РАЗРАБОТКА ВЫСОКОУРОВНЕВОЙ СРЕДЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ JaSim

О. А. Савина, А. С. Погорелов (Орел)

Сложность разрабатываемых в настоящее время имитационных моделей постоянно возрастает, все более приближаясь к сложности реальных объектов. В настоящее время существует большое количество различных средств имитационного моделирования (ИМ), многие из которых изначально разрабатывались для решения конкретных прикладных задач. В целом можно выделить несколько основных направлений, по которым развиваются средства имитационного моделирования [1].

Библиотеки поддержки имитационного моделирования. Данные средства служат инструментальным расширением базового языка программирования. Примерами таких библиотек служат C++Sim (базовый язык C++), SMPL/EML (базовый язык Pascal/Java) и SimPy (базовый язык Phyton).

Языки имитационного моделирования. Одними из первых средств имитационного моделирования были языки, такие как GPSS, SIMSCRIPT, SIMULA 67. Данные средства кроме специализированного языка включают также инструментарий для проведения экспериментов и сбора статистической информации.

Интегрированные среды. К отличительным особенностям интегрированных сред моделирования (Arena, AnyLogic, iThink и др.) следует отнести наличие средств анализа предметной области и синтеза моделей, визуализацию процесса выполнения имитационной модели в реальном времени, развитые возможности статистического анализа результатов моделирования.

Развитие распределенного и агентного моделирования открывает большие перспективы по созданию высокоуровневых (high level) имитационных моделей. Их компонентами являются логически-обособленные имитационные модели отдельных подсистем (тогда как в последовательном ИМ составляющими являются объекты, а в распределенном ИМ – логические процессы).

В США с 1995 г. ведутся исследования по созданию инструментария, обеспечивающего повторное использование и взаимодействие разнородных имитационных моделей в целях уменьшения времени и стоимости разработок [2]. В результате в 2000 году была принята серия стандартов HLA (High Level Architecture) [3–5]. В то же время проблемы, возникающие при создании высокоуровневых моделей, только отчасти решаются применением различного рода стандартов в области имитационного моделирования. Прежде всего, это обусловлено тем, что данные стандарты ориентированы в основном на использование таких языков программирования как Java или C++ и, как следствие, они имеют слабую поддержку со стороны интегрированных инструментов моделирования.

Построение высокоуровневых моделей является, как правило, результатом работы нескольких людей или коллективов. Это приводит к необходимости обеспечения совместимости её различных компонентов.

Разработанная высокоуровневая среда имитационного моделирования JaSim обеспечивает возможности интеграции имитационных моделей, реализованных в разнородных средах (в частности, на языке GPSS и EML/Java). Среда построена в соответствии со стандартом HLA, в котором ключевыми понятиями являются Federation и Runtime Infrastructure (RTI). Federation – это объединение (федерация) компонентов ИМ, каждый из которых является «федератом» (Federates). Между федератами осуществляется обмен данными с помощью программной оболочки (инфраструктуры) RTI. Эта инфраструктура обеспечивает также исполнение федератов в едином модель-

ном времени, т. е. она, по сути, является распределенной операционной системой для федерации. Для взаимодействия с RTI федераты вызывают сервисы (servises), которые обеспечивают управление федерацией, декларациями, объектами, правом доступа, распределением данных и временем.

В структуре разработанной среды JaSim можно выделить несколько уровней:

- базовый уровень – уровень выполнения (библиотеки моделирования);
- трансляционный уровень;
- уровень описания модели;
- уровень интегрированной среды.

Базовый уровень – уровень выполнения (библиотеки моделирования)

Большинство современных средств моделирования предоставляет набор концепций, составляющих основу (структурный каркас) для разработки и описания имитационных моделей, являясь в то же время основой программной реализации этих концепций. Для описания структуры в основном используется объектно-ориентированный подход, позволяющий получить прозрачную и четко структурированную базу для создания имитационных моделей.

В разработанной системе в качестве базовой библиотеки моделирования используется Java/EML [6]. Назначение библиотеки моделирования – реализация работы с базовыми элементами имитационной модели (очереди, списки, устройства) и управление модельным временем, событиями, процессами. Эта библиотека может использоваться как непосредственно для создания имитационной модели, так и в качестве основы выполнения для моделей, составленных на специализированном языке высокого уровня.

Основой внутренней организации библиотеки Java/EML является объектная структура, описывающая базовые строительные элементы модели. Основой является класс Model, отвечающий за инициализацию модели, запуск и управление ходом моделирования. Классы Queue, Device отвечают за очереди и средства, соответственно. Класс Schedule управляет модельным временем и отвечает за регистрацию событий. Класс Random содержит генератор псевдослучайной последовательности, а также набор методов – функций распределения.

Библиотеки моделирования являются мощнейшим инструментом для реализации имитационных моделей. Однако отрицательным моментом непосредственного использования таких библиотек является сложность их изучения и применения. Кроме того, при разработке имитационных моделей необходима жесткая привязка к конкретному языку программирования и учет его особенностей. Например, библиотеки, выполненные с использованием принципов объектно-ориентированного программирования, обязывают разработчика модели также придерживаться этого подхода.

Для преодоления этих недостатков в среде JaSim предусмотрены альтернативные возможности построения моделей: модели могут быть реализованы либо на языке GPSS, либо с помощью базовой библиотеки. Таким образом, среда JaSim дает возможность разработчику включать в модель ранее созданные модели на языке GPSS, создавать новые модели на основе процессного подхода в среде GPSS, или формировать дискретно-событийные модели с помощью базовой библиотеки.

Модели, реализованные на специализированном языке ИМ более выразительны. Они не перегружены информацией, не относящейся непосредственно к структуре или логике функционирования, в отличие от модели, реализованной на базовом языке, которая более сложна для понимания и интерпретации. Модель, разработанная на специализированном языке, перед использованием приводится в исполняемый вид.

Трансляционный уровень

Среда JaSim включает транслятор в виде отдельного инструмента, основной функцией которого является перевод модели со специализированного языка на базовый. Полученный код модели при необходимости исправляется и/или дополняется необходимой функциональностью, после чего модель компилируется в исполняемый файл.

Такой подход обеспечивает возможность работы с имитационной моделью на уровне языка программирования, что позволяет выполнить интеграцию имитационной модели и других программных средств на стадии разработки. В то же время необходимость создания промежуточной модели усложняет процесс разработки, что не всегда оправданно. В связи с этим была предусмотрена возможность использования смешанного подхода, при котором специализированный язык дополняется расширениями, позволяющими встраивать блоки кода базового языка программирования непосредственно в текст модели на специализированном языке. Это позволяет в некоторых случаях обойти ограничения специализированного языка за счет некоторого снижения выразительности и читаемости модели.

В рассматриваемой системе в качестве основы специализированного языка используется GPSS. Реализация транслятора выполнена в среде ANTLR [7] (ANother Tool for Language Recognition), который является генератором синтаксических анализаторов и обладает следующими особенностями:

– описание правил разбора при помощи расширенной формы Бекуса–Наура;

– возможность генерации кода на языках C++, Java, Python, C#;

– возможность создания лексических, синтаксических анализаторов, а также абстрактных синтаксических деревьев.

Уровень описания модели

На протяжении длительного времени имитационные модели разрабатывались в текстовом виде. Развитие программного обеспечения предоставило возможность визуального (графического) проектирования. В зависимости от используемого подхода это может быть блок-схема GPSS, Q-схема (при процессном подходе) или событийный граф (при событийном подходе), узлы которого представляют события, а дуги – переходы между ними. Графическое представление позволяет наглядно описать разрабатываемую модель. В случае с блок-схемами GPSS, где каждый блок имеет свое графическое обозначение, модели в текстовом и графическом представлении практически эквивалентны. Однако, например, описания модели только при помощи графа событий недостаточно. В таких случаях недостающее описание модели описывается в текстовом виде.

В разработанной системе используется как текстовое, так и графическое представление имитационных моделей. Текстовый редактор выполнен на основе стандартной Java библиотеки Swing [8]. Для графического представления используются средства визуализации графов Graphvis [9].

Уровень интегрированной среды

Интегрированная среда позволяет собрать воедино все используемые разработчиком средства и инструменты: уровня описания, уровня трансляции и базового уровня (уровня выполнения моделей). Интеграция различных средств формирования моделей в интерактивном режиме обеспечивает значительное повышение эффективности работы по созданию имитационных моделей.

Взаимосвязь между средствами описания и трансляции модели позволяет выявить большинство ошибок разработчика еще на стадии описания. Связь уровня описа-

ния и базового уровня позволяет выполнять отладку модели в интерактивном режиме. В JaSim интеграция компонентов всех рассмотренных уровней позволяет реализовать дополнительные возможности визуального иструментирования, предназначенного для выполнения кода модели средствами отладки/профилирования модели и сбора статистической информации. На уровне интегрированной среды используется консервативный подход к синхронизации модельного времени.

Выводы

Разработанная высокоуровневая среда имитационного моделирования JaSim:

- позволяет при построении широкомасштабных моделей осуществлять интеграцию имитационных моделей, реализованных на различных языках;
- дает возможность создавать новые модели на основе процессного подхода в среде GPSS (или использовать ранее созданные модели), либо формировать дискретно-событийные модели с помощью базовой библиотеки Java/EML;
- позволяет реализовать все преимущества объектно-ориентированного подхода вследствие перевода модели в процессе трансляции со специализированного языка GPSS на базовый;
- использует архитектурно-нейтральный язык Java, что обеспечивает независимость от конкретной платформы.

Литература

1. **Кельтон В., Лоу А.** Имитационное моделирование. Классика. СПб.: Питер, 2004. 847 с.
2. **Fujimoto R. M.** Time management in the high level architecture//Simulation. 1998. Vol. 71, N 6. P. 388–400.
3. IEEE Std P1516. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) _ Framework and Rules. N.Y.: Institute of Electrical and Electronics Engineers, Inc., 2000.
4. IEEE Std P1516.2. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) _ Object Model Template (OMT) Speci_cation. N.Y.: Institute of Electrical and Electronics Engineers, Inc., 2000.
5. IEEE Std P1516.3. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) _ Federation Development and Execution Process. N.Y.: Institute of Electrical and Electronics Engineers, Inc., 2000.
6. **Савина О. А., Лазарев С. А.** Язык имитационного моделирования систем EML (Event Modeling Language)//Сборник научных трудов ученых Орловской области. Вестник науки. Вып. 5. В 2-х томах. Т. 2. Орел: ОрелГТУ, 1999. С. 232–238.
7. ANTLR Parser Generator/www.antlr.org.
8. A Swing Architecture Overview java.sun.com/products/jfc/tsc/articles/architecture.
9. Graphviz – Graph Visualization Software / www.graphviz.org.