

Д.И. Харитонов<sup>1</sup>, Г.В. Тарасов<sup>2</sup>, Д.В. Леонтьев<sup>3</sup>, Р.В. Парахин<sup>4</sup>

Институт автоматизики и процессов управления ДВО РАН  
Владивосток. Россия

## Моделирование предметной области для формирования электронных коллекций\*

В настоящей статье рассматривается проблема переноса экспериментальных данных, накапливаемых исследовательскими группами, в цифровое представление. Предложен подход, направленный на глубокую автоматизацию построения информационной системы на основании модели предметной области. Для этого подхода формализован аппарат построения моделей предметных областей, разделяющий модель на шаблонный граф и древовидное описание. Причём шаблонный граф определён так, что его можно рассматривать как дерево, дополненное «избыточными» связями, и для каждого узла дерева определены роль, значение и приоритет. Шаблонный граф используется как средство контроля структуры описания, а также как средство, обеспечивающее взаимодействие между моделью предметной области и компонентой генератора программного кода. Между структурой древовидного описания и шаблонным графом существует отображение, ставящее в соответствие каждому узлу и каждому ребру структуры один узел и одно ребро шаблонного графа. В статье приведён вариант шаблонного графа для построения ER-моделей предметных областей, оперирующий такими понятиями, как классификация, коллекция, объекты коллекции, атрибуты, наблюдения, интерфейсы. Каждому интерфейсу в шаблонном графе соответствует параметризованная процедура преобразования данных, позволяющая взаимодействовать с базой данных, файлами, сетевыми источниками данных и другими информационными системами. Приведён пример ER-модели данных для построения информационной системы фораминиферов – раковинных одноклеточных организмов из группы протистов. Данный пример позволяет сформировать минимальный набор компонент, необходимый для построения информационной системы, состоящий из базы данных, подсистемы ввода данных из файлов и подсистемы построения карт концентраций.

**Ключевые слова и словосочетания:** информационные системы, базы данных, семантическое моделирование, ER-моделирование, моделирование данных, объектное моделирование, моделирование сущностей, сущности, связи.

\* Работа выполнена при поддержке комплексной программы фундаментальных исследований ДВО РАН «Дальний Восток», проект 18-5-104.

<sup>1</sup> Харитонов Дмитрий Иванович – [orcid.org/0000-0003-3359-2383](https://orcid.org/0000-0003-3359-2383), канд. техн. наук, старший научный сотрудник; Федеральное государственное бюджетное учреждение науки Институт автоматизики и процессов управления Дальневосточного отделения Российской академии наук; e-mail: [demiurg@dvo.ru](mailto:demiurg@dvo.ru).

<sup>2</sup> Тарасов Георгий Витальевич – [orcid.org/0000-0001-8855-7388](https://orcid.org/0000-0001-8855-7388), научный сотрудник; Федеральное государственное бюджетное учреждение науки Институт автоматизики и процессов управления Дальневосточного отделения Российской академии наук; e-mail: [george@dvo.ru](mailto:george@dvo.ru).

<sup>3</sup> Леонтьев Денис Васильевич – [orcid.org/0000-0002-5116-3008](https://orcid.org/0000-0002-5116-3008), аспирант; Федеральное государственное бюджетное учреждение науки Институт автоматизики и процессов управления Дальневосточного отделения Российской академии наук; e-mail: [devozh@dvo.ru](mailto:devozh@dvo.ru).

<sup>4</sup> Парахин Роман Валерьевич – [orcid.org/0000-0001-8192-5600](https://orcid.org/0000-0001-8192-5600), инженер-программист; Федеральное государственное бюджетное учреждение науки Институт автоматизики и процессов управления Дальневосточного отделения Российской академии наук; e-mail: [fadak@dvo.ru](mailto:fadak@dvo.ru).

## Modeling the domain for the formation of electronic collections

In this paper, the problem of transferring experimental data accumulated by research groups into a digital representation is considered. An approach directed on deep automation of information systems construction on the basis of domain model is offered. For this approach, the tool for constructing domain models is formalized, that divides a model into a template graph and a tree-like description. The template graph is defined in such a way that it can be considered as a tree, supplemented by "redundant" links, and for each node in the tree, the role, value and priority are defined. The template graph is used as a means of controlling the structure of the description, and also as a means procuring interactions between a domain model and the code generator component. Between the structure of a tree-like description and a template graph there is a mapping that assigns to a node and an edge of the structure one node and one edge of the template graph. The paper presents an example of the template graph for constructing ER-models of subject domains, operating with such concepts as classification, collection, collection objects, attributes, observations and interfaces. Each interface in the template graph corresponds to a parameterized data conversion procedure designed to interact with the database, with files, with network data sources and other information systems. An example of an ER data model for constructing an information system of foraminifers - shell unicellular organisms from a phylum or class of amoeboid protists is given. The example allows to form a minimal set of components necessary for building an information system, consisting of a database, a subsystem for data import from files, and a subsystem for constructing concentration maps.

**Keywords:** information systems, databases, semantic modeling, ER-modeling, data modeling, object modeling, entity modeling, entities, relationships.

В современном мире цифровые технологии проникли практически во все аспекты деятельности человека, в том числе и научную деятельность. В рамках научных исследований собираются колоссальные объемы разнообразной информации, которую необходимо хранить и обрабатывать [6]. Учитывая, что большинство современных исследований проводятся посредством моделирования на компьютере, важно иметь исходные данные в цифровом виде. Для хранения собранной информации разработано множество информационных систем и баз данных. Так, в сети Интернет представлены разнообразные коллекции данных, например, базы данных научных журналов (Web of Science, Scopus), открытые образовательные курсы (Intuit), базы данных по фармакологии (Cochrane Library [3]), каталоги биоразнообразия (FishBase [7]), энциклопедии элементов ДНК (ENCODE [8]), химические базы данных (Protein Data Bank [10], PubChem [11], ChemSpider [12]) и другие. Разнообразие содержания баз данных диктует необходимость в хранении и представлении данных в разных структурах и форматах. Многие из вышеперечисленных баз данных решают четко определенный круг задач и фактически являются узкоспециализированными. Например, химическая база данных Protein Data Bank поддерживается сообществом пользователей уже в течение более двадцати лет. Это отличный инструмент, содержащий банк данных трёхмерных структур белков и нуклеиновых кислот. Информация, полученная методами рентгеновской кристаллографии, ЯМР-спектроскопии, криоэлектронной микроскопии, вносится в базу данных биологами и биохимиками всего мира.

При этом большинство химических экспериментов уже проводятся численно, с использованием такого научного инструмента, как вычислительные кластеры. Вполне очевидно, что для развития методов обработки данных на кластере необходим инструмент, посредством которого данные могут быть получены из информационной системы, обработаны и возвращены обратно. Однако современные информационные системы пока ещё не предоставляют сервис, обеспечивающий непрерывный цикл обработки данных.

С другой стороны, для длительного существования электронных коллекций необходимо поддерживать данные в актуальном виде [15]. Если же данные не использовать в научных исследованиях и не актуализировать, они будут лежать «мертвым» грузом, устареют и потеряют свою значимость. Поэтому необходимы средства, позволяющие использовать эти данные в ежедневной деятельности, например, посредством предоставления сервиса по обработке данных, их преобразованию, совмещению, наложению, визуализации и т.д. Необходимо отметить, что для публикаций и презентаций, анализа и сравнения данных существуют различные варианты их представления [1], такие, как таблицы со статистическими расчетами; двумерные и трехмерные графики, гистограммы и диаграммы; деревья и графы; проекция данных на поверхность визуализации в виде раскраски или нормалей, отображение данных на картах, анимация изменения данных, аудиоанимация данных и т.д.

Авторы считают, что построение универсальных информационных систем, обладающих широким спектром инструментов доступа к данным, обработки и визуализации данных, является актуальной задачей. При этом основу универсальности системы может составлять модель предметной области электронной коллекции [2]. Изложение в настоящей статье построено следующим образом. Сначала описывается общий подход к автоматизации построения информационных систем электронных коллекций. Затем представляется формальное описание компоненты системы, ответственной за универсальность ER-моделей электронных коллекций. Далее приводятся аппарат описания ER-моделей и пример модели предметной области электронной коллекции данных фораминиферов. В конце статьи дано сравнение предложенного подхода с известными авторам аналогами.

### **Концепция построения системы**

Предлагаемый в настоящей статье подход к построению электронных коллекций научных данных базируется на автоматическом построении значительного объёма исходного кода информационной системы. В качестве исходных данных для такого построения используется ER-модель предметной области, задаваемая дизайнером информационной системы, исходя из имеющихся в наличии данных и информации о планируемом развитии системы. ER-модель может быть задана либо на внутреннем XML-подобном языке, либо с применением жёстко закодированного WEB-интерфейса редактирования данных, организованных в СУБД PostgreSQL в виде так называемого «программируемого справочника» (рис. 1).

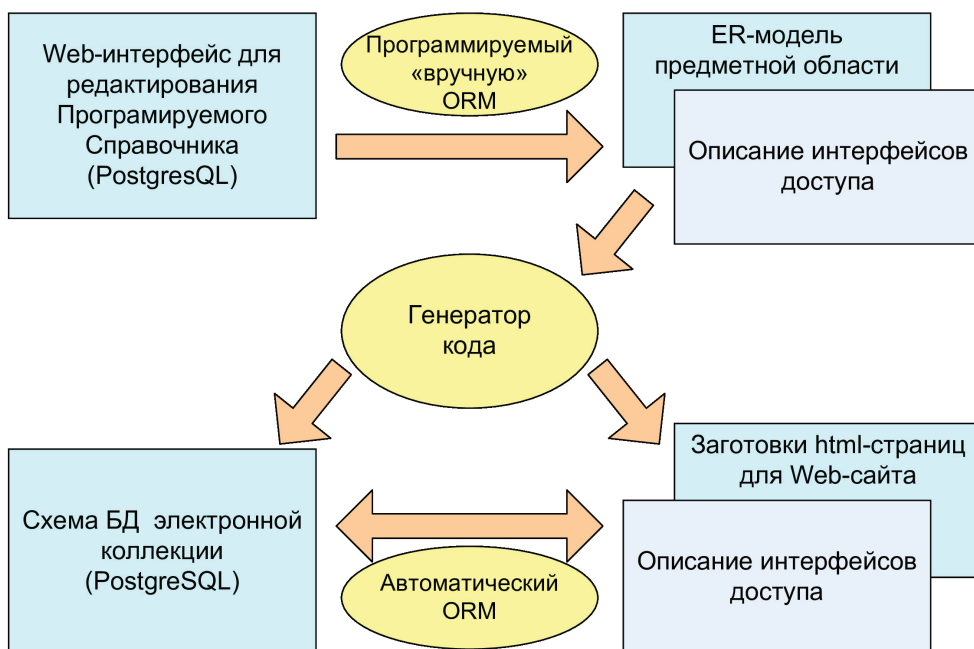


Рис. 1. Концептуальная схема автоматизации построения электронных коллекций

Сформированная модель предметной области передаётся подсистеме генерации программного кода, формирующей три части информационной системы. Часть первая – это представление данных тематической коллекции в базе данных, состоит из определения таблиц, полей, индексов и взаимосвязей между таблицами. Вторая часть – множество заготовок html-страниц, состоящее из шаблонов отображения списков выборок данных, форм для редактирования отдельных записей электронных коллекций и поисковых форм. Третья часть отвечает за подготовку данных для интерфейсов доступа к электронной коллекции.

### Программируемый справочник

В предложенной концепции построения информационной системы особую роль играет компонент системы, названный «программируемым справочником». Задачей этого компонента является формирование такого описания модели предметной области, на основании которого возможна вся дальнейшая работа по автоматизации процесса генерации кода. Для этого программируемый справочник спроектирован как максимально универсальное средство подготовки древовидных описаний с контролем структуры формируемых данных. Рассмотрим процесс подготовки описаний с формальной точки зрения на основе алгебры множеств.

Для этого нам потребуются следующие обозначения. Множество всех конечных последовательностей, составленное из символов множества  $A$ , включая пустую строку  $\varepsilon$ , будет обозначаться как  $A^*$ . Конкатенация двух строк  $u, w \in A^*$  записывается как  $uw$ . Обозначим множество  $A$ , дополненное символом  $\varepsilon$ , как  $\bar{A}$ . Множество всех подмножеств множества  $A$  будет обозначено как  $\mathcal{P}(A)$ . Везде, где не оговорено

явно, будем считать функции, определённые на множестве  $A$  (например,  $f$ ), определёнными также и на множествах  $\dot{A}, \mathcal{P}(A), A^*$ . Причём значением любой функции для элемента  $\varepsilon$  является  $\varepsilon$ . Значением функции от последовательности элементов является последовательность значений ( $\forall \xi \in A^* \Rightarrow f(\xi) = \{x_i = f(a_i) | a_i \in \xi\}$ ). А значением функции от подмножества элементов является множество значений функции от элементов подмножества ( $\forall \xi \in A^* \Rightarrow f(\xi) = \{x_i = f(a_i) | a_i \in \xi\}$ ).

Будем считать заданными следующие множества:  $\mathbf{P}$  – множество параметров,  $\mathbf{R} \equiv \{req, opt, list, choice\}$  – множество ролей,  $\mathbf{D}$  – пространство значений параметров, а также некоторый алфавит  $\mathbf{A}$ .

Данные в программируемом справочнике разделяются на шаблоны и конечные структуры. Шаблоны определяются следующим образом.

### Определение 1. Шаблонный граф

Шаблонный граф  $G$  определяется как набор  $\langle V_g, E_g, E'_g, g_0, f \rangle$ , где

1.  $V_g$  – множество вершин графа с выделенной корневой вершиной  $g_0 \in V_g$ ;
2.  $f: V_g \rightarrow \mathbf{P} \times \mathbf{D} \times \mathcal{P}(\mathbf{R}) \times \mathbb{N}$  – функция, определяющая переменную узла, значение переменной, роль переменной и приоритет узла:  $\forall g \in V_g \Rightarrow f(g) = \langle f_p(g), f_d(g), f_r(g), f_n(g) \rangle$ ;
3.  $E_g \subseteq V_g \times V_g$  – множество направленных рёбер графа, на котором определяется функция инцидентности  $()^\bullet: V_g \rightarrow V_g^*$ , так что  $\forall g \in V_g: g^\bullet = \{g_i | i \in [1..m], m \in \mathbb{N}, (g, g_i) \in E_g, \text{ причём } \forall j < k \in [1..m] \Rightarrow g_j \neq g_k, f_n(g_j) \leq f_n(g_k)\}$ ;
4.  $E'_g \subset E_g$  – покрывающее дерево графа  $\forall g \in V_g, g \neq g_0 \Rightarrow \exists! \{g_i \in V_g | i \in [0..n], n \in \mathbb{N}, g_n = g, \forall i \in [1..n] \Rightarrow (g_{i-1}, g_i) \in E'_g, \text{ причём } \forall (g, g') \in E_g, f_r(g') = req \Rightarrow (g, g') \in E'_g\}$ .

(1)

Последнее условие в определении говорит о том, что для любой вершины существует единственный выделенный путь до корневой вершины. Определённый таким образом шаблонный граф можно рассматривать как дерево, дополненное «избыточными» связями, причём для каждого узла дерева определены роль, значение и приоритет. Далее будет показано, как они используются для построения древовидного описания.

### Определение 2. Древовидное описание на основе шаблонного графа

Древовидное описание  $T$  определяется как набор  $\langle G_g, V_t, E_t, v_0, t \rangle$ , где

1.  $G_g = \langle V_g, E_g, E'_g, v_g, f \rangle$  – шаблонный граф параметров файла;
2.  $V_t$  – множество вершин дерева с корневой вершиной  $v_0 \in V_t$ ;
3.  $t: V_t \rightarrow V_g \times \mathbf{D} \times \mathbb{N}$  – функция, определяющая прототип узла в шаблоне, значение переменной и приоритет узла:  $\forall v \in V_t \Rightarrow t(v) = \langle \tau(v), t_d(v), t_n(v) \rangle$ ;
4.  $E_t \subseteq V_t \times V_t$  – множество направленных дуг дерева, причём:
  - для всех вершин определена функция инцидентности  $()^\bullet: V_t \rightarrow V_t^*$ , так что  $\forall v \in V_t: v^\bullet = \{v_i | i \in [1..m], m \in \mathbb{N}, (v, v_i) \in E_t, \text{ и } \forall j < k \in [1..m] \Rightarrow v_j \neq v_k, (f_n(v_j) < f_n(v_k)) \vee (f_n(v_j) = f_n(v_k)) \wedge (t_n(v_j) \leq t_n(v_k))\}$ ;

- для любой вершины существует единственный путь до корневой вершины:  
 $\forall v \in V_t, v \neq v_0 \Rightarrow \exists! \{v_i \in V_t \mid v_n = v, 0 < i \leq n \in \mathbb{N} \Rightarrow (v_i, v_{i-1}) \in E_t\} \equiv rt(v)$ ;
- плюс каждому ребру дерева параметров сопоставляется одно ребро в шаблонном графе параметров:  $\forall (v_1, v_2) \in E_t \Rightarrow (\tau(v_1), \tau(v_2)) \in E_g$ .

(2)

Расширим область определения функций  $f$  на  $V_t$ :

$$\forall v \in V_t \Rightarrow f(\tau(v)) = \langle f_p(\tau(v)), f_a(\tau(v)), f_r(\tau(v)), f_n(\tau(v)) \rangle.$$

Определим для узлов дерева функцию  $S_p: V_t \times V_g \rightarrow V_t^*$ , возвращающую последовательность узлов, связанных с данным и имеющих заданный прототип:

$$S_p(v, v_g) \equiv \{v_i \in V_t \mid i \in [0..n], n \in \mathbb{N}, (v, v_i) \in E_t, \tau(v_i) = v_g, \forall i > 0 \Rightarrow f_n(v_i) \geq f_n(v_{i-1})\}$$

Определим также для узлов дерева функцию  $S_r: V_t \times R \rightarrow \mathcal{P}(V_t)$ , возвращающую множество узлов, связанных с данным и имеющих заданную роль:

$$S_r(v, r) \equiv \{v' \in V_t \mid (v, v') \in E_t, r \in f_r(v')\}.$$

Тогда дерево параметров файла будем считать корректным, если:

$$\forall v \in V_t, v_g \in \tau(v)^*, req \in f_r(v_g) : \|S_p(v, v_g)\| \geq 1$$

$$\forall v \in V_t, v_g \in \tau(v)^*, lst \notin f_r(v_g) : \|S_p(v, v_g)\| \leq 1$$

$$\forall v \in V_t, v_g \in \tau(v)^*, choice \in f_r(v_g) : \|S_r(v, choice)\| = 1$$

Таким образом, в корректном дереве все узлы имеют свои обязательные (req) параметры. Каждый узел имеет не более одного значения опциональных (opt) параметров, только один из альтернативных (choice) параметров и любое количество списковых (lst) параметров.

Между структурой древовидного описания и шаблонным графом существует отображение, ставящее в соответствие каждому узлу и каждому ребру структуры один узел и одно ребро шаблонного графа, что позволяет определить алгоритм редактирования древовидного описания следующим образом:

1. На первом шаге добавляется корневая вершина описания.
2. Выполняется цикл, в котором для каждой новой вершины древовидного описания:
  - а. добавляются обязательные (обозначенные в шаблонном графе) узлы;
  - б. пользователю предоставляется возможность выбрать и добавить необходимые параметры из альтернативных, опциональных и списковых;
  - в. пользователю предоставляется возможность заполнить значения новых параметров.
3. Цикл редактирования повторяется со второго шага.

### Шаблонный граф ER-моделей предметной области

Аппарат описания предметной области записан в программируемом справочнике в виде шаблонного графа. Универсальность этого аппарата обеспечивается однозначным сопоставлением каждому узлу шаблонного графа класса на языке Python в компоненте системы, отвечающей за автоматическую генерацию кода.



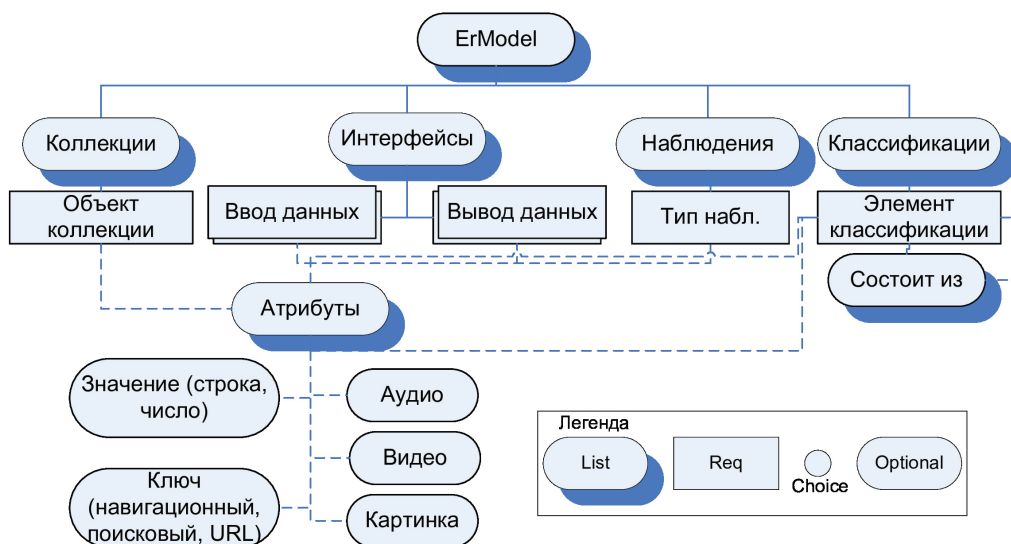


Рис. 2. Шаблонный граф для построения ER-моделей

На рисунке 2 изображён шаблонный граф, используемый для описания ER-моделей электронных коллекций научных данных. Он имеет следующие основные вершины, необходимые для автоматического построения информационной системы:

- Коллекция и объект коллекции – основные элементы, вокруг которых строится информационная система.
- Атрибуты – минимальный элемент, с которым может работать информационная система, размещаемый в базе данных либо в полях собственно таблицы объекта коллекции либо выносимый в отдельные таблицы для ускорения обработки данных.
- Классификация и элемент классификации – единственная задаваемая уже на момент формирования информационной системы, редко изменяемая информация, используемая для кластеризации данных коллекций.
- Наблюдения – тип вершины, предназначенный для описания собираемых в ходе экспериментов, экспедиций и опытов данных, непосредственно связанных с объектами электронных коллекций.
- Интерфейсы – предназначены для описания механизмов преобразования данных на этапах ввода информации в базу данных и на этапе формирования. Каждому интерфейсу в шаблонном графе соответствует параметризованная процедура преобразования данных, позволяющая взаимодействовать с базой данных, файлами, сетевыми источниками данных и другими информационными системами.

### Пример модели предметной области электронной коллекции данных фораминиферов

Рассмотрим в качестве примера ER-модель данных для построения информационной системы фораминиферов – раковинных одноклеточных организмов

из группы протистов. Известно, что в настоящее время существует около 10 000 современных видов [13], в то время как ископаемых видов фораминифер было более 40 000 видов [14]. Фораминиферы распространены повсеместно, во всех широтах и на всех глубинах, начиная от прибрежной зоны и заканчивая самыми глубокими впадинами.

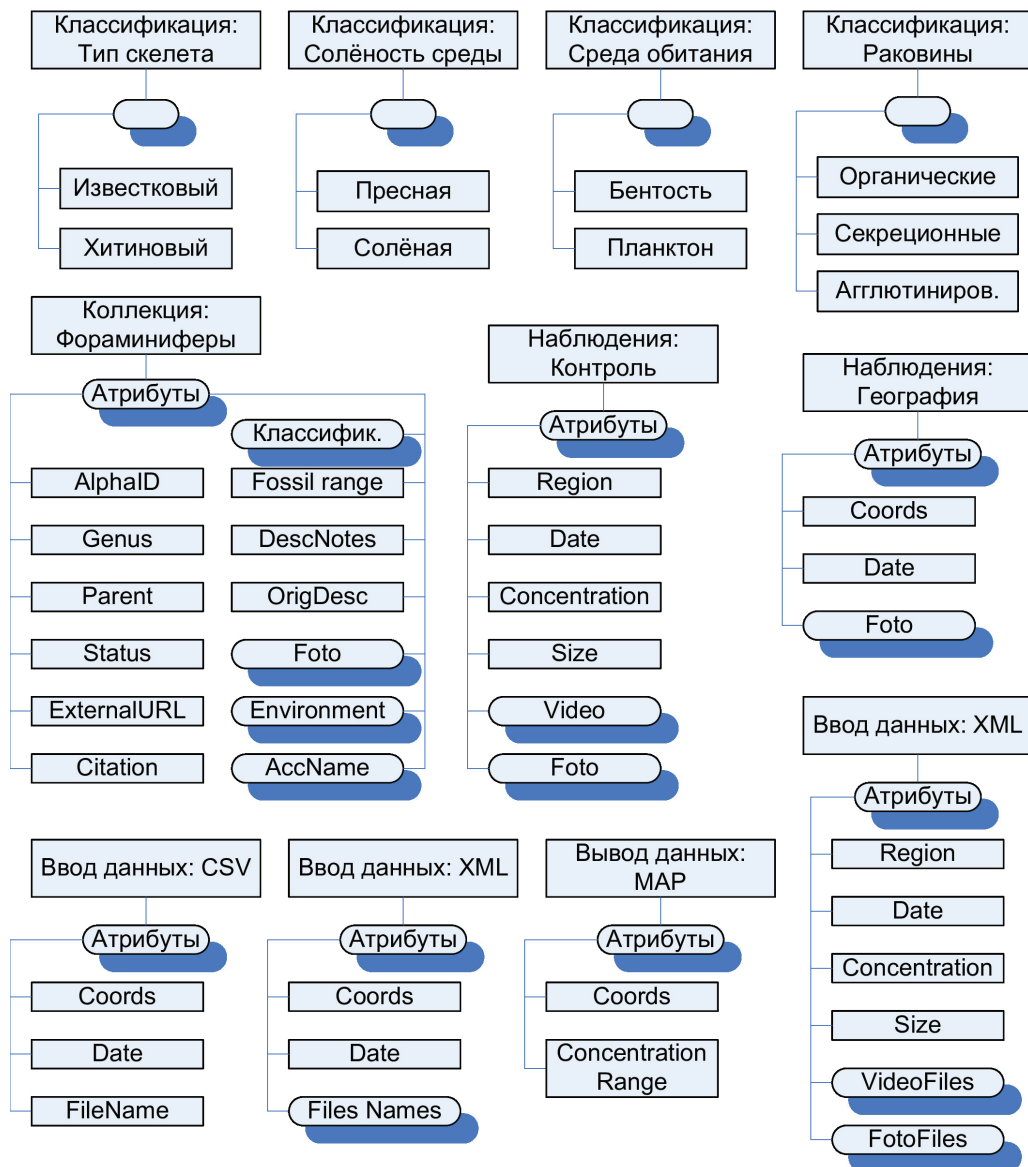


Рис. 3. ER-модель предметной области для коллекции фораминиферов

Отличительной особенностью фораминиферов является наличие раковины – защитного образования, которое окружает тело животного снаружи. Как ископаемые благодаря своим малым размерам и раковинам они могут быть обнаружены



даже в очень небольших количествах горной породы. Классификация и изучение эволюции этих простейших важны хотя бы потому, что по остаткам организмов в горных породах можно определить геологический возраст этих пород. Для разных геологических периодов истории Земли характерны свои виды, роды и семейства фораминиферов, те есть они служат весьма точными геологическими часами.

На рисунке 3 изображён пример ER-модели предметной области коллекции фораминиферов, построенный с использованием описанного в предыдущем параграфе шаблонного графа. Для упрощения представления не отображены корневой уровень и уровень списков коллекций, классификаций, наблюдений и интерфейсов.

На рисунке представлены четыре простейшие классификации фораминиферов по типу скелета, среде обитания и её солёности, а также по способу образования раковин. В модели используется только один объект коллекционирования – фораминиферы, атрибуты которых по большей части совпадают с общепринятой в мире таксономией. В модели представлены два вида наблюдений, используемые для размещения фотографий фораминиферов на географической карте и для сохранения истории измерений концентрации различных видов в различных географических регионах. В качестве интерфейсов ввода данных описан импорт наблюдений первого типа из csv-файлов, в которых записаны ссылки на дополнительные файлы с фотографиями по одной ссылке на наблюдение, а также импорт из XML-файлов обоих видов наблюдений, содержащих списки ссылок на файлы с фотографиями или видеофайлы. Единственный интерфейс вывода данных рассчитан на построение карты концентрации видов фораминиферов по регионам.

Приведённый пример позволяет сформировать минимальный набор компонент, необходимый для построения информационной системы, состоящий из базы данных, подсистемы ввода данных из файлов и подсистемы построения карт концентраций. Не стоит рассматривать приведённый пример ER-модели как полностью законченный, так как, во-первых, и списки наблюдений, и списки интерфейсов могут быть существенно дополнены, а, во-вторых, после дополнений структура ER-модели должна быть оптимизирована.

### **Сравнение с другими подходами**

Наиболее традиционным подходом для разработки информационных систем является применение ER-моделирования на этапе проектирования базы данных. Такое проектирование было разработано и опубликовано Питером Ченом ещё в 1976 году [5, 16]. Задача ER-моделирования заключается в том, чтобы предоставить как можно больше сведений об информации, хранящейся в базе данных, придать этой информации больше смысла. Построение ER-модели является наиболее простым и прямолинейным подходом к построению той или иной базы данных. При этом после разработки модели любые изменения требуют особого внимания для корректного преобразования данных в соответствии с новой схемой размещения данных. Другими словами, этот подход ориентирован на разработку жёстко регламентированных систем, для поддержания работы которых необходимо внимание квалифицированного персонала.

Развитием идей ER-моделирования в направлении проектирования программных систем стало появление языка UML [3, 4] и программных средств, ориентированных на использование этого языка. При проектировании с использованием UML для отображения системной архитектуры применяются диаграммы, создаваемые для отображения поведения и структуры системы. К этим диаграммам относятся:

- диаграммы деятельности (процессы, подлежащие автоматизации);
- диаграмма вариантов использования;
- диаграммы последовательностей (операции, упорядоченные во времени);
- диаграммы коопераций (потоки данных между участниками);
- диаграммы состояний (изменение состояний и переходы между состояниями);
- диаграммы классов (классы, интерфейсы, кооперации);
- диаграммы компонентов (модули реализации);
- диаграмма развёртывания (узлов и соединений).

Как видно из набора диаграмм, UML-язык выглядит и часто критикуется как неоправданно большой и сложный. Он включает много избыточных или практически не используемых диаграмм и конструкций. Кроме того, на практике неточная, фактически неформальная семантика UML может приводить к несовместимости программируемых инструментов из-за различного трактования спецификаций.

В последнее время для проектирования информационных систем начали применять OWL (язык описания онтологий для WEB) для представления знаний о предметной области [17, 18]. В сравнении с реляционными базами данных онтология более гибка в использовании и обеспечивают более высокую выразительность для моделирования сложных отношений. Однако разработчику информационной системы не хватает адекватных методологий, методов и инструментов, которые позволили бы ему развивать и поддерживать систему, основанную на OWL-онтологиях. Это относится, в частности, к программному доступу к онтологиям OWL, включая синхронизацию объектно-ориентированной модели с онтологией, к управлению жизненным циклом онтологии, а также к построению эффективных запросов на основании онтологий.

Существующие системы автоматической генерации кода позволяют формировать исходный код на основании моделей предметной области, сниппетов и шаблонов. В дополнение к реализованным идеям предложенный в настоящей статье подход использует в качестве носителя описания модели базу данных, в качестве языка описания более абстрактный, чем UML или OWL, механизм ER-моделей, что позволяет расширить идеологию автоматизации в направлении учёта и проверки корректности изменений.

### **Заключение**

В статье рассмотрен подход к моделированию предметных областей электронных коллекций, ориентированный на автоматизацию построения информационных систем. Под автоматизацией в данном случае подразумевается генерация программного кода, реализующего формирование схемы базы данных, объектно-реляционного отображения между базой данных и подсистемами обработки данных, а также

процедур ввода и вывода информации. Модель предметной области в этом подходе размещается в базе данных, при этом предполагается, что она может изменяться в процессе разработки и эксплуатации электронной коллекции. Поэтому одной из ближайших задач является разработка методов и алгоритмов эволюции ER-модели, включающая преобразование данных электронных коллекций при переходе от одной ER-модели к другой, построенной на основе первой.

---

1. Джосан О.В. О визуализации научных данных для высокопроизводительных параллельных приложений // Тезисы конференции ПАВТ. Россия. Н. Новгород, 2009. С. 449–456.
2. Антонов И.В. Модель онтологии предметной области для систем семантически-ориентированного доступа // Электротехника. 2011. № 12. С. 339–343.
3. Мантурова И.А. Использование универсального языка моделирования (UML) при проектировании сложных информационных систем // Проблемы управления в социально-экономических и технических системах: матер. Всерос. науч. конф.: сб. науч. стат. 2012. С. 23–25.
4. Мацяшек Л.А. Анализ и проектирование информационных систем с помощью UML 2.0. Изд. Вильямс, 2008. 816 с.
5. Иванов К.К. ER-моделирование. Особенности семантического моделирования // Молодой ученый. 2017. № 19. С. 24–26.
6. Rahat Iqbal, Faiyaz Doctor, Brian More, Shahid Mahmud, Usman Yousuf. Big Data analytics and Computational Intelligence for Cyber–Physical Systems: Recent trends and state of the art applications // *Future Generation Computer Systems*, 2017. DOI: <https://doi.org/10.1016/j.future.2017.10.021>.
7. Mike Clarke. The Cochrane Collaboration and The Cochrane Library // *Otolaryngology, Head and Neck Surgery*, 2007, Vol. 137, Iss. 4, pp. 52–54. DOI: <https://doi.org/10.1016/j.otohns.2007.05.050>.
8. Liza Q. Agustin, Ma Lourdes D. Palomares, Graham C. Mair. Fishbase: a repository of genetic information on fish // *Aquaculture*, 1995, Vol. 137, Iss. 1–4, p. 46. DOI: [https://doi.org/10.1016/0044-8486\(96\)83518-6](https://doi.org/10.1016/0044-8486(96)83518-6).
9. Hongzhu Qu, Xiangdong Fang. A Brief Review on the Human Encyclopedia of DNA Elements (ENCODE) Project // *Genomics, Proteomics & Bioinformatics*, 2013, Vol. 11, Iss. 3, pp. 135–141. DOI: <https://doi.org/10.1016/j.gpb.2013.05.001>.
10. Enrique E. Abola, Joel L. Sussman, Jaime Prilusky, Nancy O. Manning. Protein data bank archives of three-dimensional macromolecular structures // *Methods in Enzymology*, 1997, Vol. 277, pp. 556–571. DOI: [https://doi.org/10.1016/S0076-6879\(97\)77031-9](https://doi.org/10.1016/S0076-6879(97)77031-9).
11. Evan E. Bolton, Yanli Wang, Paul A. Thiessen, Stephen H. Bryant. PubChem: Integrated Platform of Small Molecules and Biological Activities // *Annual Reports in Computational Chemistry*, 2008, Vol. 4, pp. 217–241. DOI: [https://doi.org/10.1016/S1574-1400\(08\)00012-1](https://doi.org/10.1016/S1574-1400(08)00012-1).
12. Sunghwan Kim. Public Chemical Databases // *In Reference Module in Life Sciences*, 2018. DOI: <https://doi.org/10.1016/B978-0-12-809633-8.20192-1>.
13. Ald, S.M. et al. Diversity, Nomenclature, and Taxonomy of Protists // *Syst. Biol.*, 2007, Vol. 56(4), pp. 684–689. DOI: [10.1080/10635150701494127](https://doi.org/10.1080/10635150701494127).

14. Pawlowski J., Lejzerowicz F., & Esling, P. Next-generation environmental diversity surveys of foraminifera: preparing the future // *The Biological Bulletin*, 2014, 227(2), pp. 93–106.
15. Bugbee Kaylin, Ramachandran Rahul, Maskey Manil, Gatlin Patrick. The art and science of data curation: Lessons learned from constructing a virtual collection // *Computers & Geosciences*, 2017, 112. DOI: 10.1016/j.cageo.2017.11.021.
16. Chen Peter. The Entity-Relationship Model – Toward a Unified View of Data // *ACM Transactions on Database Systems*, 1976, 1(1): 9–36. DOI: 10.1145/320434.320440.
17. Cerans Karlis, Romāne Aiga. OBIS: Ontology-Based Information System Framework // *CAiSE Forum*, 2015.
18. Kamran Munir, M. Sheraz Anjum. The use of ontologies for effective knowledge modelling and information retrieval // *Applied Computing and Informatics*, Vol. 14, Issue 2, 2018, pp. 116–126.

### Транслитерация

1. Dzhosan O.V. O vizualizatsii nauchnykh dannykh dlya vysokoproizvoditel'nykh parallel'nykh prilozhenii, *Tezisy konferentsii PAVT*, Rossiya, N.Novgorod, 2009, pp. 449–456.
2. Antonov I.V. Model' ontologii predmetnoi oblasti dlya sistem semanticheski-orientirovannogo dostupa, *Elektrotehnika*, 2011, No 12, pp. 339–343.
3. Manturova I.A. Ispol'zovanie universal'nogo yazyka modelirovaniya (UML) pri proektirovanii slozhnykh informatsionnykh sistem, *Problemy upravleniya v sotsial'no-ekonomicheskikh i tekhnicheskikh sistemakh: mater. Vseros. nauch. konf.: sb. nauch. stat.*, 2012, pp. 23–25.
4. Matsyashek L.A. Analiz i proektirovanie informatsionnykh sistem s pomoshch'yu UML 2.0. Izd. Vil'yams, 2008, 816 p.
5. Ivanov K.K. ER-modelirovanie. Osobennosti semanticheskogo modelirovaniya, *Molodoi uchenyi*, 2017, No 19, pp. 24–26.

© Д.И. Харитонов, 2018

© Г.В. Тарасов, 2018

© Д.В. Леонтьев, 2018

© Р.В. Парахин, 2018

**Для цитирования:** Моделирование предметной области для формирования электронных коллекций / Д.И. Харитонов, Г.В. Тарасов, Д.В. Леонтьев, Р.В. Парахин // Территория новых возможностей. Вестник Владивостокского государственного университета экономики и сервиса. 2018. Т. 10. № 2. С. 125–136.

For citation: Modeling the domain for the formation of electronic collections / Kharitonov D.I., Tarasov G.V., Leontiev D.V., Parahin R.V., *The Territory of New Opportunities. The Herald of Vladivostok State University of Economics and Service*, 2018, Vol. 10, No 2, pp. 125–136.

DOI dx.doi.org/10.24866/VVSU/2073-3984/2018-2/125-136

Дата поступления: 26.06.2018.