о разработке новых элементов.

ИТ-услуга «Выделение и уточнение паттернов требований к ИС» представляет собой совокупность методов, приемов и способов выполнения предварительных работ, необходимых для синтеза варианта конфигурации создаваемой ИС. Полностью формализовать процесс принятия решений о наследовании терминов невозможно. Оценка подобия структур данных позволяет только сформировать положительную или отрицательную рекомендацию, а собственно принятие решения о применении термина, как и при анализе синонимов терминов остается за человеком-экспертом. Таким образом, основным результатом данной ИТ-услуги следует считать доработку существующих или создание новых аналитических описаний паттернов требований в виде онтологий, представленных иерархиями фреймов.

ИТ-услуга «Синтез архитектуры ИС на основе выделенных паттернов требований к ИС» представляет собой совокупность методов, приемов и способов синтеза слабосвязной сервис-ориентированной архитектуры ИС с применением существующих, доработанных или новых описаний паттернов требований к ИС, а также реализаций этих паттернов в виде отдельных ИТ-услуг и соответствующих ИТ-сервисов. Применение паттернов требований, каждый из которых описывается конечной онтологией взаимосвязанных терминов ПрО, в значительной степени влияет на процесс проектирования архитектуры ИС. Каждый паттерн требования предполагает свою автономную реализацию применительно, в первую очередь, к корневым терминам деревьев

онтологий, имеющих наибольший уровень абстракции. Конечной целью формирования паттернов требования является не максимальное соответствие ПрО создаваемой ИС (для учета специфики ПрО используются листья и ветви в деревьях онтологий), а максимально полное раскрытие базового узлового понятия онтологии. Следовательно, проектирование паттернов требований, реализуемых в виде набора автономных ИТ-услуг и соответствующих ИТ-сервисов, наиболее целесообразно и даёт максимальный эффект с точки зрения сервисного подхода к созданию ИС.

Таким образом, основным результатом данной ИТ-услуги следует считать синтезированные варианты описания архитектуры создаваемой ИС, которые позволяют описывать систему одновременно и как единый целостный объект исследования, и как совокупность ИТ-услуг, предлагаемых в рамках создаваемой ИС, а также как совокупность ИТ-сервисов, реализующих заявленные ИТ-услуги.

ИТ-услуга «Осуществление взаимно-однозначного отображения паттернов ИС в сущности базы данных и классы ПО создаваемой ИС» представляет собой совокупность методов, приемов и способов выполнения работ по физическому проектированию ИС в соответствии с сформированным ранее планом проекта создания данной ИС. В рамках данной ИТ-услуги осуществляется взаимно-однозначное отображение паттернов требований к ИС в сущности базы данных и классы ПО создаваемой ИС для уточняемых или новых паттернов требований, а также организация их взаимодействия в составе конфигурации создаваемой ИС.

Основными результатами данной ИТ-услуги следует считать описания схемы базы данных и классов прикладного ПО ИС, сформированные на основе выделенных ранее паттернов требований к ИС, в том числе – по результатам повторного использования элементов библиотеки готовых компонентов ИС.

Поскольку сеть фреймов и диаграмма классов основаны на применении общего математического аппарата, одинаковых типов связей и структур данных, отображение сети фреймов онтологии ПрО в диаграмму классов ПО осуществляется естественным образом: фреймы отображаются в классы с такими же названиями и наборами атрибутов и методов, связи между фреймами – в одноименные им связи генерализации и ассоциации классов.

Применительно к информационному обеспечению ИС основная сложность заключается в отсутствии в наиболее распространённой в настоящий момент реляционной модели данных иерархических связей и механизма наследования. Для реализации задачи объектно-реляционного отображения в настоящее время существует несколько типовых подходов, наиболее распространёнными из которых являются отображение каждого дочернего класса (включая наследуемые атрибуты) в отдельные таблицы базы данных, а также эмуляция механизма наследования за счет применения идентифицирующих связей между сильными (родительскими) и слабыми (дочерними) сущностями.

В рассматриваемой ИТ предлагается применять второй из описанных выше подходов. Взаимнооднозначное отображение терминов ПрО в таблицы базы данных ИС требует применения соответствующей методики построения. Согласно этой методике, база данных проектируется как набор взаимосвязанных витрин данных, реализованных в виде отдельных схем данных. В каждой из этих витрин таблицей фактов является корневой элемент конкретной иерархии фреймов, а консольным таблицам и таблицам измерений соответствуют уточняющие понятия ПрО. При этом наследуемые фреймы представляются как слабые сущности, зависящие от корневого понятия соответствующей иерархии фреймов онтологии ПрО, а каждая отдельная витрина данных представляет собой многомерный куб, который может быть свёрнут к одному универсальному отношению, содержащему все множество возможных атрибутов конкретной витрины данных.

В этом случае названия таблиц соответствуют фреймам, атрибуты таблиц – атрибутам фреймов, а присоединенные процедуры фреймов отображаются в хранимые процедуры и функции, а также в триггеры реляционной схемы данных. Иерархические связи отображаются в наборы сущностей, связанных между собой идентифицирующими связями, а связи-ассоциации – в ссылочные ограничения целостности данных, реализуемые с применением первичных и внешних ключей.

Рассмотрим особенности реализации наследования в реляционной базе данных более подробно. Для решения данной задачи предлагается выделение абстрактной обобщённой сильной сущности, которая соответствует корневому элементу конкретной иерархии фреймов и является универсальной для различных ПрО. Атрибуты такой сущности несут общий характер, а её применение предполагается при реализации требований, характерных для нескольких ИС. Для фреймов второго и выше уровней иерархии создаются слабые сущности, которые полностью зависят от более общей и универсальной сущности предыдущего уровня иерархии и дополняют её, поскольку содержат дополнительную информацию, характерную для более конкретного класса объектов ПрО ИС. Такая слабая сущность в качестве ключевого атрибута (РК) содержит внешний ключ (FK), реализующий связь «1:0..1» с обобщающим сильным типом сущности. При этом в случае, когда соответствующий ей фрейм имеет потомков, такая сущность в свою очередь является сильной для сущностей, соответствующих следующему уровню иерархии.

Для упрощения работы с базой данных ИС со стороны Поставщика ПО и Потребителя реализуются виртуальные таблицы – представления, которые содержат всю информацию об объекте ПрО, объединяя данные из нескольких таблиц (аналогично тому, как в ООП классы-потомки наследуют все атрибуты классов родителей). Использование представлений позволяет скрыть разветвлённость схемы данных от пользователей и обеспечить соответствие логического представления данных конкретной ПрО.

Несмотря на зависимость слабой сущности от сильной, распределение набора атрибутов одного реального объекта ПрО между несколькими таблицами, связанными идентифицирующими связями и отношениями «1:0..1» и созданными с целью реализации в реляционной БД иерархических связей, создаёт необходимость обеспечения целостности данных на уровне нескольких таких таблиц. Подобная целостность данных нехарактерна для классической ситуации размещения таких атрибутов в одной таблице (на этапе концептуального проектирования таблицы, имеющие связи «1:1», всегда объединяются в одну таблицу).

Вследствие этого именно необходимость отображения иерархии фреймов в реляционные схемы витрин данных обуславливает необходимость применения приведенного в предыдущем подразделе правила классификации фреймов как корневых самостоятельных понятий и в результате – представления сети фреймов в виде «низкорослого леса».

Поскольку ни целостность сущностей, ни ссылочная целостность в полной мере не обеспечивают решение данной задачи, предлагается использование триггеров. Триггеры реализуют сложные ограничения целостности данных, которые нереализуемы описательными ограничениями, устанавливаемыми при создании таблиц.

Можно выделить четыре основных типа триггеров, обеспечивающих целостность данных в обобщённой и уточняющих таблицах [7]:

а) триггеры, обеспечивающие сюръективное отображение данных в двух таблицах (обязательное наличие соответствующих записей в главной и зависимой таблицах, связь «1:1», поскольку наличие идентифицирующей связи обеспечивает только наличие связи «1:0..1»);

б) триггеры, обеспечивающие уникальность значений комбинаций атрибутов, расположенных в разных таблицах (аналог уникального ключа в одной таблице; для этой цели могут также применяться функциональные индексы);

в) триггеры, обеспечивающие контроль внесения данных в дочерние таблицы, когда сильная сущность связана с несколькими слабыми, а данные в слабых сущностях не должны пересекаться между собой (например, сильная сущность «человек», слабые – «преподаватель» и «студент»);

г) триггеры, обеспечивающие заполнение (аналог *not null option*) необязательных полей сильных сущностей, если это требуется для зависящих от них слабых сущностей (например, для абстрактной сильной сущности «человек» поле «контактный телефон» может быть необязательным, а для наследуемой от неё сущности «сотрудник» это же поле, физически расположенное в главной таблице, является обязательным).

Основные этапы построения сети фреймов и соответствующие им этапы отображения и реализации требований в описания элементов ПО и информационного обеспечения ИС приведены в табл. 5.1 [7, 8].

Пример отображения иерархии фреймов ПрО в диаграмму классов ПО ИС приведен на рис. 1. Пример отображения сети фреймов в ER-диаграмму базы данных ИС приведен на рис. 2. Полученные в

результате отображения иерархия классов и схема базы данных аналогичны фрагменту сети фреймов, приведенному на рис. И.4 и описывающему понятия физического и юридического лица. На рис. 1 пред-

ставлена иерархия «Персона»-> «Юридическое лицо», «Физическое лицо» -> «Игрок» (PersonPhysicalLuk). При этом следует отметить:

Таблица 1 – Этапы формирования сети фреймов, диаграммы классов и схемы данных информационной системы

Моделирование ПрО	Проектирование ПО	Проектирование ИО
2	3	4
Выделение абстрактных понятий и их характеристик, универсальных для различных ПрО.	Выделение обобщенных (часто абстрактных) классов и наборов полей, универсальных для различных ПрО.	Выделение обобщенных сильных сущностей и наборов атрибутов, универсальных для различных ПрО.
Группировка выделенных понятий по признаку совместного использования. Выделение «подобластей».	Формирование пакетов классов.	Формирование отдельных схем (витрин) данных.
Формирование иерархий обобщенных понятий (выделение корневых фреймов, построение деревьев, состоящих из корня и ветвей).	1. Формирование иерархий абстрактных классов. 2. Формирование параметризуемых Generic-классов. 3. Формирование диаграмм абстрактных классов, объединённых в пакеты.	1. Формирование обобщенных слабых сущностей, уточняющих сильные (связи 1: 0..1). 2. Формирование представлений, объединяющих обобщенные сильные и слабые физические сущности в логические. 3. Анализ функциональных зависимостей и определение необходимых ограничений целостности данных, в том числе реализуемых с применением триггеров. 4. Формирование ER-диаграмм.
Выделение понятий и их характеристик, характерных для конкретной ПрО.	Выделение классов и наборов полей, характерных для конкретной ПрО.	Выделение сущностей и наборов атрибутов, характерных для конкретной ПрО.
Анализ возможности наследования фреймов выделенных понятий от универсальных, выделение новых универсальных понятий.	Анализ возможности наследования классов от существующих абстрактных классов.	Анализ возможности формирования слабых сущностей, уточняющих обобщенные сильные.
Формирование иерархий понятий, характерных для конкретной ПрО (добавление листьев к иерархиям понятий).	1. Формирование иерархий классов, характерных для конкретной ПрО. 2. Параметризация классов-листьев. 3. Формирование диаграмм классов, объединённых в пакеты.	1. Формирование слабых сущностей-листьев, уточняющих обобщенные сущности. 2. Формирование представлений, объединяющих сильные и слабые физические сущности в логические. 3. Анализ функциональных зависимостей и задание ограничений целостности данных.
Расширение библиотеки паттернов новыми обобщенными понятиями, описанными фреймами.	Расширения ПО библиотеки паттернов новыми обобщенными классами.	Расширение ИО библиотеки паттернов новыми обобщенными сущностями.

а) факт повторного использования требования учета юридических и физических лиц, реализованных также в рассматриваемой ИТ;

б) отличия в приведенных иерархиях фреймов на уровне объектов-листьев, которые отражают специфику конкретных ПрО: «Внешнее лицо» для рассматриваемой ИТ и «Игрок» для создаваемой ИС.

На рис. 2 также показан фрагмент иерархии классов, реализующих паттерн проектирования ПО «Data Access Objects» (DAO), который предполагает создание иерархии специализированных классов для

обеспечения взаимодействия ПО с базой данных, аналогичной иерархии бизнес-классов.

Характерной особенностью реализации DAO-слоя ПО является применение Generic-классов, параметризуемых на этапе выполнения (применительно к языку программирования Java). Возможно также применение паттерна проектирования ПО «Factory», применяемого для динамического создания объектов классов в зависимости от заданных условий.

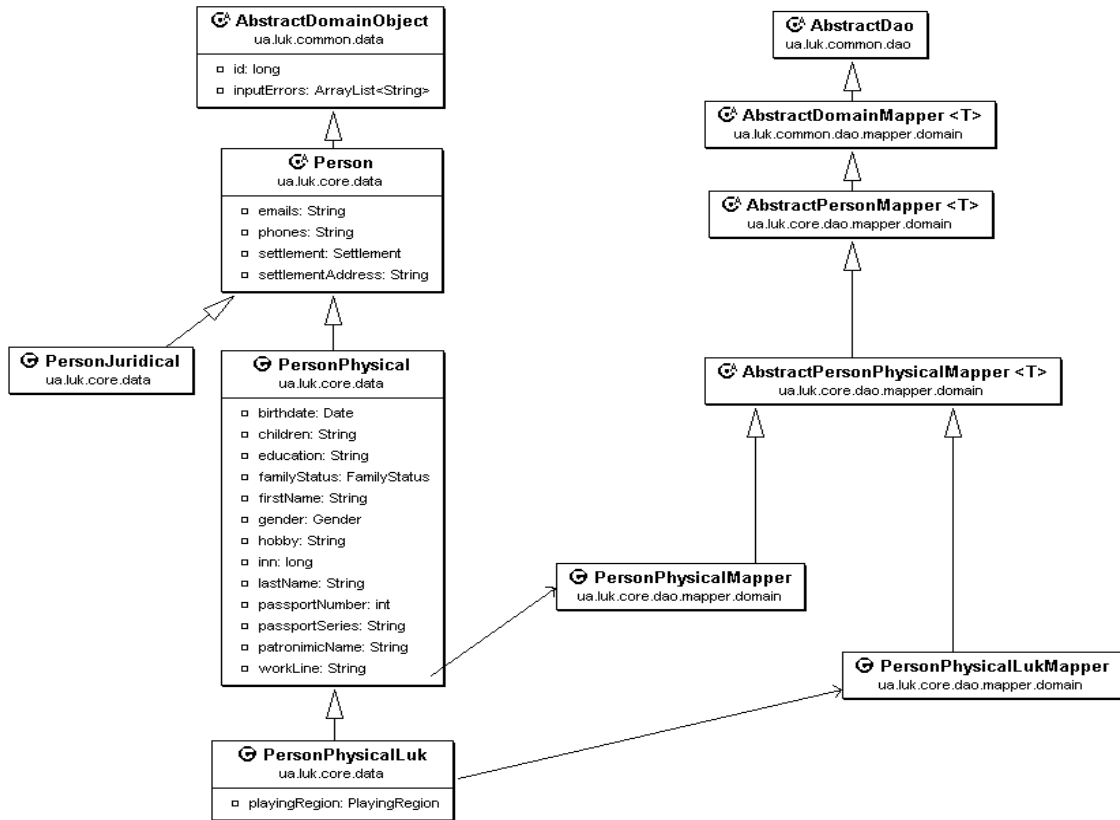


Рисунок 1 – Пример отображения сети фреймов в диаграмму классов программного обеспечения информационной системы

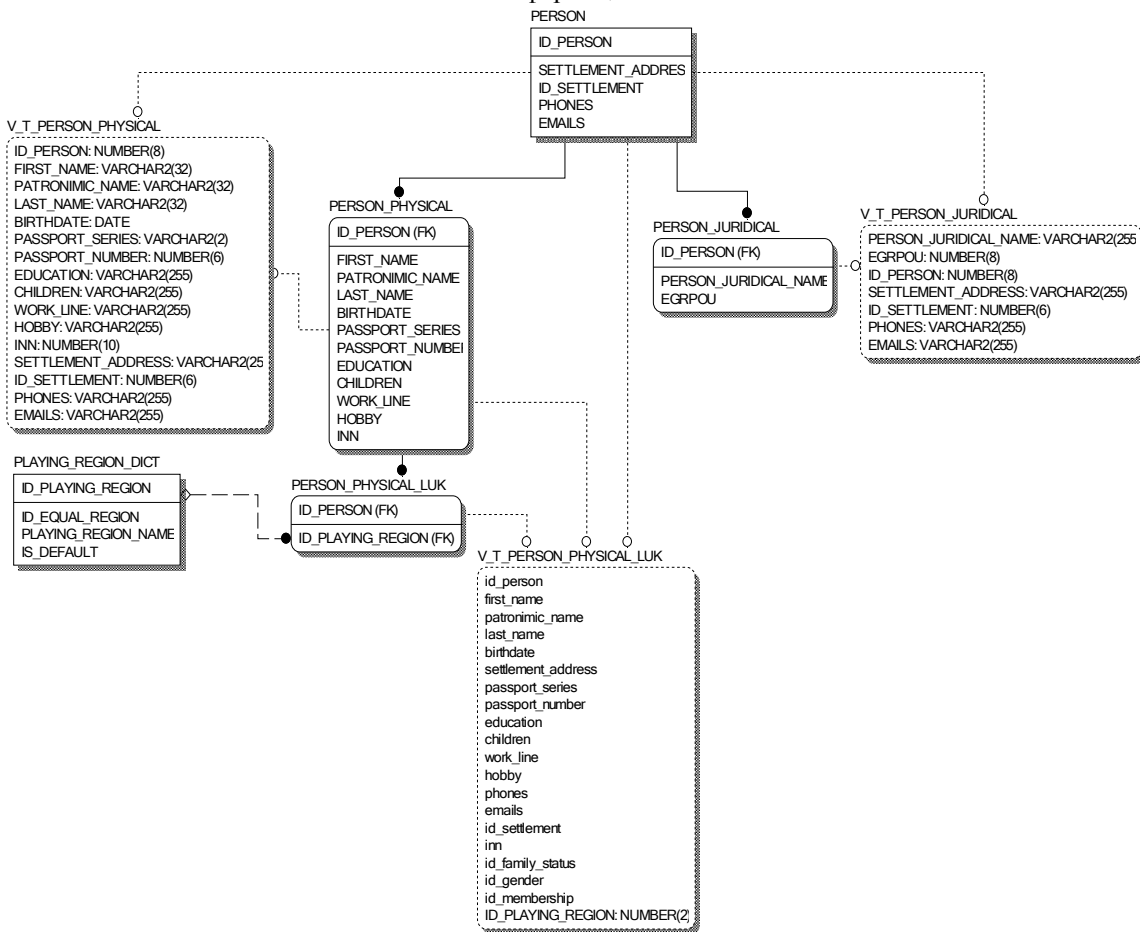


Рисунок 2 – Пример отображения сети фреймов в ER-диаграмму информационного обеспечения информационной системы

Поскольку при загрузке такими классами данных из базы осуществляется создание экземпляров классов ПО, в них нельзя применять обобщающие типы данных (в таком случае при загрузке данных об игроке, например, был бы создан объект «Физическое лицо», в котором отсутствуют атрибуты, специфичные для его наследников, и данные об игровом регионе в атрибут «playingRegion» загружены бы не были).

На рис. 2 приведена соответствующая схема данных, являющаяся результатом объектно-реляционного отображения сети фреймов в термины реляционной модели данных. Работу с этой схемой данных обеспечивают классы DAO-слоя ПО, показанные на рис. 1.

Необходимость работы с аналитическими требованиями и формирования библиотеки реализованных требований существенно увеличивают требования к ведению проектной документации. В наибольшей степени это касается наименования объектов программного и информационного обеспечения ИС. Целесообразной является разработка правил именования классов ПО, таблиц информационного обеспечения, а также их атрибутов, методов, процедур и других элементов.

Применение единой системы наименования элементов программного и информационного обеспечения необходимо для обеспечения двустороннего отображения диаграмм бизнес- и DAO-классов ПО в элементы информационного обеспечения ИС. Кроме того, это позволяет осуществлять автоматическую генерацию ограничений целостности при отображении сети фреймов, описывающих требования к ИС в элементы информационного обеспечения системы (автоматическую генерацию первичных и внешних ключей, а также описанных выше типовых триггеров на основе заранее заданных шаблонов). Для автоматической генерации ограничений целостности необходимо, чтобы правила именования были применены ко всем объектам базы данных независимо от того, созданы они вручную, или автоматически, например, с применением процедуры Forward Engineering CASE-средств. Для контроля применения сформулированных и формализованных правил именования объектов возможна также реализация процедур автоматической проверки их соблюдения с применением словаря базы данных.

Введение в модель фрейма понятия «интерфейс фрейма» позволяет также реализовать отображение сети фреймов в элементы графического интерфейса пользователя. Требование на уровне знаний может быть описано в виде фрейма f , который может быть разделен на отдельные структуры данных, описываемые интерфейсами фрейма. Как и в объектно-ориентированном программировании, несколько различных фреймов, описывающих разные требования, могут реализовывать один и тот же интерфейс, то есть иметь общие структурные элементы. Применение интерфейсов фреймов позволяет применять в различных фреймах единые описания типовой структуры данных [9]. При этом класс или фрейм может одновременно реализовывать несколько интерфейсов. Кроме того, большинство современных языков объектно-ориентированного программиро-

вания не предоставляют возможность реализации множественного наследования, реализация которого вызывает неоднозначности (например, если в наследуемых классах существуют одноименные атрибуты). Применение интерфейсов позволяет осуществлять не только вертикальный анализ иерархической структуры классов или фреймов, но и горизонтальный анализ структурных элементов отдельных классов, что компенсирует отсутствие механизмов множественного наследования.

Комбинации двух указанных видов анализа структур фреймов позволяет осуществлять отображение фреймов, описывающих требования, не только в бизнес- и DAO-классы, но и в элементы ПО, описывающие интерфейс пользователя.

Разработка варианта интерфейса пользователя для заполнения данными составной части фрейма значительно менее трудоёмка, чем для всего элемента как единого целого, и позволяет сгруппировать поля для ввода логически связанных данных в виде отдельных визуальных компонентов, соответствующих либо интерфейсам классов, либо классам-родителям, структура которых является общей для нескольких классов-потомков. В ПО такие компоненты интерфейса пользователя для работы с документами могут быть представлены отдельными блоками-контейнерами полей применительно как к web-, так и к GUI-интерфейсу ПО.

Если все фактографические поля класса описаны в реализуемых данным классом интерфейсах или наследуемых классах (структурных элементах фрейма el), которым в соответствии поставлены типовые визуальные компоненты, становится возможным автоматическое формирование интерфейса пользователя для работы с ним из типовых визуальных компонентов. При этом разработка отдельного варианта интерфейса пользователя для всего фрейма не требуется, что позволит снизить временные и трудозатраты на проектирование, избежать дублирования программного кода и обеспечить его повторное использование. Таким образом, будут снижены объем и количество ошибок в исходном коде ПО ИС, повышен коэффициент повторного использования стандартных программных компонентов [9].

Декомпозиция элемента документа упрощает адаптацию библиотеки реализованных требований к новым требованиям в ходе создания новых ИС или изменениям существующих требований в процессе их сопровождения, обеспечивает централизованное изменение требований к ИС совместно с компонентами интерфейса пользователя.

Очевидно, что одному структурному элементу фрейма el может соответствовать только один визуальный компонент pnl (сюрьективное отображение $el \rightarrow pnl$), используемый для ввода и отображения данных. Поскольку описание фрейма может содержать множество описаний интерфейсов этого фрейма, а иерархические связи между фреймами в сети фреймов позволяют определить множество наследуемых фреймов, данные соответствия позволяют в результате анализа структуры и иерархических связей фрейма сформировать интерфейс поль-

зователя для заповнення фрейма фактографіческими даними.

Поскольку понятие «интерфейс фрейма» введено в сеть фреймов из объектно-ориентированного программирования, особенности реализации интерфейсов классов в технологии объектно-ориентированного программирования действительны и применительно к представлению знаний в виде фреймов:

а) фрейм может содержать слоты, которые не принадлежат ни одному из реализуемых интерфейсов фрейма;

б) несколько интерфейсов фрейма могут регламентировать наличие во фрейме одного и того же свойства-слота;

в) свойство-слот, объявленное хотя бы в одном из реализуемых фреймом интерфейсов, один и только один раз должно быть объявлено во фрейме-прототипе.

Расширение модели сети фреймов понятием «интерфейс фрейма» (в объектно-ориентированном программировании реализация интерфейса класса, как правило, указывается при объявлении класса и описании его структуры) дополняет модель логического вывода с использованием иерархии фреймов.

Таким образом, модель формирования интерфейса пользователя ИС (*PUI*) для заповнення фрейма даними, основанная на анализе иерархических связей фрейма и его структуры, предлагается представить в следующем виде:

$$PUI(f) = \forall el \in f \Rightarrow \text{если } \exists pnl : el \rightarrow pnl = true \Rightarrow pnl \in UI \quad (1)$$

где *f* – фрейм, применяемый для описания требований в виде знаний; *el* – элемент фрейма, являющийся либо интерфейсом, реализуемым фреймом, либо фреймом-прототипом, который он наследует; *pnl* – визуальный компонент, элемент интерфейса пользователя ПО ИС; $UI = \{pnl_1, \dots, pnl_n\}$ – множество визуальных компонентов, составляющих интерфейс пользователя ИС для работы с бизнес-классом ПО, соответствующим фрейму *f*.

В модели (1) множество элементов фрейма *el* анализируется на предмет соответствия каждому из них визуального компонента интерфейса пользователя, используемого для отображения и внесения данных в бизнес-классы ПО. В случае наличия такого отображения ($el \rightarrow pnl$) визуальный компонент, соответствующий данному элементу фрейма, добавляется в интерфейс пользователя.

Данная модель может применяться для динамического формирования интерфейса пользователя ПО ИС в момент его отображения. Процедура динамического формирования интерфейса пользователя включает следующие этапы:

а) выделение множества элементов класса: интерфейсов, реализуемых фреймом, или наследуемых им фреймов-прототипов;

б) формирование множества визуальных компонентов для отображения в интерфейсе пользователя (1);

в) ранжирование множества визуальных компонентов по заданным приоритетам, определяющим порядок их расположения в интерфейсе пользователя;

г) отображение сформированного интерфейса пользователя ПО ИС для заповнення соответствующему фрейму бизнес-класса фактографіческими даними.

В результате анализа элементов класса интерфейсов формируется список компонентов интерфейса пользователя, отображаемых в порядке, определяемом рангом или приоритетом, назначаемым каждому компоненту.

Таким образом, структура фрейма и реализующего этот фрейм программного класса однозначно определяют структуру интерфейса пользователя ИС. Этот интерфейс может формироваться как предварительно разработчиками, так и автоматически во время динамики выполнения исходных кодов ПО, и регламентирует набор фактографических данных, которые отображаются пользователю, могут быть им модифицированы или внесены в бизнес-класс ПО, а соответственно и отображены в базе данных ИС.

ВЫВОДЫ. Описанный подход к реализации объектно-реляционного отображения данных уменьшает производительность системы в целом за счёт множества дополнительных сущностей и связей «1:1», которые не рекомендуется использовать в схеме базы данных. Однако применение предлагаемого подхода сводит к минимуму контролируемую избыточность данных, сохранив преимущества применения реляционной модели для организации хранения данных в ИС. Кроме того, как было показано на примере формирования интерфейса пользователя, применение сети фреймов для описания требований к ИС позволяет также сохранить преимущества применения технологий объектно-ориентированного программирования.

Таким образом, отображение сети фреймов в элементы программного и информационного обеспечения ИС обеспечивает реализацию предлагаемой интеллектуальной ИТ ускоренной разработки ИС, что позволяет облегчить Поставщику выполнение процессов создания последующих ИС на основе ранее разработанных систем, используемых в качестве прототипов. Тем самым, время разработки новых ИС значительно сокращается. Еще одним достоинством такого подхода является то, что в случае значительных изменений в БП, характерных для нескольких ИС, на этапе сопровождения данных систем в них достаточно перестроить только обобщённую часть системы, не выполняя индивидуальную доработку каждой системы.

ЛИТЕРАТУРА

1. Технология быстрого проектирования информационных систем / М.В. Евланов, М.А. Керносов, М.Э. Лотфулина // Информационные системы и технологии: материалы Междунар. науч.-техн. конф., Морское-Харьков, 22–29 сентября 2012 г.: тезисы докладов / Редкол.: А.Д. Тевяшев (отв. ред.) и др. – Харьков: НТМТ, 2012. – С. 35.

2. ГОСТ ИСО/МЭК 15288–2005. Системная инженерия. Процессы жизненного цикла систем. – Введ. 01–01–2007. – М.: Стандартинформ, 2006. – 57 с.

3. Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Д. Леффингуэлл, Д. Уидриг. – М.: Вильямс, 2002. – 448 с.

4. Вигерс К.И. Разработка требований к программному обеспечению. – М.: Русская редакция, 2004. – 576 с.

5. Кобёрн А. Современные методы описания функциональных требований к системам. – М.: «Лори», 2002. – 288 с.

6. ИТ-услуги информационной технологии управления проектами ускоренной разработки информационных систем / В.М. Левыкин, М.В. Евланов // Информационные системы и технологии: материалы 2-й Международ. науч.-техн. конф., Евпа-

тория–Харьков, 16–22 сентября 2013: тезисы докладов. – Харьков: НТМТ, 2013. – С. 53–55.

7. Паттерны проектирования требований к информационным системам: моделирование и применение / В.М. Левыкин, М.В. Евланов, М.А. Керносов. – Харьков: ООО «Компанія «СМІТ»», 2014. – 320 с.

8. Информационная технология ускоренной разработки информационных систем /М.В. Евланов, М.А. Керносов, М.Э. Керносова // Управляющие системы и машины. – 2014. – № 1 (249). – С. 62–69.

9. Информационная технология динамического формирования интерфейса пользователя системы управления электронным документооборотом / В.М. Левыкин, М.А. Керносов // Вісник ХНТУ. – 2008. – № 1 (30). – С. 182–188.

FEATURES OF DOMAIN ONTOLOGY INTO THE ELEMENTS OF INFORMATION SYSTEM MAPPING

V. Levykin, M. Ievlanov, M. Kernosov, M. Kernosova

Kharkiv National University of Radioelectronics

prosp. Lenina, 14, Kharkiv, 61166, Ukraine. E-mail: iyc@kture.kharkov.ua

The main challenge of reducing time spent on the development of modern information systems. Highlighted the problem domain analysis and portfolio of completed projects for the possibility of the use-of earlier decisions in the creation of a new information system. The basic functions of information technology, which allows to solve this problem. An approach to the display of network frames that describes the domain ontology to describe the classes of software and description of the entity database information system. The basic stages of the formation of a network of frames, class diagrams and data schema developed information system. We propose a list of works to be carried out on each of the selected phases. An approach to the problem of ensuring the integrity of data generated by the database schema information system. The model of formation of user interfaces developed information system.

Key words: information system, ontology, class, entity.

REFERENCES

1. Yevlanov, M.V., Kernosov, M.A. and Lotfulina, M.E. (2012), "Technology fast design of information systems", *Information Systems and Technology: Proceedings of the Intern. scientific-technical. conf.*, Morskoye–Kharkov, NTMT, september 22–29, p. 35.

2. ISO/IEC 15288:2002 (2006) System engineering – System life cycle processes, 01–01–2007, Standartinform, Moscow, Russia.

3. Leffingwell, D. and Widrig, D. (2002), *Printsipy raboty s trebovaniyami k programnomu obespecheniyu . Unifitsirovannyy podkhod* [Managing Software Requirements: A Unified Approach], Translated by Orekhova, N.A., Vil'yams, Moscow, Russia.

4. Wieggers, K.E. (2004), *Razrabotka trebovaniy k programnomu obespecheniyu* [Software requirements], Translated by Microsoft Corporation, Russkaya redaktsiya, Moscow, Russia.

5. Cockburn, A. (2002), *Sovremennyye metody opisaneya funktsional'nykh trebovaniy k sistemam* [Writing effective use cases], Translated by Borisova, Ye, "Lori", Moscow, Russia.

6. Levykin, V.M. and Ievlanov, M.V. (2013), "IT Services Information Technology Project Management rapid development of information systems", *Information Systems and Technology: Proceedings of the 2nd Internatio. scientific-technical. conf.*, Evpatoria–Kharkiv, NTMT, september 16–29, pp. 53–55.

7. Levykin, V.M., Ievlanov, M.V. and Kernosov, M.A. (2014), *Paterny proektuvannya vymoh do informatsiynykh system: modelyuvannya i zastosuvannya* [Design patterns requirements for information systems: modeling and application], ООО «Компанія «СМІТ», Kharkiv, Ukraine.

8. Ievlanov, M.V., Kernosov, M.A. and Kernosova, M.E. (2014), "Information technology rapid development of information systems", *Upravlyayushchiye sistemy i mashiny*, no. 1 (249), pp. 62–69.

9. Levykin, V.M. and Kernosov, M.A. (2008), "Information technology is the dynamic formation of user interface management system electronic document", *Visnyk KHNTU*, no. 1 (30), pp. 182–188.

Стаття надійшла 26.09.2014.