

СИНТЕЗ СХЕМЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ НА ОСНОВЕ ОНТОЛОГИЧЕСКОГО ОПИСАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ

УДК 004:681.5

ЕВЛАНОВ Максим Викторович

к.т.н., доц., доц. каф. информационных управляющих систем Харьковского национального университета радиоэлектроники.

Научные интересы: методологии, фреймворки, модели и методы ускоренной разработки информационных систем.

КЕРНОСОВА Марина Эдуардовна

ст. оператор ЭВМ Центра информационных систем и технологий Харьковского национального университета радиоэлектроники, студентка гр. КН-10-3.

Научные интересы: web-базированные информационные системы, экспертные системы.

ВВЕДЕНИЕ

В настоящее время все большее количество предприятий и организаций сталкиваются с необходимостью комплексной автоматизации своей деятельности, охватывающей все структурные подразделения предприятия. Значительная часть существующих информационных систем (ИС) являются тиражируемыми разработками, ориентированными на автоматизацию основных бизнес-функций типовых организаций или предприятий. В то же время применение типовых систем для автоматизации деятельности организаций с нетиповыми бизнес-процессами (БП), подверженными регулярным изменениям, затруднено. Для таких организаций требуются уникальные разработки.

Разработка современных ИС для автоматизации различных БП предприятий предполагает решение наукоёмких задач. Помимо оптимизации взаимодействия программного обеспечения (ПО) и аппаратных ресурсов большое внимание уделяется рациональному использованию трудовых ресурсов и снижению времени разработки системы. В условиях сильной конкуренции для заказчика и разработчика время является наиболее критичным ресурсом, оказывающим значительное влияние на планирование проекта по разработке ИС.

Одним из способов сокращения временных затрат на разработку является применение гибких моделей жизненного цикла (ЖЦ) системы и намеренный пропуск определённых стадий ЖЦ (традиционная каскадная модель ЖЦ разработки ИС предусматривает упорядоченную последовательность таких основных этапов: анализ требований, проектирование, реализация, тестирование и отладка, эксплуатация и сопровождение).

Таким образом, при проектировании новой системы разработчик либо строит модель ИС «с нуля», либо перестраивает ранее разработанную систему с учётом требований к новой ИС, выполняя при этом необходимые этапы проектирования.

Повторное использование типовых проектных решений более предпочтительно, поскольку позволяет значительно сократить трудозатраты на этапах проектирования всех видов обеспечения системы, а также снизить сложность её сопровождения. Однако этот вариант связан с решением ряда наукоёмких задач. Одной из этих задач является задача изначального проектирования компонентов модульной ИС таким образом, чтобы гарантировать возможность их повторного использования (в первую очередь решений по информационному и программному обеспечению системы).

АНАЛИЗ СОВРЕМЕННЫХ ПОДХОДОВ К ОПИСАНИЮ КОМПОНЕНТОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Изучение описаний стандартизованных процессов создания ИС показывает, что ключевым понятием, задающим особенности анализа и синтеза систем, является понятие «системная архитектура» или, точнее, «архитектура системы» [1]. Большинство определений понятия «архитектура» в области компьютерных наук фокусируется на артефактах, используемых для описания архитектуры как концептуальной структуры и логической организации компьютерной системы [2]. Однако в этом случае описание архитектуры рассматривается как набор практик в рамках искусства или как практики архитектуры (причем такой набор иногда также называют архитектурой). Кроме того, существует единство семантики термина «архитектура», берущей свое начало из строительной отрасли и существующей в настоящее время в гражданской и в других отраслях. В результате определение семантики понятия «архитектура системы» становится весьма затруднительным и запутанным.

Одной из первых попыток узаконить максимально общее определение понятия «Архитектура» для компьютерных наук является стандарт IEEE 1471:2000. В нем предлагается следующее определение: «Архитектура: фундаментальная организация системы, воплощенная в ее компонентах, их взаимоотношении друг с другом и с окружающей средой, а также принципы, определяющие проектирование и эволюцию системы». Однако в последующих редакциях этого стандарта данное определение было серьезно изменено. Так, был удален термин «организация», поскольку он уже имел специализированное значение в контексте стандартов ISO. Также был заменен термин «компоненты», потому что он неправильно трактовался как «программные компоненты».

Важным также является отличие понятия «архитектура ИС» от понятия «структура системы»: если структура системы описывает только статические аспекты ее построения, то архитектура ИС определяет также и динамику поведения (behavior) соответствующей ИС, а также задает интерфейсы её взаимодействия с окружающей средой (например, с внешними объектами, другими системами и т.п.) [3].

В настоящее время наметился новый виток в развитии определения понятия «Архитектура». Принятый в 2010

г. стандарт ISO/IEC/IEEE 42010 «Systems and Software Engineering – Architecture Description» дает следующее определение понятия «архитектура»: «Архитектура [системы]: фундаментальные понятия и свойства системы в окружающей ее среде, воплощенные в ее элементах, отношениях, а также в принципах ее проектирования и развития» [2]. Данное определение учитывает основные проблемы, возникающие при попытках прикладного использования понятия «Архитектура» стандарта IEEE 1471:2000, и, по замыслу создателей, является максимально общим определением, пригодным для описания архитектур практически любых систем.

Термин «система» используется в соответствии с конвенцией ISO для того, чтобы показать, что определяемый термин «архитектура» относится к предметной области системы. По сути, термин «система» в приведенном выше определении и в пределах стандарта ISO/IEC/IEEE 42010 является основой для многих других понятий, в том числе [2]:

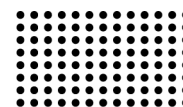
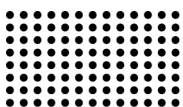
а) для определения термина «система» в соответствии с положениями стандарта ISO 15288 как «системы, которые созданы человеком и состоят из одного или нескольких следующих элементов: технические средства, программные средства, люди, процессы (например, процесс оценки), процедуры (например, инструкции оператора), основные средства и природные ресурсы (например, вода, объекты живой природы, материалы)» [1];

б) для аналогичного определения понятий «программные продукты» и «сервисы» в соответствии с описаниями стандарта ISO 12207;

в) для определения программно-основанных систем, которые в соответствии с описаниями стандарта IEEE 1471:2000 представляют собой «любую систему, в которой ПО оказывает существенное влияние на проектирование, конструирование, внедрение и развитие системы в целом» и могут включать в себя «отдельные приложения, системы в традиционном смысле этого слова, подсистемы, надсистему, линейки продуктов, семейства продуктов, целые предприятия и другие сосредоточия интересов».

То, что лежит в основе системы, определяется следующими понятиями [2]:

а) «элементы системы»: компоненты, входящие в состав системы;



б) «отношения», которые могут быть внутренними и внешними по отношению к системе;

в) «принципы проектирования и развития системы».

Различные сообщества, использующие понятие «архитектура», по-разному акцентируют внимание на этих понятиях. Так, в архитектуре ПО часто сосредотачиваются на программных компонентах в качестве элементов системы и взаимодействиях этих компонентов в качестве ключевых отношений. Системная архитектура акцентирует внимание на структурах отдельных подсистем и таких отношениях, как выделение этих подсистем. В архитектуре предприятия основное внимание уделяется принципам. Предлагаемое определение служит признанием того, что все перечисленные выше понятия могут сыграть свою роль в архитектуре.

ВЫДЕЛЕНИЕ НЕРЕШЁННЫХ ЧАСТЕЙ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

Исходя из приведенных результатов анализа, следует отметить тот важный факт, что принципы проектирования и развития системы, к которым относится и возможность повторного использования компонентов ИС, теоретически могут быть выделены архитектором путем реверс-инжиниринга из существующей системы или даже открыты в природе – в зависимости от исследуемой системы. В то же время, хотя в созданных человеком системах архитектура часто воспринимается как интенциональная позиция, такая интенциональность не является частью рассматриваемого определения. Поэтому к настоящему времени задачу повторного использования отдельных компонентов в ходе проектирования новых ИС следует считать не имеющей общего решения.

Возможность повторного использования модулей ИС должна быть заложена на этапе их проектирования. Как ИС в целом представлена решениями по видам обеспечения системы, так и каждый её модуль представлен элементами решений по видам обеспечений. Поэтому возможность повторного использования компонентов ИС должна реализовываться именно на уровне элементов видов обеспечений данных компонентов.

Ключевую роль в реализации данного свойства компонентов модульной ИС играют решения по видам

обеспечений: информационному (ИО), математическому (МО) и ПО. Именно для данных видов обеспечений ИС одной из основных целей разработки является максимально ёмкое и точное отображение особенностей предметной области (ПрО) объекта автоматизации во внутреннем представлении системы. Это позволяет конкретизировать задачу проектирования модулей ИС с гарантированной возможностью их повторного использования как задачу отражения ПрО объекта автоматизации в элементы решений по видам обеспечения системы таким образом, чтобы эти же решения с минимальными модификациями могли быть адаптированы к ПрО других ИС.

Соответственно фокус исследований смещается от этапов непосредственной разработки решений по видам обеспечений элементов системы к этапу анализа ПрО объекта автоматизации.

Анализ ПрО включает описание БП, которые могут быть охарактеризованы совокупностью входных и выходных данных, а также совокупностью правил обработки входных данных для получения выходных. При этом специфика задач проектирования ИС характеризуется системоцентричным взглядом на ПрО (анализируются важные для системы данные, которые в ней хранятся и обрабатываются).

Первоначально такое описание создается для новой ПрО, для которой разрабатываются новая ИС, впоследствии оно может быть сохранено в библиотеке готовых компонентов для ПрО, которым соответствуют выполненные проекты.

Вследствие этого можно утверждать, что описание ПрО, отраженной в ИС, является универсумом понятий, используемых в данной системе, и одновременно подмножеством понятий ПрО библиотеки готовых решений разработчика. В свою очередь, библиотека готовых решений содержит универсум понятий, доступных разработчику при разработке новых ИС на базе существующих систем.

Для проведения анализа библиотеки готовых решений на предмет возможности их повторного использования в конкретном проекте новой ИС текстового или графического описания ПрО недостаточно. Для обеспечения повторного использования компонентов ИС требуется формальное описание терминов ПрО и обеспечение учета терминов, описывающих ПрО реализо-

ванных ИС (формальное описание библиотеки существующих и новых решений).

Одним из способов формального описания ПрО ИС является формирование онтологии ПрО. Его преимуществами являются:

- возможность визуализации в виде, понятном для неспециалистов в области инженерии знаний;
- возможность совместного использования с теорией фреймов и возможность отображения в элементы ПО, а следовательно, и в элементы ИО системы;
- системоцентричное описание ПрО компонентов ИС в виде знаний, включающее не только структурные, но и динамические аспекты поведения объектов ПрО, а также знания о ситуациях их использования в системе [4].

Из сказанного выше следует, что онтология ПрО проектируемой ИС, разработанная в качестве решений по МО системы, при выполнении последующих этапов проектирования может быть трансформирована в решения по ИО и ПО системы, например, за счет отображения онтологии в UML-диаграмму классов ПО или схему данных ИС [5].

Поскольку в настоящее время существует множество методик, обеспечивающих взаимоднозначное соответствие элементов ИО и ПО (например, объектно-реляционное отображение), данная задача в работе рассматриваться не будет. В дальнейшем в работе будет принято предположение, что если онтология ПрО отражена в модель элементов ПО (диаграмму классов) или ИО (схему данных) системы, то синтез недостающей модели осуществляется с применением существующих методик [5]. При этом, исходя из особенностей системоцентричного описания онтологии, включающего описания множеств входных, выходных и хранимых в системе данных применительно к объектам ПрО, более предпочтительной является разработка методики отображения онтологии ПрО в схему данных ИС.

Исходя из сказанного выше задача исследования формулируется следующим образом: разработка и практическая апробация методики синтеза схемы данных информационной системы на основе онтологического описания предметной области.

Методика синтеза схемы данных информационной системы на основе онтологического описания предметной области

Предлагаемая в работе методика синтеза схемы данных ИС на основе онтологического описания ПрО включает следующие этапы:

- системоцентричный анализ ПрО объекта автоматизации;
- синтез онтологии ПрО на основе UML-моделей отдельных операций;
- синтез функциональной структуры проектируемой ИС;
- синтез схемы данных системы;
- расширение библиотеки готовых компонентов.

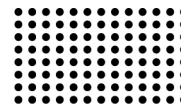
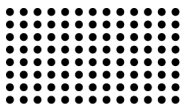
Использование онтологического описания ПрО для формального синтеза схемы данных ИС значительно повышает требования к качеству проведения предпроектного анализа объекта автоматизации и качеству сформированной в результате данного анализа онтологии.

Поэтому первым этапом предлагаемой методики является системоцентричный анализ ПрО объекта автоматизации. Проведение анализа ПрО проектируемой ИС предлагается осуществлять в несколько регламентированных шагов, основанных на комбинированном применении объектно-ориентированного и процессного подходов.

Шаг 1. С применением процессного подхода осуществляется анализ основных БП ПрО. Результатом данного анализа является набор IDEFO- или IDEF3-диаграмм, отражающих автоматизируемые БП с различных точек зрения их участников.

Шаг 2. Выделение БП, подлежащих автоматизации, на основе анализа описаний БП, созданных с различных точек зрения. Критерием выделения БП является получение в результате выполнения процесса данных, которые востребованы теми или иными их участниками, и должны быть сохранены в базе данных (БД) или получены из неё.

Шаг 3. В ходе объектно-ориентированного анализа ПрО осуществляется выделение терминов, которые максимально полно и комплексно характеризуют данную ПрО и отражают её существенные особенности. Результатом данного анализа является перечень тер-



минов, их определений и синонимов, приведенный, например, в табличном виде.

Шаг 4. В результате комбинированного применения процессного и объектно-ориентированного подходов из терминов, описывающих рассматриваемые процессы ПрО, формируются соответствующие автоматизируемым операциям БП деревья онтологий. Такие деревья онтологий состоят из групп терминов, которые непосредственно задействованы в конкретной операции.

Результатом такого анализа являются деревья онтологий, которые строятся для каждой операции, выделенной на IDEF0- или IDEF3-диаграммах. Деревья онтологий представляются в виде UML-диаграмм, где все термины являются объектами, имеющими как иерархические, так и горизонтальные связи. Поэтому каждая такая UML-диаграмма операции БП имеет вид, так называемого леса онтологий [6, 7].

В результате выполнения первого этапа данной методики формируется набор процессных и объектных описаний автоматизируемых БП ПрО, характеризующих её с различных точек зрения основных участников БП.

Данные описания взаимосвязаны только логикой протекания бизнес-процессов и во многом дублируются, что отражает ситуации использования одних и тех же объектов ПрО в различных операциях. Тем не менее, полноценной онтологической моделью предметной области такие описания еще не являются, что обуславливает необходимость осуществления второго этапа предложенной методики: синтеза онтологии ПрО на основе UML-моделей отдельных операций.

Основная сложность выполнения данного этапа 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 с точек зрения различных групп пользователей), имеет несколько различных названий. Набор атрибутов и их названий в различных ПрО также может отличаться. Поэтому синонимия атрибутов также имеет место. Вследствие этого данный этап должен осуществляться с привлечением экспертов в конкретной ПрО.

Кроме того, данный этап усложняется необходимостью максимизации повторного использования существующих компонентов ИС в случае наличия библиотеки выполненных проектов. Таким образом имеет

место наличие противоречивых требований: с одной стороны, при синтезе модели ПрО новые деревья онтологий должны быть максимально абстрагированы от её специфики (для максимизации возможностей их повторного использования в других ПрО), с другой стороны в эти деревья должно быть включено максимальное количество уже существующих узлов из библиотеки готовых компонентов, с третьей – синтезированная онтология должна в максимальной степени отражать специфику конкретной ПрО, для которой проектируется текущая система.

Поиск компромисса между данными требованиями возможен посредством выполнения следующих шагов:

Шаг 1. Синтез базового варианта онтологии ПрО проектируемой ИС на основе UML-моделей отдельных операций. Целью данного шага является объединение разрозненных UML-моделей объектов, участвующих в отдельных операциях БП, в одну общесистемную модель, установление между узлами иерархий горизонтальных ассоциативных связей и разрешение конфликтов синонимии.

Шаг 2. Сопоставление выделенных на предыдущем шаге узлов онтологии ПрО проектируемой ИС с узлами онтологий выполненных проектов. Применяется анализ подобия структур (атрибутов и методов) онтологий.

Если подобные узлы онтологии найдены – принимается решение о включении в онтологию текущей ПрО узла из библиотеки или установления между подобными онтологиями обобщающей связи (когда онтология текущей ПрО расширяет обобщающую онтологию из библиотеки).

Если подобные узлы онтологий не найдены – принимается решение, что данный узел является новым и уникальным.

Одним из способов снижения сложности проведения анализа на данном этапе предлагаемой методики является разработка специализированного CASE-средства, функциями которого является визуализация деревьев онтологий, автоматизированный анализ структур подобия узлов онтологий с использованием библиотеки описания выполненных проектов, формирование рекомендаций по их обобщению, автоматизация разрешения конфликтов синонимии и т.д.

Результатом данного этапа является общесистемная онтология ПрО для проектируемой ИС, использующая онтологии существующих компонентов из библиотеки готовых решений.

Наличие данной модели делает возможным осуществление третьего этапа предлагаемой методики: синтеза функциональной структуры проектируемой ИС.

Результаты анализа ПрО и синтеза её онтологии, описанные в виде совокупности процессных и объектных диаграмм, позволяют выделить функциональные задачи проектируемой системы.

Синтез функциональной структуры осуществляется "снизу вверх", начиная с масштаба функций функциональных задач.

Критерием разбиения ИС на отдельные функции является минимизация количества разрываемых связей между онтологиями, которые описывают отдельную функцию, и онтологиями других функций, с которыми она взаимосвязана.

Далее осуществляется анализ выделенных функций на предмет наличия ситуаций их совместного использования в БП объекта автоматизации, что позволяет объединить их в отдельные функциональные задачи. Аналогичным образом функциональные задачи объединяются в комплексы задач, а те в свою очередь – в подсистемы. Взаимосвязь между выделенными функциональными задачами, комплексами задач и подсистемами, как правило, визуализируется в виде комплекса DFD-диаграмм различного уровня системного масштаба.

Этот этап также подразумевает участие экспертов, потому что выделенный критерий не позволяет формализовать данные операции в полном объёме.

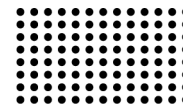
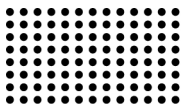
Наличие функциональной структуры ИС позволяет перейти к следующему четвёртому этапу методики: синтезу схемы данных системы. Основная сложность данного этапа заключается в осуществлении объектно-реляционного отображения (рассмотрим наиболее распространённый случай, когда описанные с помощью иерархий фреймов и UML-диаграмм онтологии отображаются в таблицы реляционной схемы данных БД). При этом целесообразно рассматривать синтез именно физической схемы данных, поскольку вместе с онтологиями будут повторно использованы и элементы схемы данных – таблицы, представления, последовательно-

сти, хранимые процедуры и т.д. В таком случае названия таблиц соответствуют узлам онтологий (фреймам), а атрибуты таблиц – атрибутам фреймов.

Основная сложность такого отображения заключается в различных математических аппаратах фреймовой модели представления знаний и реляционной алгебры, которая особенно сильно проявляется при описании связей объектов онтологии и сущностей БД.

Таким образом, обеспечение взаимнооднозначного отображения терминов ПрО в таблицы БД ИС требует применения соответствующей методики построения БД: структурно схема данных проектируется как набор взаимосвязанных витрин данных, в каждой из которых таблицей фактов является корневой элемент конкретной иерархии фреймов, а консольным таблицам и таблицам измерений соответствуют производные понятия ПрО. Каждая отдельная иерархия фреймов будет представлять собой многомерный куб, который может быть свёрнут к одному универсальному отношению, содержащему все множество возможных атрибутов конкретной иерархии понятий ПрО, что позволит осуществлять сравнительный анализ терминов различных ПрО (с целью определения совместимости ПрО и применимости существующих решений для новой ИС) на множестве характеризующих их атрибутов и избежать проблемы синонимии терминов.

Также при реализации объектно-реляционного отображения необходимо решить проблему реализации в ИО одного из основных принципов ООП – наследования. Для решения данной задачи предлагается выделение абстрактной обобщённой сильной сущности, универсальной для различных ПрО, атрибуты которой носят общий характер для любой ИС. Непосредственно при проектировании новой системы создаётся слабая сущность, которая полностью зависит от универсальной и дополняет её, поскольку содержит дополнительную информацию, характерную для конкретной ПрО разрабатываемой ИС. Данная слабая сущность в качестве ключевого атрибута (PK) содержит внешний ключ (FK), реализующий связь «1:1» с универсальным сильным типом сущности. В этом случае для упрощения работы с БД ИС со стороны разработчика ПО и конечного пользователя реализуются виртуальные таблицы – представления, которые содержат всю информацию об объекте ПрО, объединяя данные из нескольких таблиц. Исполь-



зование представлений позволяет скрыть разветвлённость схемы данных от пользователей и обеспечить соответствие логического представления данных конкретной ПрО.

Несмотря на то, что слабая сущность полностью зависит от сильной (записи в зависимой не могут существовать без соответствующих записей в независимой сущности), появляется необходимость контроля целостности данных на уровне двух таблиц. Поскольку ни целостность сущностей, ни ссылочная целостность в полной мере не обеспечивает решение данной задачи, предлагается использование триггеров. Триггеры реализуют сложные ограничения целостности данных, которые нереализуемы декларативными ограничениями, устанавливаемыми при создании таблиц.

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

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 – главной и зависимой (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).

Результатом данного этапа является синтезированная схема данных, которая включает в себя уже

реализованные элементы ИО ИС в случае наличия соответствующих им элементов онтологий ПрО в библиотеке готовых решений организации-разработчика.

Завершающим пятым этапом методики является расширение библиотеки готовых компонентов путем включения в глобальную онтологию библиотеки новых элементов, описывающих новые знания, добытые во время проектирования новой ИС.

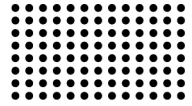
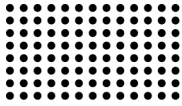
При этом над универсумом терминов множества ПрО, сохраненных в библиотеке, возможно выполнение операций расширения и очистки. Критерием создания новых терминов (удаления существующих) в иерархии терминов ПрО является наличие (отсутствие) горизонтальных понятийных связей (аналог отношения ассоциации в диаграмме классов) между анализируемым термином и остальными элементами онтологии.

Результатом данного этапа является расширенная библиотека знаний компании-разработчика о ПрО выполненных проектов.

Предложенная методика синтеза схемы данных информационной системы на основе онтологического описания ПрО, использующая объектно-реляционное отображение данных, уменьшает производительность системы в целом за счёт множества дополнительных сущностей и связей «1:1», которые не рекомендуется использовать в схеме БД. Но создание такой гибкой методики и универсальной библиотеки готовых компонентов позволяет облегчить разработчику процесс проектирования последующих ИС на основе ранее разработанных используемых в качестве прототипов, что значительно сокращает время разработки новых ИС. А 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.

ВЫВОДЫ

В работе предложена методика синтеза схемы данных информационной системы на основе онтологического описания ПрО. Данная методика основана на отображении в ИО ИС онтологии ПрО проектируемой системы. Её применение усложняет стадию анализа требований к ИС и повышает требования к качеству ведения проектной документации (формирование и



поддержание библиотек онтологий), но позволяет формально определить возможность повторного использования библиотеки существующих компонентов ИС, автоматизировать её расширение, решить задачи обновления версий и сопровождения множеств внедрений тиражируемых ИС, используемых в БП различных предприятий.

Перспективным направлением дальнейших научных исследований является создание математических моделей, формализующих выполнение этапов предложенной методики: синтеза онтологий процессов на основе анализа текстовых и визуальных описаний ПрО, анализа подобия узлов онтологий, синтеза онтологии ПрО на основе онтологий процессов, генерации схемы данных и т.д.

ЛИТЕРАТУРА:

1. GOST ISO/MEHК 15288–2005. Sistemnaja inzhenerija. Processy zhiznennogo cikla sistem. – Vved. 01–01–2007. – М. : Standartinform, 2006. – 57 s.
2. Sajjt ISO/IEC/IEEE 42010 Website [Elektronnyj resurs]. – Rezhim dostupa: <http://www.iso-architecture.org/ieee-1471/index.html>. – Zagolovok s ehkrana.
3. Arkhitektury, modeli i tekhnologii programmno obespechenija informacionno-upravljajushhikh sistem: monografija / Tkachuk N.V., Shekhovcov V.A., Kuklenko D.V., Sokol V.E. Pod red. M.D. Godlevskogo. – Khar'kov: NTU «KhPI», 2005. – 546 s.
4. Evlanov M.V. Tekhnologija bystrogo proektirovanija informacionnykh sistem / M. V. Evlanov, M. A. Kernosov, M.Eh. Lotfulina // Informacionnye sistemy i tekhnologii : Mezhdunarodnaja nauchno-tehnicheskaja konferencija, 22-29 sentjabrja 2012 g. : tezisy dokladov – Khar'kov : NTMT, 2012. – S. 35.
5. Lotfulina M.Eh. Tekhnologija razrabotki informacionnykh sistem posredstvom adaptacii gibkikh prototipov k predmetnoj oblasti ob'ekta avtomatizacii [Tekst] / M.Eh. Lotfulina // 16-ij Mehd. mol. forum «Radioelektronika i molodezh' v XXI veke». – Khar'kov: KhNUREh, 2012. – S. 180-181.
6. Maciaszek L.A. Requirements Analysis and System Design / L.A. Maciaszek: 2d ed. – Reading: Addison Wesley, Harlow England, 2005. – 504 p.
7. Booch G. Object-oriented Analysis and Design with Applications / G. Booch: 3d ed. – Reading: Addison-Wesley, 2007. – 693 p.