

УДК 004.94

С.Ю. Гавриленко<sup>1</sup>, В.Г. Иванов<sup>2</sup><sup>1</sup> Национальный технический университет "ХПИ", Харьков<sup>2</sup> Харьковский национальный технический университет радиоэлектроники, Харьков

## РАЗРАБОТКА СИСТЕМЫ ГРАФИЧЕСКОГО ОПИСАНИЯ И МОДЕЛИРОВАНИЯ РАСПРЕДЕЛЕННЫХ ПРОГРАММНЫХ ОБЪЕКТОВ ПРИ ПРОЕКТИРОВАНИИ ИНФОРМАЦИОННЫХ СИСТЕМ

Рассмотрена проблема построения среды дореализационного моделирования и тестирования для разработки программного обеспечения. Проанализированы возможности некоторых современных CASE-средств, и тенденции их дальнейшего развития. Выявлены сложности, возникающие при подключении механизмов имитационного моделирования к основным функциональным возможностям CASE-средств, предложены варианты их решения.

**Ключевые слова:** программное обеспечение, моделирование, тестирование, CASE-средства.

### Постановка проблемы и анализ литературы

Использование инструментальных средств во многом влияет на успех работы над проектом вне зависимости от сферы их применения, будь то управление проектом или автоматизация проектирования (кодирования). На данный момент существует большое количество различных инструментальных средств, которые постоянно развиваются как количественно, так и качественно. Но, как правило, именно качественные изменения в таких средствах встречаются с проблемами, решение которых требует большого количества ресурсов.

Для увеличения эффективности ведения бизнеса в современном мире широко используют различные инструментальные средства. Спектр сфер автоматизации в бизнес-процессах достаточно широк, поэтому инструментальные средства можно классифицировать в следующие группы:

1. Средства управления проектом.
2. Средства создания диаграмм.
3. Средства имитационного моделирования/анимации.
4. Средства создания информационных систем.
5. Интегрированные многофункциональные средства.

Для организаций, занимающихся разработкой сложных программных комплексов, в большинстве случаев наиболее востребованными являются инструментальные средства, принадлежащие к четвертой группе. К ним относятся (CASE-средства): Rational Software Architect (IBM), Power Designer (Sybase), S-Designer (PowerSoft), Umbrello UML Modeller и т.д.

Основные задачи таких средств следующие:

- формирование функциональной структуры (архитектуры) информационной системы;

- структурирование (моделирование) данных, в том числе: создание концептуальной модели структуры базы данных, автоматическая генерация физической модели БД и др.

- быстрая разработка приложений (визуальное программирование) [1].

Наиболее популярными среди средств создания информационных систем являются CASE-средства, которые акцентируют свое внимание на проектировании программного обеспечения. Рассмотрим возможности некоторых CASE-средств.

*Rational Software Architect 8.0* – последняя версия среды из линейки CASE-средств от фирмы IBM. Структура средства построена по модульному принципу: содержит ядро и набор модулей, которые могут подключаться к нему независимо. Ядро обеспечивает использование UML (UML2.2) и BPMN 2 нотаций, также поддерживает моделирование для языков Java, C# и VB.NET (Microsoft Visual Basic .NET). Наборы модулей содержат расширения:

- развернутого моделирования (Deployment modeling extension), включающие в себя темы, слои, схемы обратной связи;

- поддержки моделирования сервис-ориентированной архитектуры и WebSphere (SOA and WebSphere modeling extension) ;

- расширения для разработки на C++ (C++ extension), включающие в себя визуализацию, комбинированное моделирование, согласованное моделирование;

- моделирования коммуникационных приложений (Communications applications extensions), содержащие средства для ускорения разработки распределенных приложений на основе NGN (Next Generation Networks) сетей;

- каркаса интегрированной архитектуры (Integrated architecture frameworks extension), обеспечивающего поддержку моделирования архитектур

ных решений, связанных с бизнесом (организация, стратегии, цели и т.п.);

- динамической прогонки модели (Model simulation extension), позволяющего пошагово прогонять созданную модель на различных этапах моделирования [2].

*Power Designer* (CASE-средство от Sybase, последняя версия 15.3) поддерживает следующие технологии моделирования:

- моделирование данных (возможно отображение концептуальной, логической и физической модели данных) для проектирования хранилищ данных, основанное на IE (Information Engineering) методологии или нотации IDEF 1/x;

- моделирование приложений, используя язык графического описания для объектного моделирования UML и управление реализацией персистентных объектов за счет расширенной возможности описания объектно-реляционного отображения (Object-Relational mapping); применение техники моделирования с использованием языка XML и связь таких моделей с UML и моделями данных;

- моделирование бизнес процессов, включающих диаграммы описания бизнес процессов, интуитивно понятные нетехническим пользователям, моделирование языка исполнения бизнес процессов с поддержкой нотаций BPEL4WS, ebXML, BPMN 2;

- моделирование архитектуры предприятия от бизнес целей до реализации, с использованием технологии соединения и синхронизации (Link and Synch).

*Power Designer* поддерживает более 60 СУБД, платформы для разработки приложений, такие как Java J2EE, Microsoft .NET, Web Services и PowerBuilder, а также языки исполнения процессов, такие как ebXML и BPEL4WS [3].

### Тенденции развития современных CASE-средств

Описание структуры данных и процессов, происходящих в системе, является главной задачей, которую должно решать CASE-средство. На данный момент современные CASE-средства развиваются в сторону охвата более широкого спектра систем и процессов, которые поддаются описанию в их среде. Использование UML и BPMN 2 нотаций позволяют эффективно описывать как структуры данных, поведение так и бизнес процессы. Еще одной тенденцией развития является расширение поддерживаемых языков и платформ программирования для генерации программного кода. Данная тенденция обусловлена как большим количеством языков и платформ, так и спецификой их применения для конкретных проектов. Также, современные CASE-средства постепенно срачиваются со средами программирования, что позволяет увеличить эффективность разработки программного обеспечения. Процесс срачивания таких

средств не несет в себе окончательный характер. Таким образом, возможно рассматривать CASE-средства и средства для программирования ПО как отдельные среды, выполняющие различные задачи, но с возможностью подключения друг к другу.

В то же время, рабочее приложение включает в себя не только статическую структуру данных и описаний, но и динамику их изменения. Конечно, CASE-средства позволяют описывать и поведение системы, но моделирование поведения ограничивается в основном только его описанием. С этой точки зрения, CASE-средство Rational Software Architect 8.0 включило в свой состав качественно новый модуль динамической прогонки модели. Сам механизм прогонки позволяет только поэтапно (поэлементно) прогонять определенную диаграмму, что уменьшает количество ошибок, возникающих на этапе проектирования.

Развитием идеи динамической прогонки диаграмм является идея создания среды дореализационного моделирования и тестирования. Она должна быть интегрированной со средой разработки (проектирования) для того, чтобы использовать модели имплементируемой системы. Совокупность входных моделей, написанная на языке UML, должна быть преобразована в единую формализованную внутреннюю форму, после чего производится ее тестирование и (или) моделирование [4].

### Проблемы внедрения возможности имитационного моделирования в CASE-средства

Поскольку большинство самых серьезных ошибок в программных средствах закладывается именно на ранних этапах разработки, то целесообразно было бы добавить в CASE-средства возможность полноценного моделирования поведения системы – имитационное моделирование. Но в данной ситуации возникают как минимум четыре проблемы:

- существующие инструменты CASE-средств недостаточны для адекватного отображения динамики моделируемой системы;

- аппарат CASE-средств недостаточно формализован, что не позволяет проводить полноценное имитационное моделирование

- отсутствие аппарата, реализующего имитационное моделирование;

- отсутствие необходимого набора входных данных для проведения имитационного моделирования.

Для решения первой проблемы можно использовать язык графического моделирования UML, который содержит в себе весь необходимый инструментарий для описания разрабатываемой системы с различных точек зрения. Так, для наиболее полного описания системы достаточно использовать модели трех типов:

- модель классов;
- модель состояний;
- модель взаимодействия.

Каждый тип диаграмм описывает отдельную сторону системы. Модель классов представляет статические, структурные аспекты системы, связанные с данными. Модель состояний представляет временные, поведенческие, управленческие аспекты системы. Модель взаимодействия представляет кооперацию отдельных объектов, другими словами, все аспекты системы, связанные с взаимодействиями.

Благодаря объектно-ориентированному подходу UML обладает рядом характерных свойств, таких как индивидуальность, классификация, наследование, полиморфизм, абстракция и т.д. UML позволяет также описывать иерархическую структуру (подсистемы с различными интерфейсы, уровни, разделы), представления готовых решений (образцы, каркасы, библиотеки), основные типы управления программной средой (внешнее управление, внутреннее управление) и т.д. [5].

Решение второй и третьей проблемы требует синтеза возможностей CASE-средства с возможностями средств имитационного моделирования, то есть с возможностями другого типа инструментального средства, например системы моделирования CPN Tools.

CPN Tools – совокупность средств имитационного моделирования, использующих аппарат Раскрашенных Сетей Петри (Colored Petri Nets). Аппарат CPN позволяет моделировать:

- распределенные и параллельные процессы;
- синхронное и асинхронное взаимодействие;
- детерминированные и стохастические системы;
- как с учетом временных параметров, так и без.

CPN модель – это выполняемое представление системы, состоящее из состояний, событий и переходов. События и переходы обеспечивают изменение состояний внутри модели. Симуляция такой модели позволяет рассмотреть поведение самой системы. Не смотря на не большой словарь функциональных элементов, аппарат CPN является достаточно гибким при составлении модели. Использование высокоуровневых языков программирования вместе с аппаратом CPN позволяет одновременно организовывать сложную структуру данных и обеспечить возможность моделирования сложных в поведенческом плане систем (параллелизм, стохастичность, распределенность, модульность и т.п.).

CPN модель может представляться в виде ориентированного графа с двумя типами вершин. Данное свойство аппарата CPN позволяет наглядно представлять модели, что улучшает читабельность и способствует более быстрому составлению модели [6].

Следовательно, для реализации возможности имитационного моделирования для CASE-средств

необходимо в качестве базовой модели системы иметь совокупность UML диаграмм, которая впоследствии будет преобразована в CPN модель. Но преобразованная CPN модель сама по себе не может решить проблему отсутствия необходимого набора данных для проведения имитационного моделирования. Причинами, усложняющими решение данной задачи, являются:

- даже максимально полная UML модель может не содержать всех необходимых сведений для проведения имитационного моделирования;
- наличие дополнительных параметров (настроек) для проведения имитационного моделирования.

Если первая причина обусловлена различием в специфике применения языков моделирования, то вторая – необходимое условие для проведения имитационного моделирования. Следовательно, решением данной проблемы является возможность редактирования CPN модели и задание параметров моделирования вручную. Таким образом, для проведения имитационного моделирования разрабатываемого программного обеспечения, CASE-средство должно включать в себя специализированный модуль со следующими особенностями:

- в качестве входных данных принимается модель системы, описанная с помощью языка объектно-ориентированного моделирования UML;
- входная UML-модель автоматически преобразовывается в CPN-модель;
- в случае неполных данных, пользователь может вручную изменить (дополнить) CPN-модель;
- пользователь вручную задает параметры моделирования;
- по результатам имитационного моделирования среда создает отчет.

Из всех перечисленных характеристик среды особое внимание следует уделить автоматическому преобразованию UML-модели в CPN-модель, поскольку само преобразование словарей и схем нотаций является сложной задачей. Как следствие, потребуются дополнительные исследования в этом направлении.

## Выводы

В данной статье была рассмотрена проблема построения среды дореализационного моделирования и тестирования в контексте существующих инструментальных средств для разработки ПО. Были проанализированы возможности некоторых современных CASE-средств, и тенденции их дальнейшего развития. Особое внимание уделили возможности динамической прогонки модели, которая является новаторской для существующих CASE-средств.

В основной части статьи были рассмотрены сложности, возникающие при подключении механизмов имитационного моделирования к основным функциональным возможностям CASE-средств, и

варианты их решений. Это позволило получить список характеристик, которыми должен обладать модуль, занимающийся имитационным моделированием разрабатываемой системы до этапа ее реализации.

### Список литературы

1. Силич В.А. Системные технологии проектирования бизнес-процессов: учебное пособие / В.А. Силич, М.П. Силич. – Томск: ТПУ, 2000. – 108 с.
2. What's new in IBM Rational Software Architect 8.0. [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.ibm.com/developerworks/rational/library/10/whats-new-in-rational-software-architect-8/index.html>.
3. Сайт официального дистрибьютора Sybase. [Электронный ресурс]. – Режим доступа к сайту: <http://www.sybase.ru/products>.

4. Иванов В.Г. Разработка среды дореализационного моделирования и тестирования программного обеспечения / В.Г. Иванов, А.М. Нычик // Системы обработки информации. – X: ХУ ПС, 2010. – Вып. 9(90). – С. 40-43.

5. Рамбо Дж. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо, М. Блаха. – [2-е изд.]. – СПб.: Питер, 2007. – 544 с.

6. Vijay Gehlot. Carmen Nigro. An introduction to systems modeling and simulation with colored petri nets [Электронный ресурс] / Vijay Gehlot. – Режим доступа к ресурсу: <http://www.informs-sim.org/wsc10papers/012.pdf>.

Поступила в редколлегию 18.11.2013

**Рецензент:** д-р техн. наук, проф. И.В. Гребенник, Харьковский национальный университет радиоэлектроники, Харьков.

## РОЗРОБКА СИСТЕМИ ГРАФІЧНОГО ОПИСУ І МОДЕЛЮВАННЯ РОЗПОДІЛЕНИХ ПРОГРАМНИХ ОБ'ЄКТІВ ПРИ ПРОЕКТУВАННІ ІНФОРМАЦІЙНИХ СИСТЕМ

С.Ю. Гавриленко, В.Г. Іванов

Розглянута проблема побудови середовища дореалізаційного моделювання і тестування для розробки програмного забезпечення. Проаналізовані можливості деяких сучасних case-засобів, і тенденції їх подальшого розвитку. Виявлені складнощі, що виникають при підключенні механізмів імітаційного моделювання до основних функціональних можливостей case-засобів, запропоновані варіанти їх рішень.

**Ключові слова:** програмне забезпечення, моделювання, тестування, case-засоби.

## DEVELOPMENT OF SYSTEM OF GRAPHIC DESCRIPTION AND DESIGN OF THE UP-DIFFUSED PROGRAMMATIC OBJECTS AT PLANNING OF INFORMATIVE SYSTEMS

S.G. Gavrilenko, V.G. Ivanov

The problem of construction of environment of prior to realization design and testing is considered for software development. Possibilities of some modern case-facilities, and their further progress trends, are analysed. Complications, arising up during connecting of mechanisms of imitation design to basic functional possibilities of case-facilities, are exposed, the variants of their decision are offered.

**Keywords:** software, designs, testing, case-facilities.