

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ DSM ДЛЯ СОЗДАНИЯ ДИНАМИЧЕСКИ НАСТРАИВАЕМЫХ СИСТЕМ*

Л.Н. Лядова¹

В данной статье рассматривается перспективный подход к созданию информационных систем, ориентированный на построение и использование предметно-ориентированных моделей. Представлена CASE-система METAS, предназначенная для создания адаптируемых информационных систем, основанная на применении многоуровневых моделей, управляющих функционированием информационной системы. Базовые понятия, лежащие в основе реализации CASE-системы, – это метамодель и метамоделирование. В работе рассматриваются основные преимущества, обеспечиваемые применением технологии DSM (Domain-Specific Modeling) и подхода MDSD (Model-Driven Software Development) для создания информационных систем различного назначения. Описаны основные возможности CASE-системы METAS и подходы к их реализации.

Введение

Одним из ключевых требований, предъявляемых к современным информационным системам (ИС), является возможность их динамической настройки на меняющиеся условия эксплуатации и потребности пользователей и бизнес-процессов. Наличие средств, обеспечивающих настройку ИС, определяет такие их характеристики, как стоимость разработки и сопровождения, время, необходимое на адаптацию к потребностям бизнес-систем, деятельность которых автоматизируется этими ИС.

Понятие *адаптируемости* рассматривается достаточно широко: в него включают такие взаимосвязанные нефункциональные требования к ИС как способность к развитию, гибкость, расширяемость, интероперабельность и т.п. Это делает его не только интересным для исследования, но и критичным свойством в практике создания ИС,

* Работа выполнена при поддержке РФФИ (проект № 08-07-90006-Бел_а)

¹ 614990, г. Пермь, ул. Букирева, 15, ПГУ, lnl@pm.ru

определяющим эффективность вложений в их разработку и внедрение, эксплуатацию и сопровождение, гарантирующим «живучесть» ИС. *Адаптация информационных систем* – это процесс их настройки на меняющиеся условия эксплуатации и потребности пользователей и бизнес-процессов как при создании новых систем, так и при сопровождении существующих. Этот итеративный процесс, требующий соответствующей поддержки, можно считать важнейшей частью жизненного цикла ИС.

Наиболее перспективным подходом к разработке технологий, инструментальных средств создания адаптируемых информационных систем является подход, в основе которого лежат понятия метамодели и метамоделирования [Almeida et al., 2004].

Максимальная гибкость при создании ИС достигается, если работа системы строится на *использовании моделей, которые могут изменяться в процессе функционирования системы, управляющих ее поведением*. В данной работе представлен подход к созданию адаптируемых информационных систем, основанный на построении *многоуровневых моделей и использовании метаданных*, представляющих информационные системы и их окружение с различных точек зрения и на различных уровнях детализации.

Создание адаптируемых систем предполагает использование соответствующих инструментальных средств, поддерживающих предъявляемые к ИС требования. Таким образом, можно считать, что свойство адаптируемости является необходимым не только для разрабатываемых систем, но и для применяемых при их разработке инструментальных средств.

1. Метамоделирование и CASE-системы

Модель системы – это абстрактное описание характеристик системы, важных с точки зрения цели моделирования, ее поведения на некотором формальном языке. При создании крупной ИС нельзя ограничиваться созданием только одной модели: если система сложная, то учет всех ее характеристик в одной модели приведет к чрезвычайной сложности модели. Таким образом, предполагается использование совокупности моделей, которые могут быть практически независимыми друг от друга и позволят сделать акценты на разных характеристиках и аспектах поведения системы при решении различных задач поддержания ее жизненного цикла.

Существуют различные определения метамодели. Исходя из того, что модели, создаваемые при разработке ИС, должны быть описаны на каком-либо формальном языке – языке моделирования, мы будем считать, что

метамодель – это модель языка моделирования, применяемого для формализации описания системы. Таким образом, *метамоделирование* – это возможность создавать и изменять метамодель, то есть тот язык (включая его семантику), на котором строятся модели ИС.

Классический вариант метамоделирования при создании ИС представляет четырехуровневая иерархия моделей. В этой иерархии каждый вышестоящий уровень определяет язык для описания нижестоящего уровня.

При решении задачи моделирования ИС на самом низком уровне находятся *данные, описывающие состояние предметной области* ИС, т.е. это – *модель состояния*. Следующий уровень является *онтологической метамоделью* для этого уровня (онтологическая метамодель описывает предметно-зависимый язык моделирования) и содержит *модель предметной области* ИС. Расположенный над ним уровень определяет *лингвистическую метамодель* для расположенных ниже в иерархии уровней (*лингвистическая метамодель* описывает предметно-независимый язык моделирования): на этом уровне находится модель языка моделирования, с которым работают аналитики, разработчики, CASE-средства и пр. Самый верхний уровень определяет язык, на котором описываются лингвистические метамодели.

Число уровней при реализации конкретных систем может изменяться. На одном и том же уровне могут быть использованы различные языки, ориентированные на описания ИС с различных точек зрения, с ориентацией на решение различных задач.

В зависимости от количества уровней создаваемых моделей и способа их использования при разработке ИС информационные системы и технологии их создания можно разделить на несколько классов:

- В *традиционной информационной* системе «внутри» системы находятся только данные, описывающие состояние ее предметной области. Эти данные соответствуют некоторой модели предметной области, которая может быть описана на любом языке (в том числе и на естественном), определенном «вне» системы. Модель разрабатывается аналитиками, после чего разработчики реализуют ее при помощи выбранных инструментальных средств создания ИС, систем программирования. В случае изменения модели приходится переписывать и перекомпилировать исходные коды приложений системы.

- В *традиционных CASE-технологиях* модель предметной области определяется формально и находится «внутри» CASE-системы. *Модель описывается в терминах метамодели*, которая может быть определена на любом языке и реализуется с помощью CASE-средств. Изменение модели ведет к необходимости изменения кода, сгенерированного CASE-

системой. Изменение метамодели влечет за собой переписывание и перекомпиляцию самой CASE-системы, однако такие изменения происходят крайне редко. Современные CASE-средства предоставляют инструменты для создания и редактирования моделей, а также позволяют сгенерировать большую часть кода информационной системы. Программные компоненты, реализующие специфические для конкретной системы поведенческие и функциональные аспекты, дописываются чаще всего вручную. В случае изменения модели CASE-система позволяет заново сгенерировать код приложений ИС. После повторной генерации обычно требуется ручная доработка кода.

– *Информационные системы, управляемые метаданными*, обеспечивают более мощные возможности для динамической адаптации. В данном случае также используется три уровня моделей, однако построенная модель предметной области находится «внутри» информационной системы в процессе ее эксплуатации. Таким образом, программное обеспечение ИС выступает в роли *интерпретатора*, а модель – в роли «*управляющей системы*», задающей правила функционирования ИС. Недостатком такого подхода является то, что несколько снижается производительность системы в ходе ее эксплуатации. Кроме того, если отсутствует возможность подключения внешних программных компонентов, расширяющих функциональность системы, то страдает универсальность вследствие невозможности реализации специфических для конкретной системы функций, отражающих бизнес-логику предметной области. Соответственно, метамодель должна быть максимально мощной.

– *Технология DSM (Domain Specific Modeling) с генерацией кода* обеспечивает моделирование в терминах предметной области. В данном случае для решения каждой задачи применяется свой язык моделирования, в котором используются исключительно понятия и отношения из соответствующей предметной области ИС. Здесь используется уже *мета-метамодель*, которая реализуется Мета-CASE-средством. При помощи этого средства описывается *метамодель, которая определяет предметно-зависимый язык моделирования*. На основе этой модели генерируется CASE-средство, при помощи которого описывается модель предметной области и генерируется ИС. Мета-CASE-и CASE-средства могут быть объединены в одну CASE-систему. Использование предметно-зависимого языка (*Domain Specific Language, DSL*) позволяет существенно упростить процесс создания моделей предметной области, в котором могут принимать активное участие эксперты – специалисты в данной предметной области.

– *Технология DSM с интерпретацией метаданных* обеспечивает максимальные возможности адаптации. Данный вариант является комбинацией двух предыдущих. Метамодел, модель и данные ИС находятся «внутри» ИС. В этом случае CASE-средства позволяют создать модели и интерпретировать их в ходе эксплуатации системы (для этого разрабатываются специальные *run-time компоненты*). Чтобы данный подход оказался применимым на практике, необходимо чтобы мета-метамодел была максимально выразительной. Интерпретация сразу двух уровней метамоделей приводит к ощутимой потере производительности, однако при достаточной выразительности мета-метамоделей получается чрезвычайно гибкая система. Данный подход реализуется в представленной в данной работе CASE-технологии METAS, разрабатываемой в АНО науки и образования «Институт компьютинга».

2. Технологии создания адаптируемых ИС и DSM

При разработке программного обеспечения, управляемого моделями (*MDSD – Model-Driven Software Development*), внимание *фокусируется на создании моделей, а не на разработке программного кода*. Основное назначение моделей – визуализация, спецификация, конструирование и документирование ИС. Подход MDSD использует модели нескольких уровней для повышения степени абстракции и уменьшения сложности процесса разработки программного обеспечения через разделение его на несколько уровней, на каждом из которых используется собственная модель (метамодел). Поддержка создания и использования моделей предполагает реализацию средств создания и использования языков моделирования. Большинство инструментальных средств ориентировано на визуальное конструирование моделей. Метамоделирование повышает гибкость инструментов визуального конструирования, которая намного превосходит модифицируемость традиционных систем программирования, и адаптируемость создаваемых с их помощью ИС. Разрабатываемые модели трансформируются в процессе разработки ИС, их настройки.

Таким образом, технология создания адаптируемых информационных систем предполагает разработку инструментальных средств, которым полностью «подвластны» собственные модели (метамодел). Эти инструментальные системы должны удовлетворять следующим *требованиям*: соответствие стандартам, компонентная архитектура, предметная ориентация, модельная ориентированность [Bordin et al., 2007].

Стандарт языка нужен для обеспечения его унифицированного использования, совместимости разных инструментальных средств и

переносимости создаваемых с его помощью ИС на различные платформы. *Стандартизация* обеспечивает интероперабельность моделей. Однако особенности реализации одного и того же языка в разных системах (например, механизмы его расширения) приводят к появлению различных «диалектов» языка. Модели, созданные разными разработчиками или с помощью различных инструментов, могут значительно различаться. В качестве основного языка моделирования в настоящее время чаще всего выступает UML, а в качестве средства обеспечения совместимости предлагается использовать XMI (XML Metadata Interchange) – специальное приложение XML, предназначенное для решения этой задачи. *Многоплатформенность* гарантирует возможность переноса созданных решений на различные платформы. Однако для самих инструментальных средств она необязательна: одна из основных идей создания систем *MDA (Model Driven Architecture)* – это *кросс-платформенная генерация кода*, таким образом, инструментальные средства создания ИС могут работать в любой среде, где разработчик с их помощью может конструировать приложения для выполнения на нужной ему платформе.

Адаптируемая ИС не может быть «монолитной» – только компонентный подход к построению ИС, обеспечивает необходимую гибкость. Инструментальные средства компонентного конструирования также не должны быть «монолитными» с точки зрения их реализации – они должны являться «конструктором инструментов». Таким образом, *средства разработки ИС должны иметь не просто компонентную архитектуру – они должны иметь архитектуру, динамически изменяемую (расширяемую) их пользователем – разработчиком ИС*; они должны предоставлять разработчикам возможности «самонастройки», расширения собственной функциональности, подключения новых компонентов, реализующих возможность создания ИС с новыми характеристиками.

Предметная ориентация подразумевает использование специального языка (или языков), ориентированного на предметную область (DSL). Язык, ориентированный на метамоделирование (MOF – Meta Object Facility) уже предложен и находит широкое применение, как и UML. При этом во многих проектах используются свои, альтернативные варианты предметно-ориентированных средств метамоделирования.

3. Технология METAS создания адаптируемых систем

CASE-технология METAS (METAdata System) – это технология для создания систем, управляемых метаданными, основанная на принципах DSM. Ключевым моментом технологии является использование

взаимосвязанных многоуровневых моделей, метаданных, описывающих информационную систему, создаваемую для определенной предметной области, с различных точек зрения и на разных уровнях абстракции.

Основное отличие представленной в данной статье CASE-системы от большинства других CASE-средств, генерирующих код приложений на каком-либо языке программирования в соответствии с построенными моделями, описывающим предметную область, состоит в том, что система METAS использует это описание *во время эксплуатации ИС*, выполняя функции отображения данных, их поиска и обработки по определенным этими метаданными правилам, *интерпретируя* их [Лядова, 2007].

Метамоделирование и использование метаданных в режиме интерпретации дает возможность *гибкой динамической настройки разработанной ИС* в соответствии с меняющимися потребностями пользователей и условиями эксплуатации.

Модели представляют собой *формализованное описание ИС*, для разработки которого применяются инструменты визуального конструирования и предметно-ориентированные языки. Созданные модели транслируются во внутреннее представление, размещаемое в отдельной *базе метаданных*, используемое для настройки приложения на условия его эксплуатации в процессе его разработки, а затем – загрузки и выполнения.

С помощью метаданных описываются следующие аспекты ИС: объекты предметной области ИС и их поведение; визуальный интерфейс пользователя ИС; бизнес-операции и бизнес-процессы предметной области; первичные и сводные документы и отчеты и т.д. Каждая модель описывает определенную часть, «срез» ИС (некоторые модели могут описывать одни и те же компоненты ИС, но с различных точек зрения).

CASE-система предусматривает средства расширения собственной функциональности, «самонастройки», в частности, подключения новых программных компонентов, реализующих «нестандартную» функциональность, обеспечивающих перенос системы на различные платформы, а также средства расширения существующих моделей, создания предметно-ориентированных языков различного назначения и новых моделей на их основе, анализа, классификации и каталогизации документов и поддержки работы аналитиков и пр.

Список литературы

[Лядова, 2007] Лядова Л.Н. Технология создания динамически адаптируемых информационных систем // Труды междунар. науч.-техн. конф. «Интеллектуальные системы» (AIS'07). Т. 2. – М.: Физматлит, 2007.

[**Almeida et al., 2004**] Almeida J.P., Pires L.F., Sinderen M. Costs and Benefits of Multiple Levels of Models in MDA Development // Proc. Second European Workshop on Model Driven Architecture (MDA) with an emphasis on Methodologies and Transformations “Computer Science at Kent”. Canterbury, UK, 2004.

[**Bordin et al., 2007**] Bordin M., Vardanega T. A Domain Specific Metamodel for Reusable Object-Oriented High-Integrity Components // Proc. 7th OOPSLA Workshop on Domain-Specific Modeling. Montreal, Canada, 2007.