Литература

1. Исследование закономерностей и тенденций развития самоорганизующихся систем на примере веб-пространства и биологических сообществ. – [Электронный ресурс]. Режим доступа: http://web.iis.nsk.su/about (дата обращения: 14.02.2016).
2. *Хакен Г.* Синергетика. – М.: Мир, 1980. 404 с.
3. Бас*ин М.А., Шилович И.И.* Синергетика и INTERNET. – СПб.: Наука, 1999. 71 с.
4. Аутопойезис [Электронный ресурс]. // Национальная социологическая энциклопедия. – [Электронный ресурс]. URL: http://voluntary.ru/dictionary/591/word/autopoiezis (дата обращения: 14.02.2016).
5. *Матурана Умберто Р.* Древо познания: биологические корни человеческого понимания. – М.: Прогресс-Традиция, 2001. 223 с.
6. *Лавренчук Е.А.* Аутопойезис социальных сетей в интернет-пространстве: автореф. дисс. ... кандидата философских наук: 09.00.11. – М., 2011. 20 с.
7. *Лавренчук Е.А.* Аутопойезис. – [Электронный ресурс]. URL: http://vox-journal.org/content/Vox11-Lavrenchuk-Au.pdf (дата обращения: 14.02.2016).
8. *Bates M.* The Design of Browsing and Berrypicking Techniques for the Online Search Interface // Online Information Review. 1989. Vol. 13. Iss. 5. P. 407-424.
9. *Shultze S.A.* Collaborative Foraging Approach to Web Browsing Enrichment // Human Factors in Computing Systems: CHI Conference Proceedings. – N. Y., 2002. P. 860-861.
10. *Шокин Ю.И., Веснин А.Ю., Добрынин А.А., Клименко О.А., Рычкова Е.В., Петров И.С.* Исследование научного веб-пространства Сибирского отделения Российской академии наук // Вычислительные технологии. 2012. Т. 17. № 6. С. 85-98.
11. Ш*окин Ю.И., Веснин А.Ю., Добрынин А.А., Клименко О.А., Рычкова Е.В.* Анализ веб-пространства академических сообществ методами вебометрики и теории графов. // Информационные технологии. 2014. № 12. С. 31-40.

Problems of research of the processes of self-organization in a web-space

Olga A. Klimenko *PhD, Senior Research Associate*

Roman Yu. Fedorov *PhD, Senior Research Associate*

The article is devoted to the problems of development of interdisciplinary approaches to research of the principles of self-organization of a web space and regularities of its communicative interactions. As a methodological basis of the research there was chosen synthesis of biosocial interpretations of communicative processes and their simulation by means of a webometrics.

Keywords – Internet communities; web space; webometrics; graph theory.

УДК 004.031.42:65.011.56

АЛГОРИТМ ГЕНЕРАЦИИ ПРОГРАММНЫХ КОМПОНЕНТОВ МОДЕЛЬНО-ОРИЕНТИРОВАННОЙ СИСТЕМЫ

Алексей Александрович Коробко, *м.н.с,*
Тел.: +7 913 575 7387, e-mail: agok@icm.krasn.ru
Институт вычислительного моделирования
Сибирского отделения РАН (ИВМ СО РАН),
http://icm.krasn.ru

Описан авторский подход к построению модельно-ориентированных систем. Представлено теоретико-множественное описание мета-матамодели системы учета ре-

зультатов научной деятельности. Предложен комплекс алгоритмов генерации программных компонентов.

Ключевые слова: веб-система, оперативный сбор данных, модельно-ориентированная архитектура, динамическая генерация пользовательского интерфейса, реляционная модель данных, метаданные

Введение

Рано или поздно каждая организация сталкивается с необходимостью оперативного получения регламентированной информации от своих сотрудников, подразделений или связанных структур. Эффективность сбора данных определяется, во-первых, скоростью формирования учётных и отчётных форм, во-вторых, скоростью получения ответов, т.е. заполнения предоставленных форм и их агрегации для дальнейшей аналитической обработки. Задачи, связанные с оперативным предоставлением форм и агрегацией получаемой информации, можно решить с использованием технологий распределённого сбора данных [1]. Создание настраиваемой системы сбора данных, адаптирующейся к меняющимся потребностям пользователей, позволит решить проблемы связанные с динамическим расширением тематического наполнения. Наиболее перспективный подход, позволяющий достичь нужного уровня адаптируемости информационной системы, основан на разработке программного обеспечения с использованием модельно-ориентированной архитектуры – Model Driven Architecture (MDA) [2; 3].

А.А. Коробко

Использование MDA при создании программного обеспечения позволяет кардинально повысить эффективность разработки за счёт устранения части этапа «ручного» программирования и повышения управляемости разработки. В работе [4] предложена оригинальная реализация модельно-ориентированного подхода к разработке программного обеспечения, позволяющая строить модельно-ориентированную систему с возможностью оперативного управления метамоделью и изменения тематического наполнения системы силами пользователей. Одной из актуальных проблем в рамках предложенного подхода является решение задачи автоматизированного формирования программных компонентов, составляющих модельно-ориентированную систему.

Модельно-ориентированная система

Построение модельно-ориентированных систем сбора данных в рамках оригинальной реализации подхода к модельно-ориентированной разработке (MDD) включает в себя этапы проектирования моделей абстрактного уровня и моделей прикладного уровня [5]. Проектирование моделей абстрактного уровня включает процесс построения мета-метамодели (M3), модели языка моделирования и процесс построения метамодели (M2), метамодели различных специфичных областей приложения. Проектирование прикладных моделей состоит из процесса построения моделей прикладного уровня (M1) и этапа формирования экземпляров концептов (M0) определённых на уровне M1 [6]. В отличие от классического подхода в авторской реализации, во-первых, предлагается объединить процессы формирования управляющей и прикладной моделей (рис. 1), включение управляющей модели в состав системы позволяет оперативно реагировать на изменяющиеся требования к тематическому наполнению. Во-вторых, предлагается добавить процесс формирования модельно-ориентированной системы, который позволит получить в результате выполнения всех процессов не программный код, а рабочую систему.

Предложенная оригинальная реализация модельно-ориентированного подхода к разработке программного обеспечения позволяет получать модельно-ориентированную

систему с функцией управления метамоделью и возможностью изменения своего функционала силами обычных пользователей.

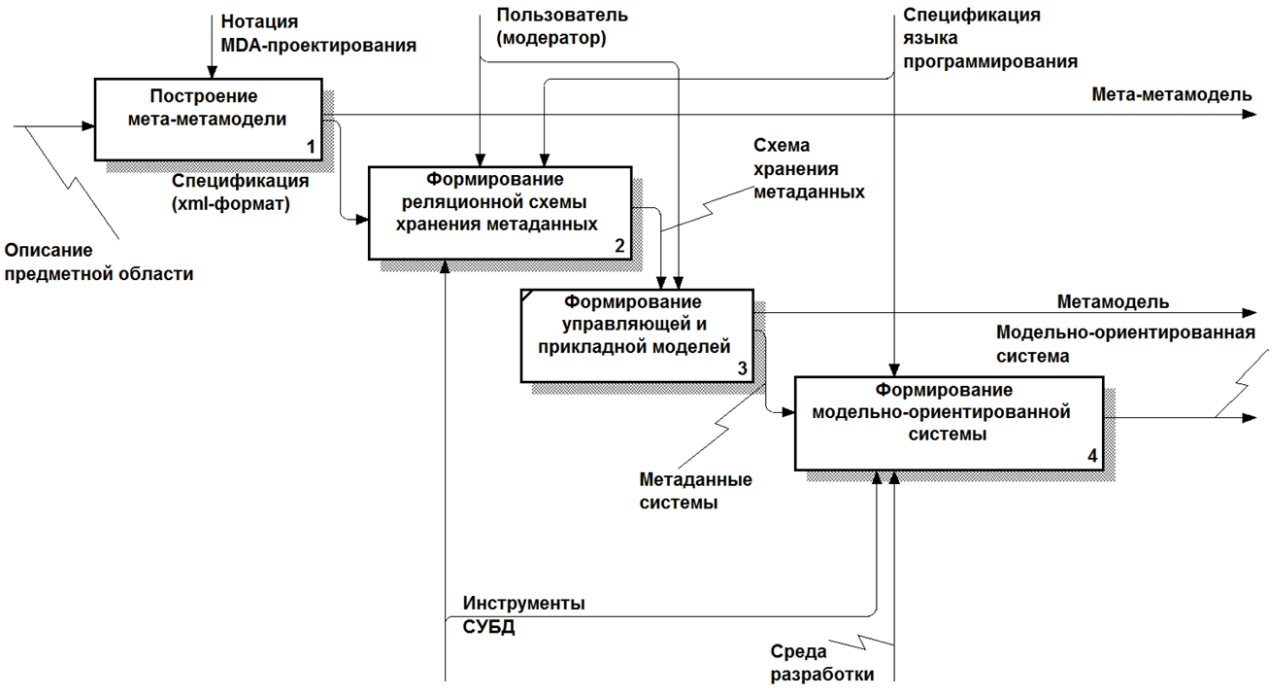

Рис. 1. Диаграмма оригинальной схемы проектирования системы

Алгоритм генерации программных компонентов модельно-ориентированной системы

Предложенная реализация модельно-ориентированного подхода использовалась при создании системы учёта результатов научной деятельности и веб-системы сбора данных мониторинга чрезвычайных ситуаций. Для обеих систем были построены мета-метамодели, создан инструментарий для формирования управляющей и прикладной модели. На основании метаданных (прикладной модели) автоматически строится модельно-ориентированная система, состоящая из программных компонент и тематического наполнения в соответствии с заданной предметной областью.

Алгоритм генерации программных компонентов основывается на анализе классов и отношений между ними, описанных в мета-метамодели. Поскольку спецификация веб-системы сбора данных является частным случаем спецификации системы учёта результатов научной деятельности, при построении алгоритма будем опираться на более полную мета-метамодель. В работе [7] представлена мета-метамодель системы учета результатов научной деятельности (рис. 2), содержащая описание трёх классов объектов: класс «Объект научной деятельности» – N, класс «Атрибут научной деятельности» – F и класс «Тип научной деятельности» – G. Объекты класса «Атрибут научной деятельности» описываются тройкой $F=(A, T, D)$, где A – наименование атрибута, T – имя специализированного типа атрибута, D – флаг темпоральности атрибута. В рамках мета-метамодели задан ряд отношений, определяющих связи между экземплярами (объектами) классов модели. Между объектами научной деятельности определены два вида отношений: «Вложенность» и «Зависимость». Одно-многозначное отношение «Вложенность» – φ задано на множестве N, $\varphi \subseteq N \times N$. Отношение используется для задания организационной иерархии объектов научной деятельности. Одно-многозначное отношение «Зависимость» – χ задано на множестве N, $\chi \subseteq N \times N$. Отношение позволяет связывать объекты друг с другом, реализуя различные функциональные взаимодействия. Много-многозначное соответствие между объектами и атрибутами научной деятельно-

сти «Обладание» обозначим как θ, θ⊆N×F. Одно-многозначное соответствие между объектами и типами научной активности «Группировка» – ψ, ψ⊆N×G.

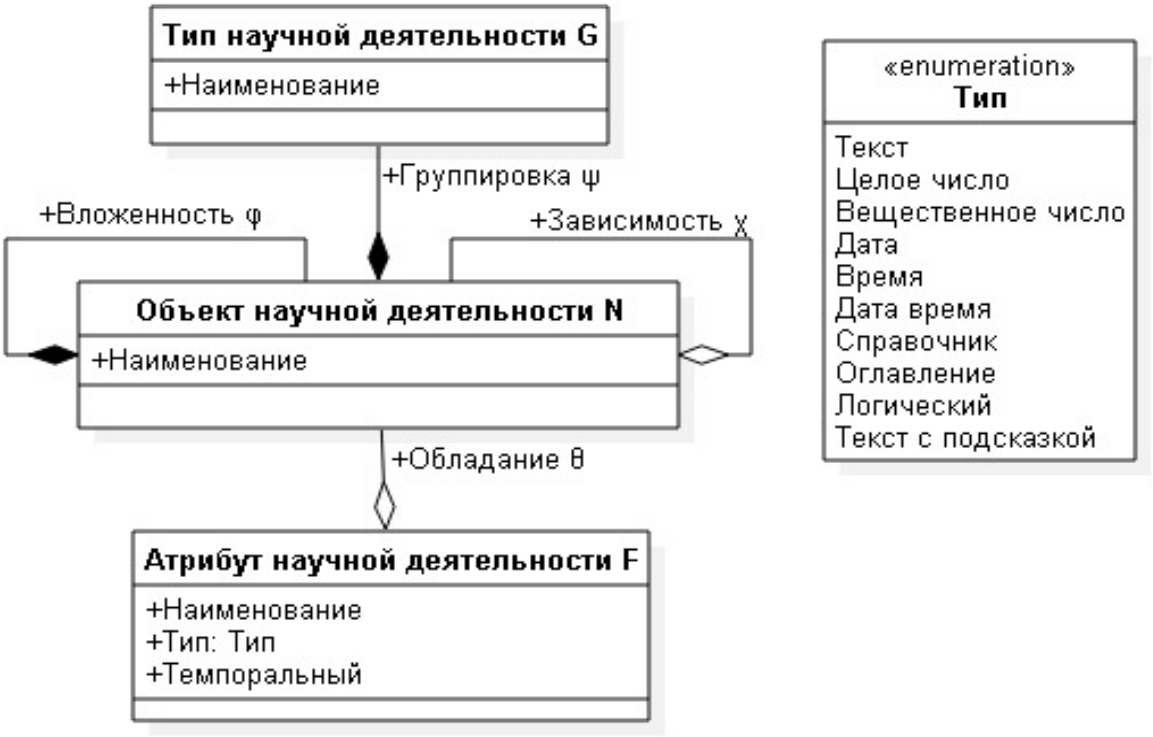

Рис. 2. Мета-метамодель системы учета результатов научной деятельности

С учётом приведённых обозначений для управляющей модели алгоритм генерации программных компонентов выглядит следующим образом (рис. 3). Согласно предложенной оригинальной реализации модельно-ориентированного подхода информация об управляющей и прикладной моделях хранится в базе данных в виде метаданных. Информация о множествах и отношениях между ними при реализации алгоритма, получается посредством sql-запросов к базе данных. На основе анализа метаданных системы в соответствии с предлагаемым алгоритмом происходит формирование описания программных компонентов в виде json-объектов. Модельно-ориентированная система состоит из серверной и клиентской частей, серверная часть обеспечивает анализ метаданных и формирование программных компонентов, клиентская часть формируется в соответствии с тематическим наполнением системы – хранимыми метаданными. Обмен информацией между сервером и клиентами происходит в формате json, поскольку этот формат оптимален для сериализации сложных структур, описывающих отношения между элементами множеств управляющей модели.

Входными параметрами алгоритма генерации программных компонентов являются множество типов научной деятельности G, множество объектов научной деятельности N и отношение группировки ψ между объектами и типами научной деятельности. На первом шаге алгоритма происходит начальная инициализация json-объекта, который позволяет сочетать в себе разнородные объекты, строить иерархию, и его заполненный вариант является результатом, возвращаемым пользователю. В первом цикле мы проходим все значения множества типов научной деятельности, и для каждого добавляем дочерний узел r к корневому элементу json-объекта R. В следующем вложенном цикле в соответствии с отношением группировки ψ перебираем связанные с типом научной деятельности g объекты научной деятельности n. Для каждого объекта n добавляем в r дочерний узел d, который используем на следующих шагах алгоритма для присоединения атрибутов и реализации иерархической зависимости между объектами научной деятельности.

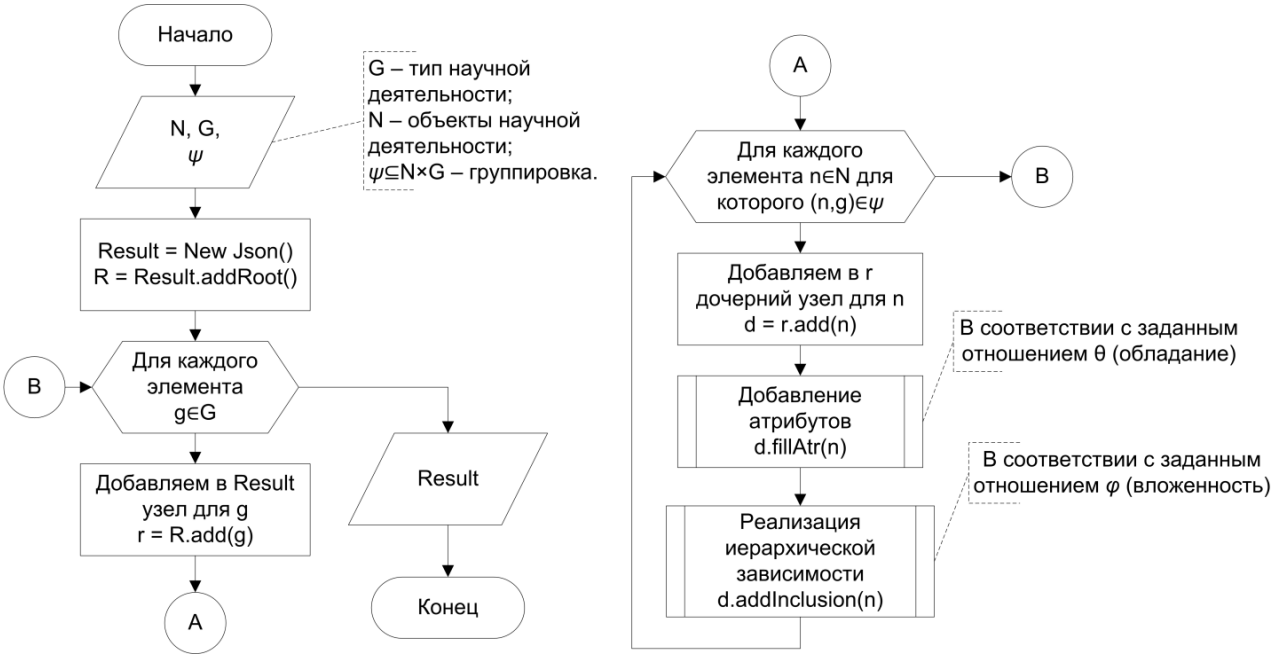

Рис. 3. Алгоритм генерации программных компонентов

Добавление атрибутов – сложный процесс, подробно расписанный на рисунке 4. Входными параметрами алгоритма, реализующего отношения обладания и зависимости, являются: обрабатываемый объект n, узел иерархии r, множество объектов научной деятельности N, множество атрибутов научной деятельности F и отношения зависимости χ между объектами научной деятельности и обладания θ между объектами научной деятельности и атрибутами. Работа алгоритма начинается с обработки в цикле атрибутов F, связанных отношением обладания с обрабатываемым объектом n. Информацию о каждом атрибуте добавляем в узел r, затем заполняем наименование атрибута, тип и флаг темпоральности. После обработки всех связанных атрибутов переходим к отношению зависимости χ. В цикле обрабатываются объекты научной деятельности из N, связанные отношением зависимости с обрабатываемым объектом n, и по каждому связанному объекту в узел r добавляется информация. Результатом работы процесса является изменённый узел иерархии r, для объекта «Статья в журнале» системы учёта результатов научной деятельности фрагмент json-массива атрибутов выглядит следующим образом:

```
items: [{ fieldLabel: 'Идентификатор', xtype: 'hidden', name: 'idpub',
    allowBlank: false, anchor:'100%'
  },{ fieldLabel: 'Тип публикации', xtype: 'hidden', name: 'idptype',
    allowBlank: false, anchor:'100%'
  },{ fieldLabel: 'Название', xtype: 'textarea', name: 'pub',
    allowBlank: false, anchor:'100%'
  },{ xtype: 'agcombo', fieldLabel: 'Авторы', name: 'idauth[]', valueField: 'idauth',
    displayField: 'auth', store: 'auth', anchor: '100%'
  },{ xtype: 'agcombo', fieldLabel: 'Журнал', name: 'idpar', valueField: 'idpub',
    displayField: 'pub', store: 'pub', multiSelect: false,
    filters: [{ property: 'idptype', value: '2', anyMatch: false, caseSensitive: false }],
    allowBlank: false, anchor: '100%'
}, ...]
```

Массив атрибутов содержит в себе информацию, полученную как из управляющей, так и прикладной моделей, т.е. полную информацию необходимую для построения программных компонентов системы, связанных с атрибутами.

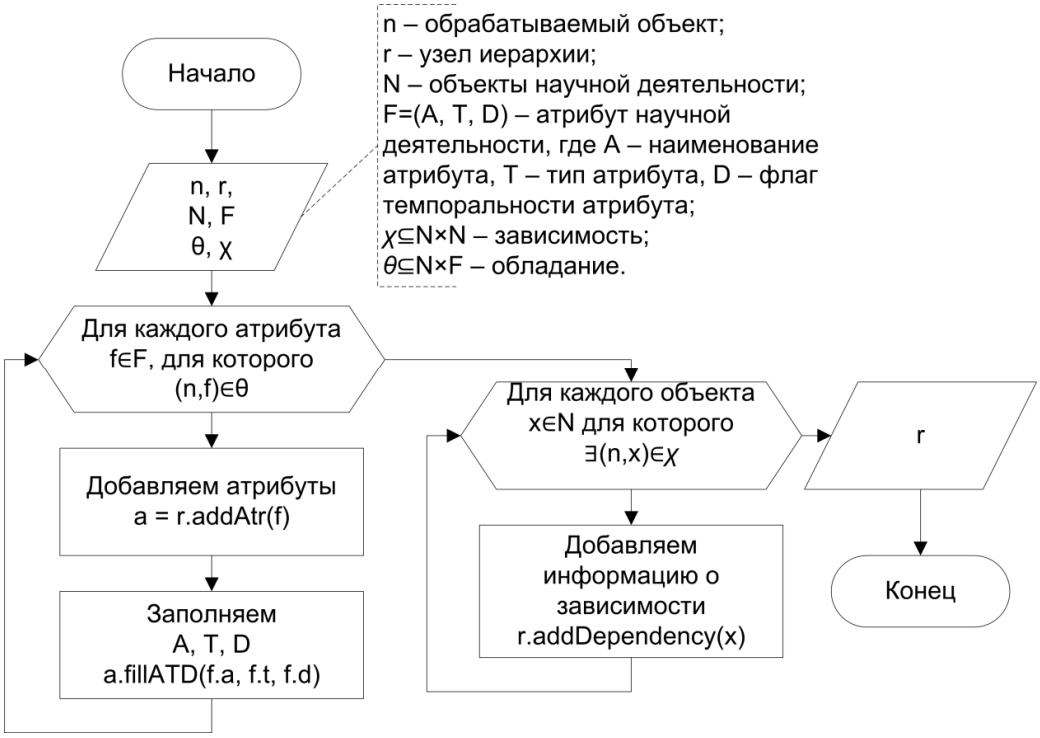

Рис. 4. Алгоритм процесса реализации отношений обладания и зависимости

Процесс реализации иерархической зависимости вызывается из основного алгоритма генерации программных компонентов и является инициатором процесса реализации отношений обладания и зависимости (рис.5). Входными параметрами алгоритма являются: обрабатываемый объект n, родительский узел r, объекты научной деятельности N и отношение вложенности φ. Работа алгоритма проходит в цикле, обрабатывающем объекты научной деятельности v, связанные отношением вложенности φ с объектом n. Для каждого объекта v в родительский узел r добавляется дочерний узел c. Затем для вновь созданного узла заполняется информация об атрибутах с помощью алгоритма реализации отношений обладания и зависимости. Поскольку отношение вложенности не ограничивается одним уровнем, происходит проверка у текущего объекта отношения вложенности с другими объектами, и в случае подтверждения наличия отношения происходит рекурсия, т.е. инициация процесса реализации иерархической зависимости с новыми параметрами. В случае отсутствия вложенности процесс прекращает работу и возвращает изменённый узел иерархии r, для системы учёта результатов научной деятельности узел иерархии выглядит следующим образом:

```
items: [{ text:  'Издание',
          xtype: 'ptype_button', idptype: "21", idparpub: "", istype: "0", childcnt: "3",
          menu: [{ xtype: 'ptype_menu',idptype: '23', idparpub: '',
                   istype: '1', text: 'Материалы'
          },{ xtype: 'ptype_menu',idptype: '2', idparpub: '', istype: '1',  text: 'Журнал'
          },{ xtype: 'ptype_menu', idptype: '3', idparpub: '',  istype: '1',
                   text: 'Сборник статей'
          }]
        },{ text: 'Статья',
          xtype: 'ptype_button', idptype: "19", idparpub: "", istype: "0", childcnt: "2",
          menu: [{text: 'Статья в журнале',
                   xtype: 'ptype_menu',idptype: '14', idparpub: '2', istype: '1'
          },{text: 'Статья в сборнике',
                   xtype: 'ptype_menu', idptype: '20', idparpub: '3', istype: '1'
          }],
        }, ...]
```

Поскольку для отображения иерархии в системе учёта результатов научной деятельности используется меню, при построении мы используем соответствующие типы узлов, а вложенность реализуем с помощью подменю. В представленном примере мы не видим информацию об атрибутах, поскольку алгоритм для их формирования вызывается после вызова соответствующего элемента меню, что позволяет не производить лишние вычисления и уменьшает время формирования иерархии компонентов.

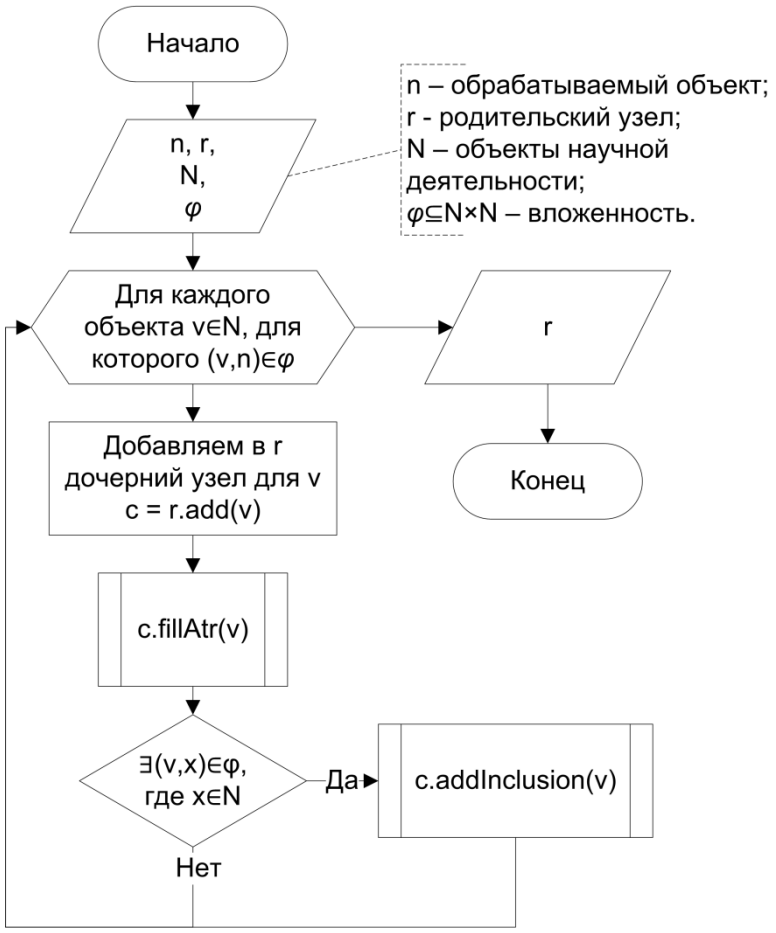

Рис. 5. Алгоритм процесса реализации отношения вложенности

После завершения процессов добавления атрибутов и реализации иерархических зависимостей основной процесс алгоритма генерации программных компонентов переходит в цикле к следующему объекту научной деятельности n, а в случае перебора всех элементов отношения ψ к следующему типу научной деятельности g. Результатом работы алгоритма является полностью заполненный json-объект, который используется для формирования клиентского приложения.

Предложенный алгоритм генерации программных компонентов позволяет оперативно реагировать на изменения управляющей и прикладной моделей. Оперативность достигается за счёт исключения этапа перепрограммирования из процесса реагирования системы на изменение тематического наполнения. Изменением моделей занимается модератор, затем с использованием предложенного алгоритма автоматически формируются программные компоненты, тут же используемые для построения системы.

Заключение

Предложенный алгоритм генерации программных компонентов использован в реализации модельно-ориентированной системы с функцией управления метамоделью и возможностью изменения своего функционала силами обычных пользователей. Использование предложенного алгоритма позволяет достичь высокого уровня адаптируемости информационной системы в соответствии с меняющимися потребностями пользователей.

Предложенный алгоритм был апробирован в Институте вычислительного моделирования при создании системы учёта результатов научной деятельности и веб-системы сбора данных, являющейся частью системы консолидации и анализа данных центров мониторинга и прогнозирования чрезвычайных ситуаций.

Литература

1. *Коробко А.А., Ничепорчук В.В., Ноженков А.И.* Динамическое формирование интерфейса ВЕБ-системы сбора данных мониторинга чрезвычайных ситуаций //Информатизация и связь. 2014. № 3. С. 59-64.

2. Object Management Group Model Driven Architecture (MDA), MDA Guide rev. 2.0, OMG Document ormsc/2014-06-01, http://www.omg.org/cgi-bin/doc?ormsc/14-06-01.

3. OMG Meta Object Facility (MOF) Core Specification, ver. 2.5, OMG Document formal/2015-06-05, http://www.omg.org/spec/MOF/2.5/.

4. *Коробко А.А.* Модельно-ориентированная система сбора данных // Проблемы информатизации региона. ПИР 2015: Материалы XIV Всероссийской научно-практической конференции – Красноярск. 2015. С. 116-123.

5. *Seidewitz E.* What Models Mean, IEEE Software, 2003, vol. 20, no. 5, pp.26-32.

6. *Atkinson C., Kühne T.* Model-Driven Development: A Metamodeling Foundation (Spec. issue on Model-Driven Development) // IEEE Software. 2003. Vol. 20. No. 5. P.36-41.

7. *Korobko Aleksei A., Korobko Anna V.* Constructing Domain model for Scientific Activity Management System //Modeling of Artificial Intelligence. 2015. Vol. (6). Is. 2. pp. 82-89.

Algorithm of program components generation for the model-driven system

Aleksei Aleksandrovich Korobko, *scientific researcher,*

Institute of Computational Modelling of the Siberian Branch of the RAS (ICM SB RAS),

Author's approach to the construction of model-driven system is described. Set-theoretic description of meta-metamodel for Scientific Activity Management System is presented. A set of algorithms for program components generation is suggested.

Keywords: scientific activity results, web-system, ad-hoc data consolidation, model driven architecture, dynamic generating user interface, relational data model.

УДК 004.891.3

РАЗРАБОТКА ЭКСПЕРТНОЙ СИСТЕМЫ АНАЛИЗА СТРУКТУРЫ ЭЛЕКТРОКАРДИОСИГНАЛА

Кристина Андреевна Красовицкая*, магистрант*
Тел.:+7 983 463 1106, e-mail: kristina.kras1993@gmail.com,
Иркутский национальный исследовательский технический университет
www.istu.edu/
Евгений Александрович Черкашин*, старший научный сотрудник, к. т.н., доцент*
Тел.:+79148706754, e-mail: eugeneai@irnok.net
Иркутский национальный исследовательский технический университет
Институт динамики систем и теории управления имени В.М. Матросова
Сибирского отделения Российской академии наук
www.istu.edu/, http://www.icc.irk.ru/

В данной работе представлен этап разработки кардиологической экспертной системы. Рассмотрены задачи выделения вектора данных из линии напряжения на ЭКГ, нахождения пиков PQRST, анализа ЭКГ посредством вейвлет-преобразования.

Ключевые слова: ЭКГ, анализ, биомедицинский сигнал, машинное обучение