

## ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ НА ОСНОВЕ СЕМАНТИЧЕСКИХ МОДЕЛЕЙ

*Бикмуллина И.И.*

*ФГБОУ ВПО Казанский национальный исследовательский технический университет*

*им. А.Н. Туполева-КАИ,*

*420111, г. Казань, ул. К. Маркса, д.10.*

*e-mail: elsiyar-b@yandex.ru*

*поступила в редакцию 27 августа 2013 года*

### Аннотация

Интеллектуальная среда рассматривается в контексте инвариантных вычислительных средств для моделирования предметной области различной профессиональной тематики. Показано, что для придания этим средам свойства интеллектуальности целесообразно использовать семантические (смысловые) модели. Семантические модели включают базовый семантический язык создания профессиональных языков описания предметной области (декларативная составляющая), инструментальные средства интерпретации семантических описаний какой-либо предметной области (операционная составляющая). Рассматриваются декларативные средства описания структурных свойств выбранной предметной области.

**Ключевые слова:** *диаграмма классов; семантика; предметная область; структурных свойства.*

**Введение.** В настоящей работе рассматривается задача синтеза структурной модели информационной технологии: синтез диаграммы классов. Цель данной работы облегчить и улучшить работу программистов. Современным разработчикам объектно-ориентированных приложений приходится создавать эффективные программные продукты в достаточно жёсткие промежутки времени. При этом в процессе разработки программного обеспечения хватает всевозможных трудностей.

Главное – это естественная сложность предметной области, к которой относится решаемая задача. Всякий раз, когда при разработке программного обеспечения возникает необходимость автоматизировать созданные человеком сложные системы, избежать этой сложности нельзя – ею можно только «овладеть». Для этого необходима хорошая предметно-ориентированная модель, проникающая значительно дальше поверхностного взгляда на проблему. Если в такой модели удастся правильно отразить внутреннюю структуру предметной области, то разработчики программного обеспечения получают именно тот инструмент, в котором они нуждаются. Хорошая модель предметной области представляет огромную ценность, но построить ее нелегко. Умеют это делать немногие, а научить других этому искусству очень трудно.

**Основная часть.** Созданные CASE-средства с использованием интеллектуальных методов позволяют преодолеть трудности, возникающие у разработчиков программных систем, более быстрыми темпами и с меньшими трудозатратами. В России широко используемыми CASE-средствами, поддерживающими объектно-ориентированные методы, являются Rational Rose Enterprise Edition 2000/2002 (фирмы Rational Software Corporation), Oracle Developer Suite 2000 (фирмы Oracle) и др. Следует заметить, что Г. Буч и Дж. Рамбоух являются создателями языка UML (Unified Modeling Language), появившегося в свет в 1994 г., который положен в основу CASE-средства Rational Rose Enterprise Edition, сотрудником фирмы, его разработавшей, в настоящее время является Г. Буч.

В последние годы наиболее перспективной методологией для создания программных систем признана объектно-ориентированная методология, одним из основоположников которой является Г. Буч. Центральное место в объектно-ориентированном анализе и

проектировании программ занимает разработка логической модели системы в виде диаграммы классов (class diagram).

Составными частями процесса разработки информационной технологии с использованием CASE – средств являются:

- разработка словаря предметной области;
- разработка диаграмм с помощью языка UML (унифицированный язык моделирования);
- синтез программ.

Унифицированный язык моделирования UML (Unified Modeling Language) с помощью диаграмм детально описывает архитектуру системы. Созданная модель делает более лёгким процесс разработки и изменения программной системы, обеспечивает учёт всех технических требований к ней.

UML представляет собой относительно открытый стандарт, находящийся под управлением группы OMG (Object Management Group – группа управления объектами), открытого консорциума компаний. Группа OMG была сформирована для создания стандартов, поддерживающих межсистемное взаимодействие, в частности взаимодействие объектно-ориентированных систем. Возможно, группа OMG более известна по стандартам CORBA (Common Object Request Broker Architecture – общая архитектура посредников запросов к объектам).

Уже существует путь программного продукта от замысла до создания машинного кода с помощью UML-диаграмм и, например, пакета Rational Rose.

Для большинства программистов при создании программных систем более очевидна необходимость процесса создания кода, чем моделирования самой системы. К тому же предварительное создание модели системы включает в себя дополнительные трудозатраты, результат которых виден только через некоторое время, и это притом, что освоение сложных CASE-средств требует значительных усилий.

Используя CASE-мышление, программист уже сделает свои программы лучше, поскольку многие программисты не знают признаков неудачной программной архитектуры. А вот представление программных объектов в диаграммах UML позволяет наглядно увидеть ошибки и недоработки в полученной иерархии, легко этой иерархией манипулировать, что при ручном кодировании программист вряд ли может осуществить.

Для того чтобы перейти к созданию и сопровождению кода при помощи CASE-средства, поддерживающего язык UML, такого, например, как Rational Rose, программист должен перестроить свое представление о создании программ. Необходимо мыслить уже в терминах языка UML, мыслить диаграммами, а переход к такому типу мышления требует примерно такого же усилия, как переход от процедурного программирования к объектно-ориентированному.

Чтобы разрешить эту сложную ситуацию, Стив Меллор (Steve Mellor) и Мартин Фаулер (Martin Fowler) независимо пришли к определению трех режимов использования UML разработчиками: режим эскиза, режим проектирования и режим языка программирования.

Безусловно, самый главный из трех – это режим использования UML для эскизирования. В этом режиме разработчики используют UML для обмена информацией о различных аспектах системы. В режиме проектирования можно использовать эскизы при прямой и обратной разработке. Сущность эскизирования, или эскизного моделирования, в избирательности. В процессе прямой разработки вы делаете наброски отдельных элементов программы, которую собираетесь написать, и обычно обсуждаете их с некоторыми разработчиками из вашей команды. При этом с помощью эскизов вы хотите облегчить обмен идеями и вариантами того, что вы собираетесь делать.

Предлагается искусственно интеллектуальный подход к разработке UML модели на примере построения диаграмм классов.

Отсюда в настоящей работе рассматривается задача синтеза структурной модели информационной технологии: синтез диаграммы классов.

Примером может послужить синтез диаграмм классов:

1) построение всевозможных моделей диаграммы классов;

2) построение оптимальной модели.

В первом случае выбор подходящего варианта из представленных диаграмм-классов остаётся за проектировщиком.

Общая схема процесса приведена на рисунке 1.



Рисунок 1. – Укрупненная схема процесса проектирования.

Если поставить задачу построения оптимальной диаграммы, то необходимо к описанию добавлять соответствующие правила принятия решений. Например, в первую очередь выбирать отношение наследования, затем – отношение агрегации.

Синтез приведет к структурному решению как описание процесса проектирования, а последний блок будет выводить геометрическое представление диаграммы.

Структура данных является составной частью системы описания и необходима при формировании базовых концепций интеллектуальной обработки информации о какой-либо предметной области в той же степени, что и логика предикатов. Семантика, описывающая структурные свойства объекта предметной области, может быть выражена базовыми элементами структуры и формулами структуризации [2]. Базовые структурные элементы предметной области могут рассматриваться как элементарные данные, имеющие собственный, присущий только им смысл, их нельзя расчленить на более мелкие элементы. Первоочередная проблема заключается в том, какими средствами можно сделать такую структуру достаточно ясной и устойчивой. Нас в первую очередь интересуют конструктивные методы. Конструктивность заключается в том, что сначала задается некоторое множество элементарных данных структурной модели программы, затем на основе этого множества конструируется новое множество, элементы которого устроены более сложно, чем элементы исходного множества. Для такого конструирования множеств данных структурной модели программы необходимо аксиоматически определить соответствующие правила. Следовательно, если создать формальную систему, которая в качестве механизма структуризации данных включает в себя правила их построения, отражающие логику свойств структурной модели предметной области, то будет получена именно та система, которая нам нужна, т.е. система обработки описаний с четким теоретическим фундаментом.

### Заключение.

1. Разрабатывается формальная система порождения структурных решений, которая позволяет описать пошаговый процесс построения структурных решений по предварительно заданной системе понятий.

2. Предлагается рассматривать структурную семантическую модель предметной области как логическую модель описания наследственно конечной иерархической структурной надстройки над моделью последовательности простых данных, характеризующей последовательность базовых данных вычислителя.

### Список литературы

- 1) Барвайс Дж. Справочная книга по математической логике: В 4-х частях, Ч.I. Теория моделей. М.: Наука, Главная редакция физико-математической литературы. 1982. 392 с.
- 2) Осуга С. Обработка знаний. М.: Мир. 1989. 289 с.
- 3) Шрейдер Ю.А. Типология как основа интеграции знаний // НТИ. 1981. №11. С.1-5.