

# ОПЫТ ПОСТРОЕНИЯ СРЕДЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ РАСПРЕДЕЛЁННЫХ СИСТЕМ С ПРИМЕНЕНИЕМ HLA

## EXPERIENCE IN BUILDING SIMULATION ENVIRONMENT OF DISTRIBUTED SYSTEMS USING HLA

**В. В. Балашов, Д. Ю. Волканов, Р. Л. Смелянский, М. В. Чистолинов**  
(Москва)

### **Введение**

При анализе функционирования распределённых систем, состоящих из множества узлов, выделяют ряд задач. При анализе морских систем такими задачами являются:

- управление движением морскими объектами (в порту или открытой акватории);
- отработка группового взаимодействия, например в группе «авианосец и самолёты»;
- анализ функционирования бортового вычислителя морской системы.

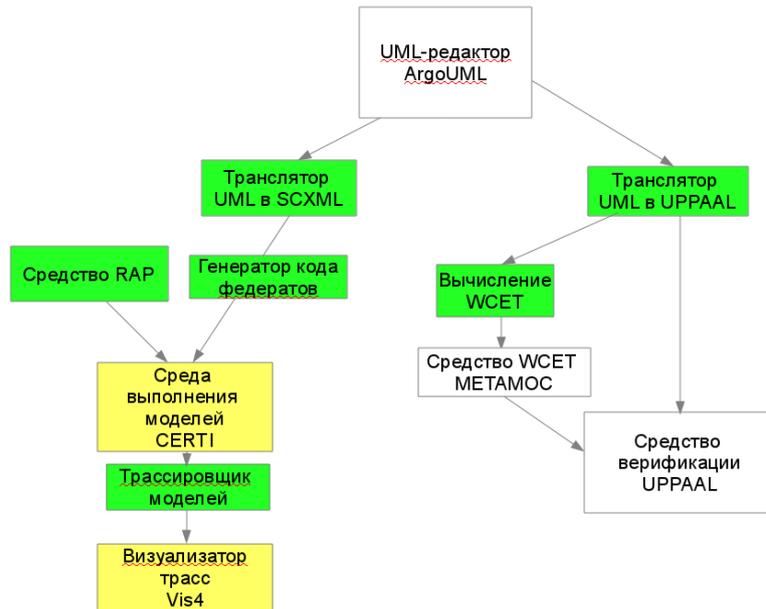
Для решения данных задач часто применяется имитационное моделирование. Имитационное моделирование – это метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему, с которой проводятся эксперименты с целью получения информации об этой системе. Начиная с 1994 года, в лаборатории вычислительных комплексов факультета ВМК МГУ ведётся разработка системы моделирования ДИАНА, совместимой с международными стандартами имитационного моделирования и современными методами анализа распределённых систем. Описание опыта применения этой среды моделирования для решения вышеуказанных задач приведено в этой работе.

### **1. Архитектура системы моделирования ДИАНА**

Для обеспечения интероперабельности разрабатываемой системы моделирования (СМ) СМ необходимо, чтобы модели, создаваемые в СМ, были совместимы со стандартом HLA [1]. В этом стандарте отдельные участники имитационного эксперимента, вне зависимости от их типа (программа, человек, аппаратное устройство), называются федератами. Совокупность федератов образует федерацию. Каждый федерат подключается к инфраструктуре RTI (Run-Time Infrastructure), которая обеспечивает их синхронизацию, реализуя, таким образом, функции среды выполнения. Фактически стандарт HLA описывает интерфейс между средой выполнения RTI и участниками моделирования (федератами).

Интерфейс HLA содержит лишь набор низкоуровневых примитивов, облегчающих сопряжение имитационных моделей, предоставляемых различными разработчиками. Однако разработка новых моделей с применением лишь примитивов стандарта сложна и чревата ошибками. Для этой цели более пригоден язык высокого уровня. В качестве такого языка был выбран универсальный язык моделирования UML [2], позволяющий описывать как структуру системы, так и динамические аспекты её поведения.

Архитектура разработанной системы моделирования приведена на рис. 1. Белым цветом обозначены открытые средства, используемые в СМ без модификаций, жёлтым – средства, модифицированные для разрабатываемой СМ, а зелёным – средства, полностью разработанные и реализованные для СМ ДИАНА.



**Рис. 1. Архитектура среды моделирования ДИАНА**

В состав среды моделирования входят следующие подсистемы:

- редактор диаграмм состояний UML; в качестве редактора используется средство ArgoUML;
- транслятор диаграмм UML в исполняемые модели, совместимые со стандартом HLA. Он состоит из двух подсистем: транслятора UML в SCXML и генератора кода федератов;
- среда выполнения моделей на основе системы CERTI [3], дополненная средствами для поддержки моделирования PBC PB, интеграции с натурными каналами и интеграции с библиотекой времени компиляции Proto-X, кодирующей данные с использованием встроенных типов языка C++;
- средство трассировки, представляющее собой специальный федерат HLA; оно позволяет записывать в трассу в формате OTF [4] события, возникающие в ходе моделирования;
- средство анализа и визуализации трасс, обеспечивающее интеграцию с форматом описания трасс OTF;
- средство верификации UPPAAL [5], предназначенное для проверки свойства поведения объектов, представленных в виде сетей временных автоматов. Оно снабжено транслятором, преобразующим диаграммы состояний UML в сети временных автоматов с учетом оценок наихудшего времени выполнения программы, полученных с помощью средства Metamos;
- интегрированная среда разработки, объединяющая все указанные выше средства и запускающая процедуру моделирования. Также среда разработки интегрирована со средством для решения задачи выбора оптимального набора механизмов обеспечения отказоустойчивости (RAP).

## **2. Методика использования системы моделирования ДИАНА**

При решении задач управления морскими объектами и отработки взаимодействия группы «авианосец и самолёты», рекомендуется придерживаться типичной последовательности действий, приведенной ниже.

В виде моделей рекомендуется описать поведение морского объекта, его реакцию на различные внешние воздействия. При помощи верификатора можно проверить свойства корректности и безопасности поведения. Примером такого свойства может быть проверка невозможности ситуации, когда к занятому причалу пытается причалить ещё одно судно. Результатом моделирования является файл трассы, по которому можно проследить изменение параметров движения в процессе функционирования системы.

Основным средством, реализующим взаимодействие всех созданных инструментов с пользователем, является интегрированная среда разработки моделей. Предполагается, что все остальные средства запускаются через ее интерфейс, если явно не указано противное.

На первом этапе разработки модели обычно проводится описание её поведения в виде диаграммы состояния UML. Для этого необходимо воспользоваться редактором ArgoUML. На данном этапе необходимо определить цель моделирования и построить диаграммы состояний UML так, чтобы конструируемая модель была корректна (то есть исследуемые характеристики объекта в модели должны быть эквивалентны своим образам в исходном объекте) и адекватна (то есть ей присущи лишь характеристики, существенные для цели моделирования). В диаграмму могут быть включены фрагменты кода на C++. После редактирования модели необходимо средствами ArgoUML экспортировать полученные диаграммы в формат XMI.

На следующем этапе проводится проверка того, что модель удовлетворяет некоторым заданным свойствам. Этап необходим при исследовании сложных моделей поведения, корректность которых должна быть строго доказана. Для проведения верификации необходимо преобразовать UML-диаграммы, представленные в формате XMI, в сети временные автоматы UPPAAL с помощью разработанного транслятора.

Если транслятор не обнаружил ошибок, то можно приступить к верификации модели с помощью средства UPPAAL [5]. Для этого проверяемое свойство поведения диаграмм UML необходимо задать в виде формулы темпоральной логики TCTL. Примером такого свойства может служить такое свойство: *как только возникла проблема в машинном отделении, сигнал alarm зазвучит сразу и будет звучать не менее 5 секунд*. После задания интересующих свойств запускается процедура верификации. Верификатор проверяет выполнимость заданной формулы в дереве трасс вычислений сети временных автоматов, соответствующих диаграммам состояний UML. Если свойство не выполнено, то верификатор строит контрпример – одну из трасс, на которых проверяемое свойство нарушается. Этот контрпример можно конвертировать в трассу переходов UML с конкретными значениями параметров и таймеров. С временными автоматами можно работать и непосредственно в графическом интерфейсе средства UPPAAL.

Затем можно приступить к трансляции модели из UML-представления в код на C++. После того, как код федератов получен, можно запустить имитационный эксперимент в среде CERTI.

Результатом моделирования является файл, содержащий трассу эксперимента в формате OTF, а также несколько служебных файлов, просмотр которых может быть полезен для отладки моделей. Для просмотра и исследования трассы эксперимента используется разработанное нами средство анализа и визуализации трасс. Оно позволяет получать информацию о событиях модели и информационных обменах между ее компонентами, осуществлять поиск события, навигацию по трассе, масштабирование трассы. В зависимости от результатов анализа трассы можно вернуться на один из предыдущих этапов, изменить модель и повторить эксперимент.

### 3. Апробация системы моделирования ДИАНА

Помимо моделирования поведения морских объектов СМ ДИАНА также может быть использована для моделирования функционирования бортовых вычислительных систем (БВС) морских навигационных комплексов. В рамках одного из проектов были рассмотрены фрагменты двух БВС, структурные схемы которых в упрощенном виде представлены на рис.2. В состав первого фрагмента входили приборы контроллеров МКИО (КМ1 и КМ2), вычислительные приборы комплексной обработки (ВЦ1 и ВЦ2) и приборы связи с потребителями (ПСЦ1 и ПСЦ2). Экземпляры одноименных приборов обменивались информацией по радиальным каналам (РКО). Состав и конфигурация второго фрагмента были иными. В нем отсутствовали приборы связи, но при этом появились три трансляционных прибора (ТЦА, ТА1 и ТА2).

Для всех приборов, кроме трансляционных, были разработаны распределенные (сложные) модели, воспроизводящие особенности многопоточной организации вычислительного процесса. Для приборов трансляции были разработаны упрощенные последовательные модели. Кроме моделей приборов в результирующие модели фрагментов вошли модели каналов обмена и модель движения объекта, которая обеспечивала наполнение передаваемых по каналам обмена массивов осмысленной навигационной информацией.

Работа с моделями фрагментов БВС проводилась в двух направлениях – автономного и полунатурного моделирования. В первом случае моделирование осуществлялось с использованием лишь разработанных имитационных моделей, во втором случае созданные модели применялись в составе стенда, куда, кроме них, входили также и реальные приборы (полунатурное моделирование).

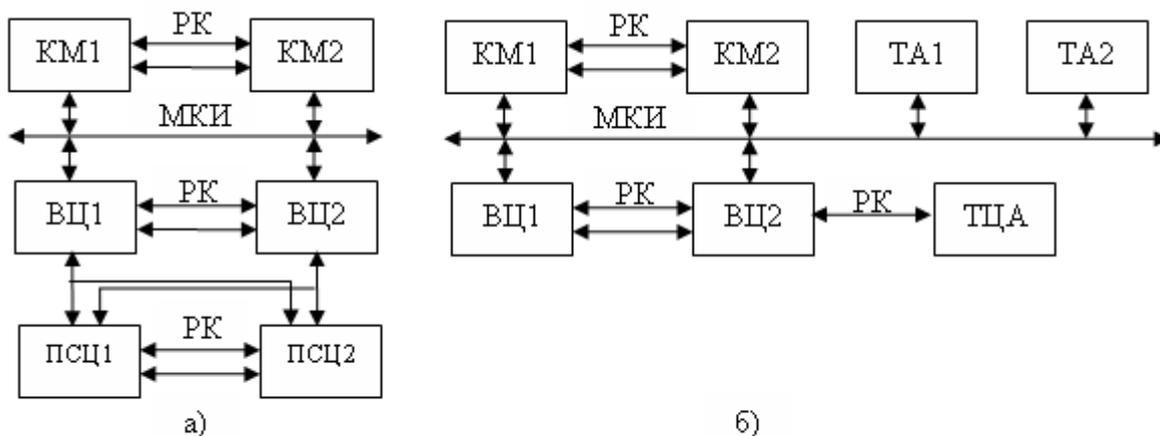


Рис. 2. Структура фрагментов БВС №1 (а) и БВС №2 (б)

При этом рассматривались разные варианты. Самым простым среди них был вариант, когда полунатурное моделирование отличалось от автономного лишь применением реального МКИО. Более сложными были ситуации, когда в стенде из реальных приборов в качестве контроллера МКИО выступала его модель или когда заменялись на модели трансляционные приборы.

Большое внимание на этапе апробации системы было уделено проблеме переносимости программного обеспечения. Эта проблема имеет общий характер, но в данном случае речь шла о возможности перенесения без изменений программы разработанной и отлаженной в системе моделирования ДИАНА в реальный прибор и наоборот. Ко-

нечно, эта проблема в большей степени касается системного программного обеспечения (СПО) и в гораздо меньшей степени функционального программного обеспечения.

В случае с СПО, создаваемого разработчиками БВС, проблема по понятным причинам не имеет исчерпывающего решения из-за его существенной аппаратной зависимости ввиду оперирования конкретными физическими адресами реальных адаптеров МКИО. Это касается многих диагностических программ. Однако в ряде важных случаев требование переносимости СПО может быть удовлетворено и, прежде всего, в случае СПО, реализующего в приборах БВС драйверы МКИО. Конечно, и в этом случае можно отказаться от требования переносимости этого СПО и воспроизвести в модели работу с МКИО с использованием библиотеки поддержки моделирования в системе ДИАНА. Такой путь оправдан при построении упрощенной модели окружения для автономной отладки некоторой другой достаточно сложной модели прибора или при исследовании временных соотношений на канале МКИО. По данной схеме были построены модели приборов ТЦА, ТА1, ТА2. Данные модели приборов работают как оконечные устройства МКИО, формирующие временную диаграмму работы канала обмена, заголовки, контрольные суммы и некоторую служебную информацию для реальных информационных массивов.

#### 4. Заключение

В данной работе описана архитектура, методика использования и пример апробации системы имитационного моделирования ДИАНА для решения комплекса задач, возникающих при анализе морских систем. Данную систему можно использовать как для анализа поведения морских объектов, так и для анализа функционирования БВС систем морского базирования. Анализ функционирования БВС успешно проводился не только для морских систем, но и для моделирования бортового оборудования летательного аппарата [6]. Для развития СМ ДИАНА в части задач отработки группового взаимодействия мы планируем решить следующие задачи:

- рассмотреть альтернативные модели описания поведения объектов;
- автоматизировать обработку результатов моделирования.

#### Литература

1. Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules// IEEE, 2010 – с. 26.
2. Гома Х. UML. Проектирование систем реального времени, распределенных и параллельных приложений. М.: ДМК – 2011. – 704 с.
3. Noulard E., Rousselot J.-Y., CERTI, an Open Source RTI, why and how// Spring Simulation Interoperability Workshop. San Diego, USA, 2009.
4. Пашков В. Н., Волканов Д. Ю. О подходах к трассировке распределенных вычислительных систем реального времени// Материалы 17-ой международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2011), 25–31 мая 2011 г., Алушта. – М.: Изд-во МАИ-ПРИНТ, 2011.
5. Bengtsson J., Larsen K. G., Larsson F., Pettersson P., Yi W. UPPAAL – a Tool Suite for Automatic Verification of Real-Time Systems// Lecture Notes in Computer Science. – 1996. – v. 1066. – p. 232–243.
6. Грибов Д. И., Смелянский Р. Л. Комплексное моделирование бортового оборудования летательного аппарата// Методы и средства обработки информации. Труды второй Всероссийской научной конференции. – М.: Издательский отдел факультета вычислительной математики и кибернетики МГУ им. М.В. Ломоносова, 2005. – С. 59–74.