

Мицель Артур Александрович

Д-р техн. наук, профессор кафедры автоматизированных систем управления ТУСУРа

Телефон: (3822) 41 31 57

Эл. почта: maa@asu.tusur.ru

Каштанова Ольга Валерьевна

Ассистент кафедры автоматизированных систем управления ТУСУРа

Телефон: (3822) 41 31 57

Дворянидова Ирина Игоревна

Студентка кафедры ВММФ Томского политехнического университета

A.A. Mitsel, O.V. Kashtanova, I.I. Dvoryanidova

Optimization's models of goods and services order's performer choice taking into consideration customer's and supplier's interests

In this article the problem of products and services suppliers choice during the purchase competition for State institutions was considered. Two optimization's models which take into account customers' and suppliers' interests were developed. Customer's interests were taken into account with the help of rating matrix, which based on point methodology. Model's efficiency is demonstrated by the example.

УДК 658.012.011.56

С.А. Мытник

Представление метаданных в адаптируемых информационных системах

В работе рассмотрены различные модели представления данных и их использование для создания адаптируемых информационных систем учетного профиля. Предложено комбинированное решение, позволяющее выбрать положительные аспекты использования реляционной и иерархической моделей представления данных в задачах создания адаптируемых систем учетного профиля. Рассмотрены возможности применения объектно-ориентированных методов анализа и моделирования в задачах создания адаптируемых моделей. Предложен метод формализации результатов объектно-ориентированного анализа с целью их использования вычислительной техникой для автоматического воссоздания структур объектов на основе метаданных.

Структуры данных в адаптируемых информационных системах

Для многих систем учетного профиля характерно частое изменение структур хранимых данных. При этом интенсивность изменения структур данных значительно превышает интенсивность изменения протекающих в объекте автоматизации бизнес-процессов. В настоящее время любое изменение автоматизированной информационной системы учетного профиля (АИС УП) сопровождается привлечением разработчиков системы, что вызывает комплекс негативных последствий, в частности:

- дополнительные затраты, повышающие совокупную стоимость владения АИС УП;
- торможение протекающих на объекте автоматизации технологических процессов в связи с проведением работ по настройке и запуску каждой новой версии АИС УП;
- существенное снижение эффективности от внедрения АИС УП в случаях, когда совокупные затраты на выпуск версии продукта, адаптированного к изменившимся условиям, превышают экономический эффект от его внедрения;
- растяжение сроков внесения изменений в АИС УП в целях ее адаптации к изменившимся условиям в случаях, когда для этого привлекается внешний по отношению к объекту автоматизации подрядчик;
- повышение сложности и неупорядоченности АИС УП при каждой ее адаптации, что ведет, в конечном итоге, к разрушению системы.

Вышеперечисленные негативные последствия изменения АИС УП в большинстве случаев составляют серьезную проблему для ее владельцев (пользователей). Решением данной проблемы будет являться создание технологии разработки программного обеспечения, способного изменять структуру хранимых данных под воздействием пользователя прозрачным для него образом.

Концептуальной основой для механизмов хранения информации об объектах предметной области с произвольной структурой послужила модель представления данных языка XML (см. www.xml.com). Способность этой модели представлять в иерархическом виде любые структурированные данные неоднократно подтверждалась на практике, и для языка XML нашлось применение в самых разных областях сферы информационных технологий. К основным преимуществам иерархической модели хранения данных можно отнести простоту изменения структуры хранимых данных и простоту отображения в ней информации об объектах предметной области, которая в большинстве случаев имеет иерархическую природу.

Однако для хранения и обработки данных более предпочтительной была и остается в настоящее время реляционная модель, поддерживаемая многочисленными системами управления базами данных. Такое положение дел обусловлено следующими причинами:

- реляционная модель имеет глубокую теоретическую базу, вследствие чего технологии хранения данных, основанные на реляционной модели, более устойчивы и скорее ведут разработчиков к положительным результатам;
- скорость обработки данных, представленных в реляционной модели, существенно выше, чем скорость обработки иерархических данных;
- СУБД, построенные на основе реляционной модели, обеспечивают концептуальную целостность хранимых данных на протяжении всего времени существования базы данных;
- структурированный язык запросов (SQL), применяемый для обработки данных, хранящихся по реляционной модели, имеет многолетнюю историю, и эффективность его применения в решении задач анализа не оставляет сомнений.

Ко всему прочему следует заметить, что АИС УП редко используется в чистом виде и обычно входит в состав комплексной автоматизированной системы управления объектом автоматизации (АСУ ОА) в качестве средства управления содержанием базы данных. Соответственно АИС УП должна взаимодействовать с прочими компонентами АСУ ОА, при этом взаимодействие обычно обеспечивается на уровне данных. Принимая во внимание факт предпочтительности реляционной модели для большинства компонентов АСУ ОА (кроме того, такие компоненты могут уже существовать и их переработка может быть нецелесообразной), использование иерархической модели хранения данных в АИС УП невозможно (либо ведет к серьезным проблемам в будущем).

Решением дилеммы выбора между иерархической и реляционной моделью является эмуляция работы иерархической модели внутри АИС УП, использующей реляционную модель. Такая эмуляция может быть организована следующим образом.

1. Описание структуры хранимых данных (метаданные) создается в предположении о том, что для хранения данных об объектах предметной области используется иерархическая модель. При этом описание метаданных производится в терминах предметной области.

2. На основе метаданных программное обеспечение адаптируемой АИС УП генерирует набор реляционных таблиц СУБД, в которых данные хранятся физически. Генерация таблиц происходит автоматически, прозрачным для пользователя образом.

3. Обращение прикладного программного обеспечения (ППО) к базе данных за информацией об объекте предметной области происходит через специального программного агента, который, с одной стороны, преобразует данные из реляционной модели в иерархическую (интерфейс с СУБД), а с другой — из иерархической модели в реляционную (интерфейс с ППО). Преобразование данных из одной модели в другую программный агент производит на основе метаданных об объекте предметной области.

Сопоставление метаданных с конкретным объектом предметной области происходит посредством классификации объектов, хранимых в АИС УП. При этом классификационный признак, по которому разделяются объекты предметной области для сопоставления с метаданными, выделяется в виде элемента системы словарей, справочников, классификаторов (система ССК) и хранится в БД наряду с общей информацией (рис. 1).

В предлагаемом подходе к хранению информации произвольной структуры особое место занимает взаимодействие ППО с базой данных АИС УП через специального программно-

го агента. Такой способ организации работы необходим для большей гибкости ППО и удобства работы программистов (что в конечном итоге скажется на сроках разработки любой адаптируемой АИС УП, основанной на предлагаемой технологии). Преобразование данных из реляционной структуры в иерархическую и обратно на основе метаданных является сложной операцией, требующей привлечения высококвалифицированных специалистов. Программный агент — посредник между базой данных и ППО — инкапсулирует выполнение функций по генерации SQL-запросов к реляционной базе данных на основе метаданных объекта предметной области и преобразования результата их выполнения в иерархическую структуру. Таким образом, программист, разрабатывающий ППО, освобождается от обязанности разбираться в механизмах хранения данных и работает с объектами с изменяемой структурой так же, как с классическими объектами с постоянной структурой. Программный агент-посредник автоматически воссоздает структуру объекта предметной области на основе метаданных при обращении к экземпляру соответствующего класса.

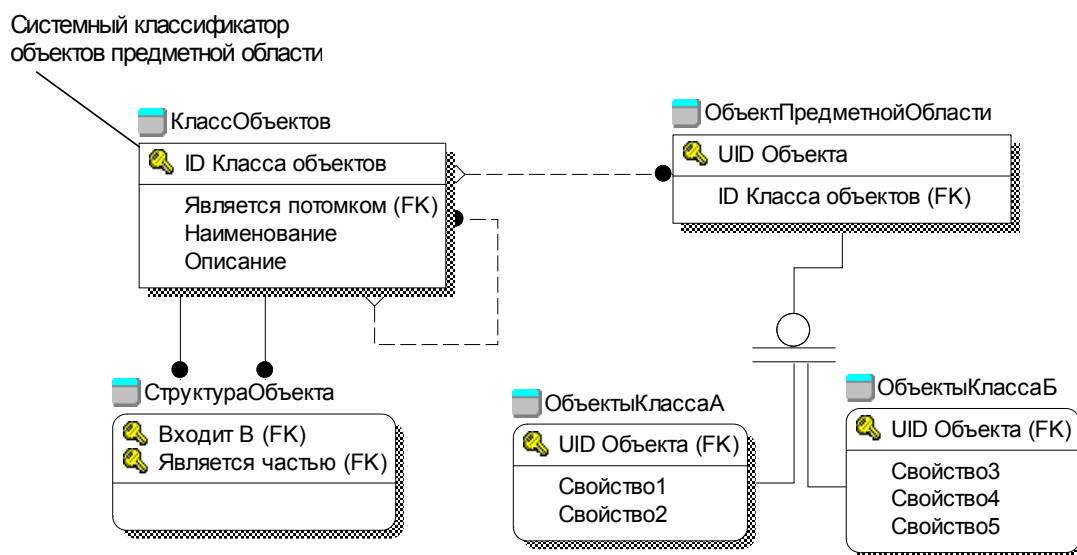


Рис. 1 — Сопоставление метаданных с конкретным объектом предметной области

Применение объектно-ориентированных методов для определения структуры данных адаптируемых информационных систем

Исходя из вышесказанного, можно сделать вывод о том, что метаданные играют ключевую роль в технологии создания адаптируемого программного обеспечения. Метаданные формируются посредством анализа предметной области как системы и обычно выражаются в виде концептуальной модели. Широко распространенный в настоящее время метод объектно-ориентированного анализа и проектирования позволяет получить модель системы в виде иерархии отдельных понятий предметной области. Такой результат моделирования является адекватным способом представления информации о данных и их структуре в случаях работы с иерархической моделью.

В основе объектно-ориентированного анализа и моделирования, как и большинства любых других методов исследования сложных систем, лежит абстрагирование. Смысл абстрагирования заключается в выделении из всего множества информации об исследуемом объекте важной, необходимой для решения проблемы. Иначе говоря, «абстракция выделяет существенные характеристики некоторого объекта, отличающие его от всех других видов объектов, и, таким образом, четко определяет его концептуальные границы с точки зрения наблюдателя» [1]. Фактически, абстракция позволяет сформировать в сознании исследователя (интеллектуальной среде) мыслеобраз, в котором заключена вся существенная в рамках разрабатываемой системы информация. Эта информация в дальнейшем преобразуется в знания, на основе которых строится модель предметной области.

В процессе анализа базовые абстракции разбиваются на более мелкие понятия, которые затем представляются в виде иерархии понятий предметной области. Классики объектно-ориентированного анализа и проектирования выделяют два основных вида иерархий —

структура классов (*is a*) и структура объектов «целое-часть» (*has a*) [1, 2]. По сути, построение иерархии «*is a*» является результатом классификации различных объектов предметной области. Построение структуры объектов (иерархия «*has a*»), в свою очередь, является результатом объектной декомпозиции анализируемой системы, позволяющей разбивать выделенные крупные объекты со сложной внутренней структурой и поведением на ряд более мелких взаимосвязанных частей.

В основе построения иерархии «*is a*» лежит выделение общих признаков объектов или классов и формирование на их основе общего класса-предка, обладающего всеми выделенными признаками. Полученный класс-предок будет связан с исходными классами-потомками отношением обобщения (генерализации). Назовем такую процедуру структурно-обобщающей классификацией. Пусть в результате абстрагирования мы выделили из предметной области два объекта: объект O_1 , обладающий множеством атрибутов A_1 , и объект O_2 , обладающий множеством атрибутов A_2 . Допустим, семантика описания этих объектов разная, то есть $A_1 \neq A_2$. Следовательно, каждый из этих объектов будет принадлежать своему классу — C_1 и C_2 соответственно. В результате анализа структуры и семантики описания каждого из полученных классов были выяснены общие закономерности и выделен некоторый суперкласс C с набором атрибутов A . При этом $A \subset A_1$, $A \subset A_2$.

В этом случае можно сказать, что C_1 *is a* C , C_2 *is a* C , то есть классы C_1 и C_2 связаны с классом C отношением обобщения. Говоря в терминах объектно-ориентированного программирования, класс C является родительским классом, или суперклассом, для классов C_1 и C_2 .

Таким образом, процесс построения иерархии классов «*is a*» заключается в поиске соответствий между выявленными в процессе анализа классами, объектами и выявлении обобщающих классов путем построения пересечений множеств атрибутов обобщаемых классов. Этот процесс является итеративным, так как при построении иерархии классов «*is a*» на каждом шаге возможно появление новых классов, которые можно обобщить. Обычно процесс заканчивается при появлении в вершинах построенного дерева классов, соответствующих базовым понятиям предметной области, для которых характерна высокая степень абстракции.

Построение иерархии «*has a*» является основой объектной декомпозиции анализируемой системы. При выделении объекта предметной области исследователь начинает разбивать его структуру. При этом часто он понимает, что объект не является атомарным, а состоит из множества других объектов. Допустим, имеется некоторый объект O с множеством атрибутов $\{A_1, A_2, A_3\}$. При детальном рассмотрении атрибута A_2 исследователь выясняет, что он имеет собственную сложную структуру и поведение, т.е. сам, по сути, является объектом. В этом случае атрибут A_2 выделяется как отдельный объект O_1 , обладающий множеством атрибутов A_2 . Таким образом, объект O разбит на два объекта O и O_1 , связанных отношением агрегации, т.е. O *has a* O_1 .

Построение обеих иерархий («*is a*» и «*part of*») является двумя взаимодополняющими процессами. Выделение обобщающих понятий позволяет на основе полученных закономерностей более подробно представить структуру конкретных объектов. Декомпозиция объектов приводит к появлению новых схожих понятий, требующих обобщения. Оба этих взаимодействующих процесса являются единым комплексным процессом анализа структуры системы. Процесс анализа структуры — итеративный, каждая из итераций состоит в общем случае из двух частей:

- восхождение от конкретного к абстрактному с целью классификации и описания объектов предметной области, а также выяснения закономерностей и получения обобщений;
- спуск от абстрактного к конкретному с целью проверки адекватности полученной модели и закономерностей, а также поиска претендентов на дальнейшее обобщение и декомпозицию.

Представление метаданных в адаптируемых информационных системах

Широкое распространение объектно-ориентированных методов привело к появлению унифицированного языка моделирования (стандарт Unified Modeling Language — UML [3]) как формы представления моделей и средств описания метаданных (стандарт Meta Object Facility — MOF [4]) как формы описания семантики моделей. UML получил широчайшее распространение благодаря его использованию в задачах проектирования информационных систем различного профиля. MOF, являясь основой UML, известен меньше и применяется в

основном в задачах интеграции баз данных и создания хранилищ данных. Совокупность стандартов UML и MOF является адекватным средством для работы с иерархическими моделями предметной области, полученными с помощью объектно-ориентированных методов, однако их применение для решения задачи представления метаданных в адаптируемой информационной системе весьма затруднительно по следующим причинам:

- отсутствуют формальные методы обработки информации, представленной посредством UML и MOF, с помощью вычислительной техники (либо нам о них неизвестно), что не позволяет адаптируемой информационной системе однозначно воссоздать структуру объекта по ее описанию;
- назначение UML и MOF выходит далеко за рамки описания структур данных, а вытекающая отсюда сложность этих средств является излишней в задаче создания систем с изменяемой структурой данных.

Тем не менее опыт создания универсальных языков описания данных и метаданных можно положить в основу средства описания структур хранимых данных для адаптируемых информационных систем. При этом такое средство должно удовлетворять следующим условиям.

1. Логически данные адаптируемой информационной системы должны храниться в иерархической модели для обеспечения гибкости их структуры (физически данные могут храниться в любой модели, включая реляционную).
2. Метаданные должны храниться в иерархической модели для обеспечения единства подходов в работе с данными и метаданными.
3. Описание структур хранимых данных для адаптируемых информационных систем должно позволять обработку информации с помощью вычислительной техники, для того чтобы программное обеспечение могло автоматически воссоздавать структуру хранимых объектов.
4. Описание структур хранимых данных должно строиться на основе результатов объектно-ориентированного анализа предметной области и отражать иерархии «*is a*» и «*has a*».
5. Описание структур хранимых данных должно позволять отображение иерархической модели представления данных в реляционную.

Представим иерархию «*is a*» в виде ориентированного графа типа «дерево»

$$G_{isa} = (O, R_{isa}),$$

где O — множество узлов, соответствующих понятиям модели, метамодели и т.д.; R_{isa} — множество связей, соответствующих отношениям «*is a*» между элементами модели, метамодели и т.д., $R_{isa} \subset O \times O$.

В вершине дерева находится элемент, соответствующий абсолютно абстрактному понятию «элемент модели». Этот элемент, по аналогии с MOF, порождает базовые понятия «Класс», «Атрибут», «Тип атрибута». Далее строится модель предметной области, при этом «Класс» порождает классы модели предметной области в виде иерархии, «Атрибут» порождает атрибуты, которыми обладают классы модели предметной области, «Тип атрибута» соответственно порождает типы атрибутов.

Определение сути любого элемента модели при этом производится путем построения пути от узла, соответствующего рассматриваемому элементу, к корню дерева.

На рис. 2 приведен пример такого графа для абстрактной предметной области.

Иерархия «*has a*» может быть представлена в виде ориентированного графа

$$G_{hasa} = (O, R_{hasa}),$$

где O — множество узлов, соответствующих понятиям модели, метамодели и т.д.; R_{hasa} — множество связей, соответствующих отношениям «*has a*» между элементами модели, метамодели и т.д., $R_{hasa} \subset O \times O$.

На таком графе отображаются понятия типа «объект O_1 включает в свой состав объект O_2 », «объект O_1 имеет атрибут A_1 », «атрибут A_1 имеет тип T_1 ».

Полученные графы не имеют смысла каждый сам по себе. Ни граф G_{isa} , ни граф G_{hasa} не позволяют получить полную информацию о структуре интересующего объекта. Следовательно, для решения задачи воссоздания структуры объекта по описанию необходимо совместить графы G_{isa} и G_{hasa} на одном графе с двумя типами связей:

$$G = (O, R_{isa}, R_{hasa}).$$

На основе полученного описания задача воссоздания структуры объекта решается следующим образом (рис. 3).

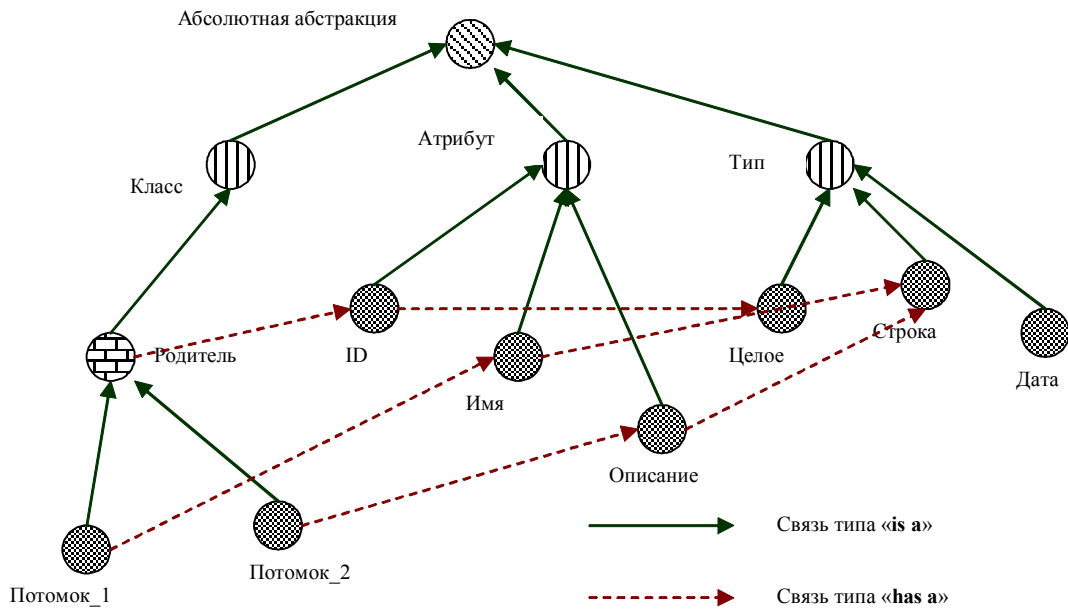


Рис. 2 — Представление метаданных с помощью графа

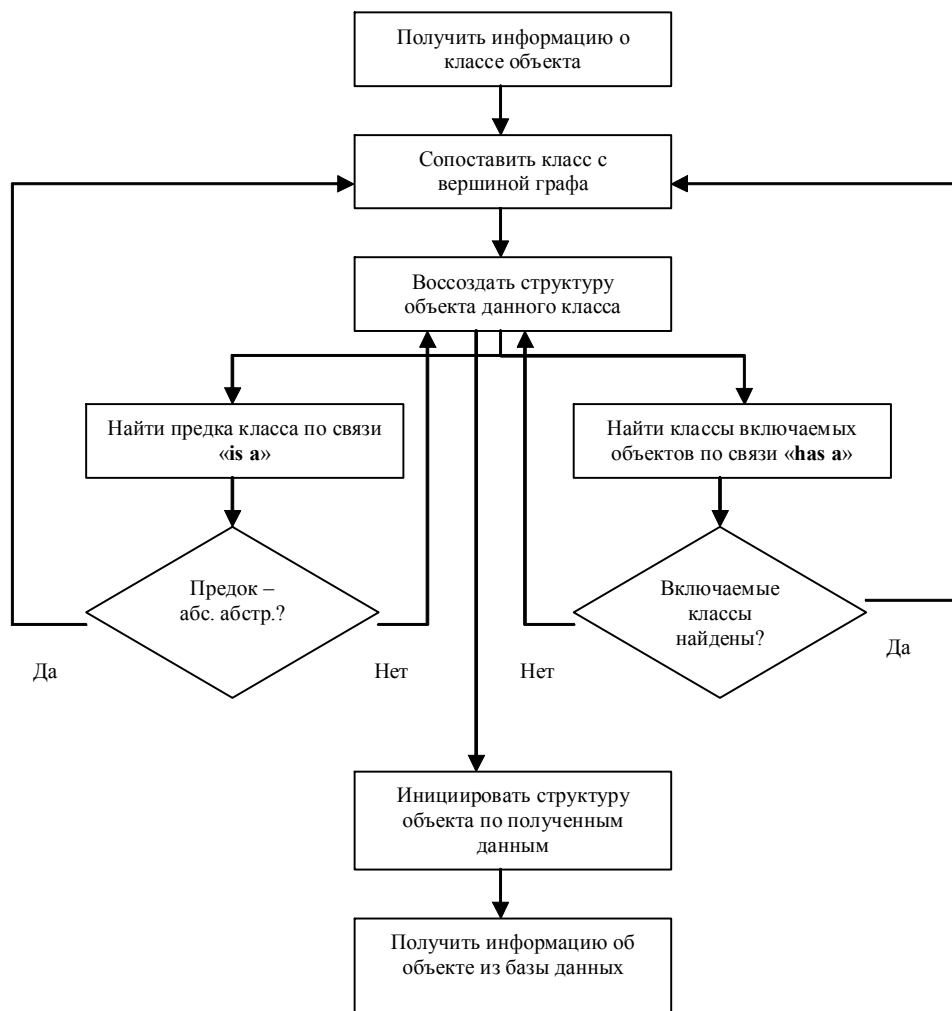


Рис. 3 — Алгоритм воссоздания структуры объекта по графу метаданных

1. Для объекта определяется его класс. Сведения о классификации (типах) объектов хранятся в системе ССК АИС УП, связанной с БД АИС УП, хранящей информацию об объектах предметной области.

2. Для найденного класса на графе находится соответствующая классу вершина. Сопоставление может производиться по имени, идентификатору класса или любым другим способом в зависимости от принятой системы именования классов на графе.

3. Воссоздание структуры объекта производится с помощью рекурсивной функции, выполняющей поиск вышестоящих классов-предков по связям «*is a*», а также классов включаемых объектов по связям «*has a*» (восхождение от конкретного к абстрактному). Для каждой найденной вершины функция запускает себя с целью продолжения воссоздания структуры объекта. Поиск предков по связям «*is a*» заканчивается при возникновении в качестве вышестоящего предка вершины, соответствующей абсолютной абстракции. Поиск классов включаемых объектов заканчивается при отсутствии у рассматриваемой вершины связей типа «*has a*».

4. Результаты выполнения рекурсивной функции собираются в обратном порядке (спуск от абстрактного к конкретному). Структуру объекта можно считать воссозданной.

5. На основе воссозданной структуры объекта загрузку информации о нем из БД можно считать тривиальной задачей. Способ ее решения зависит от выбранного способа хранения данных. К примеру, при хранении данных в реляционной БД метаданные, представленные в виде графа, могут хранить информацию об именах таблиц и столбцов, на основе которых программное обеспечение может легко сгенерировать запрос на SQL для выборки данных об объекте.

Литература

1. Буч Г. Объектный анализ и программирование с примерами приложений на C++. 2-е изд. / Пер. с англ. под ред. И. Романовского и Ф. Андреева. – М.: Бином, СПб.: Невский диалект, 1998. – 560 с.

2. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения / Пер. с англ. под ред. В. Горбункова. – СПб.: Питер, 2002. – 496 с.

3. Object Management Group. Unified Modeling Language Specification. March 2003. Version 1.5.

4. Object Management Group. Meta Object Facility Specification. April 2002. Version 1.4.

Мытник Сергей Александрович

Инженер кафедры автоматизации обработки информации ТУСУРа

Телефон: (3822) 41 43 11

Эл. почта: msa@muma.tusur.ru

S.A. Mytnik

Metadata representation in adaptable information systems

In work various models of data presentation and their use for creation of adaptable information systems are considered. Is offered the combined decision, allowing to choose positive aspects of use of relational and hierarchical model of data presentation in tasks of creation of adaptable information systems. Considered an opportunity of application of object-oriented methods of the analysis and modelling in tasks of creation of adaptable models. The method of formalization of results of the object-oriented analysis is offered with the purpose of their use by computer facilities for an automatic reconstruction of structures of objects on a basis metadata.
