
ПРОГРАММНАЯ АРХИТЕКТУРА СРЕДСТВ ВИРТУАЛЬНОГО ПРОТОТИПИРОВАНИЯ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ, ОСНОВАННАЯ НА АГЕНТНОЙ ТЕХНОЛОГИИ

А.С. Ковтуненко, С.С. Валеев, В.А. Масленников (Уфа)

Введение

Применение имитационного моделирования при проектировании и построении информационно-управляющих систем (ИУС) различного назначения позволяет повысить эффективность труда вовлеченных специалистов практически на всех этапах жизненного цикла. При создании больших распределенных программно-аппаратных систем обработки данных, критичных ко времени (ДКВ), таких как системы управления сложными организационно-техническими системами, MES-системы, распределенные компьютерные модели, часто возникает необходимость виртуального прототипирования системы или ее фрагментов. Создание качественного виртуального прототипа за счет оптимизации проектирования программных и аппаратных уровней системы, позволяет существенно сократить затраты на создание и сопровождение ИУС.

Состояние вопроса и актуальность

В настоящее время на рынке программного обеспечения наблюдается нехватка эффективных программных средств распределенного имитационного моделирования, которые позволяли бы быстро строить сквозные виртуальные прототипы ИУС, обеспечивали бы возможность модификации моделей без существенных затрат, поддерживали создание распределенных моделей, не требуя от оператора навыков системного программирования. Разработка подобных программных средств требует выработки концепции распределенного моделирования сложных ИУС, создания обобщенной архитектуры программно-аппаратного комплекса имитационного моделирования, а также разработки программной архитектуры. Таким образом, задача создания программных средств распределенного компьютерного моделирования сложных ИУС является актуальной научной задачей.

Проблемам создания программных систем универсального назначения, которые позволяли бы проектировать и строить распределенные имитационные модели посвящены исследования научных школ как в России, так и за рубежом. В частности, для решения задач распределенного компьютерного моделирования и управления разработан стандарт IEEE 1516–2010 (HLA – high-level architecture). Наиболее известными в этой области являются работы Р. Фуджимото и др. В России системам распределенного имитационного моделирования посвящены работы Е. Б. Замятиной, А. И. Каляева, А. И. Микова, В. В. Окольниковой, Р. Л. Смелянского [1, 2, 3].

В настоящей работе предлагается концепция виртуального прототипирования, в основе которой лежит агентно-ориентированный подход к построению информационных и вычислительных систем. Она заключается в том, что элементы системы считаются автономными и решающими конечный и предопределенный набор задач обработки данных. Потоки данных между элементами при этом рассматриваются независимо от функционала самих элементов. Также предложена обобщенная архитектура программно-аппаратного комплекса распределенного компьютерного моделирования, программная архитектура систем обработки ДКВ, а также программная платформа ABSynth, которая является реализацией предложенных концептуальных и архитектурных решений.

Обобщенная архитектура программно-аппаратного комплекса имитационного моделирования

Задача виртуального прототипирования сложных ИУС рассматривается в классе задач создания программных систем распределенной обработки данных. Анализ архитектурных особенностей построения распределенных программных комплексов обработки данных показал, что наиболее распространенные решения (клиент-серверные, сервис-ориентированные) для рассматриваемых задач являются малоэффективными, так как не обеспечивают достаточной надежности и универсальности программных средств.

Среди основных факторов надежности распределенных имитационных моделей наиболее важными являются живучесть и гибкость. Под живучестью подразумевается способность программной системы сохранять работоспособность в случае отказа ее компонентов. Под гибкостью программных систем понимается возможность наращивания функционала или реконфигурации системы в процессе ее эксплуатации, без остановки, перезапуска и дополнительных затрат. Универсальность программных средств подразумевает высокие показатели сопровождения, удобства применения и эффективности. Для улучшения всех этих показателей предлагается обобщенная архитектура программно-аппаратного комплекса имитационного моделирования (рисунок 1). В основе обобщенной архитектуры лежит идея метапрограммирования [4]. Предлагаемая архитектура характеризуется наличием следующих программных компонент:

платформа распределенного исполнения подзадач моделирования, которая обеспечивает необходимый уровень виртуализации вычислительных ресурсов сети для различных компонент вычислительной модели;

библиотека компонент, в которых реализованы типовые или уникальные подзадачи имитационного моделирования;

компонента создания и запуска имитационных моделей, которая обрабатывает спецификацию модели и на базе платформы из компонент библиотеки строит распределенное приложение, непосредственно осуществляющее моделирование.

Предлагаемая обобщенная архитектура позволяет оптимизировать взаимодействия между разработчиками имитационной модели. Согласно нее эксперт декомпозирует глобальную задачу моделирования на автономные подзадачи и формулирует алгоритмисту задание на разработку вычислительных процедур, соответствующих каждому фрагменту. Алгоритмист реализует их и помещает в библиотеку в виде программных компонент. Для создания и запуска имитационной модели эксперт создает спецификацию модели, в которой указывает топологию взаимодействия между фрагментами модели, создает и запускает модель на распределенной платформе.

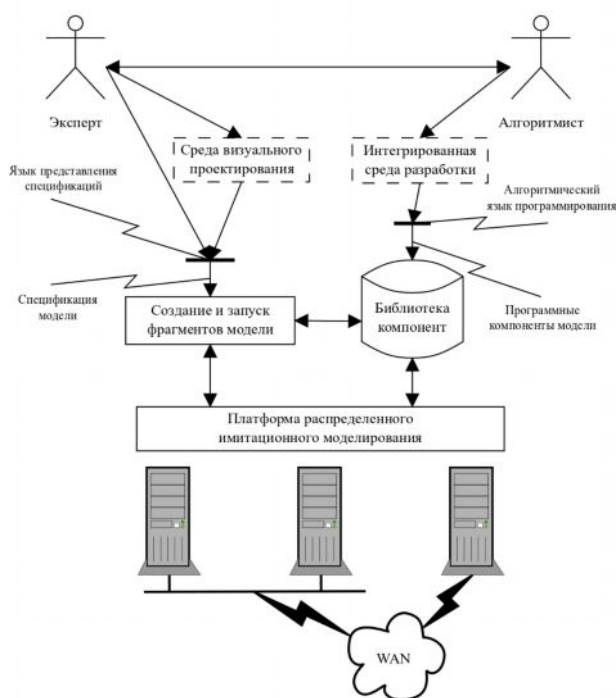


Рис. 1. Обобщенная архитектура программно-аппаратного комплекса распределенного имитационного моделирования

Программная архитектура распределенного имитационного моделирования

В основе программной архитектуры лежит математическая модель процесса имитационного моделирования, а также агентно-ориентированный подход. Пусть имитационная модель задана множеством атрибутов $V = \{v_1, v_2, \dots, v_n\}$ – переменных, значения которых в процессе моделирования необходимо поддерживать актуальными. Подзадачи моделирования задаются множеством операций $F = \{f_1, f_2, \dots, f_m\}$ по вычислению текущих значений совокупности исходящих атрибутов $V_D \subseteq V$ с использованием текущих значений совокупности входящих атрибутов $V_G \subseteq V$, при этом $V_D \cap V_G = \emptyset$. Каждая операция может быть представлена кортежем $f = \langle V_G^f, V_D^f \rangle$. Также задается отображение $T(f): F \rightarrow R$, которое каждой операции ставит в соответствие вещественное число, показывающее период ее повторения в секундах.

Задача имитационного моделирования может быть представлена ориентированным биграфом, множество вершин которого $N = V \cup F$ соответствуют атрибутам и операциям их преобразования, а множество дуг $C: V \rightarrow F$ – зависимостям между значениями атрибутов или компонентам потоков данных, передаваемым между подзадачами. На основании графового представления задачи была предложена агентно-ориентированная программная архитектура распределенного имитационного моделирования. В основе данной программной архитектуры лежат следующие положения.

Задача распределенной обработки ДКВ представляется совокупностью подзадач – операций, каждой из которых соответствует программный агент, называемый *целевым*. Целевой агент содержит следующие элементы (рис. 2):

вектор состояния – текущие значения совокупности производных атрибутов, значения которых агент должен поддерживать актуальными;

регистр входящих потоков – текущие значения совокупности производящих атрибутов;

главную процедуру – периодическую процедуру которая соответствует операции обработки данных;

периодическую процедуру поддержания актуальности входящих сигналов (обновления регистра входящих потоков).

Топология потоков данных между подзадачами согласно агентно-ориентированному подходу задается для каждого целевого агента указанием источников входящих потоков, а также периода обновления значений в регистре. Это позволяет обеспечить живучесть имитационной модели, поскольку сбой одного или нескольких целевых агентов не приведет к отказу.

Создание спецификации имитационной модели и задание параметров целевых агентов осуществляется путем метапрограммирования с использованием разработанного языка описания спецификаций (Task Specification Description Language – TSDL). Он основан на стандарте разметки XML. В TSDL-описании распределенной системы обработки данных для каждого целевого агента указываются входящие потоки, размерность состояния, указатель на процедуру обработки из библиотеки компонент, периодичность обновления регистра входящих сигналов и вектора состояния, а также указатель на узел вычислительной сети, на котором выполняются задачи данного целевого агента.

Управление распределенным моделированием – запуск, останов и уничтожение производится согласно трехуровневой иерархической схеме управления.

а. Верхний уровень управления представлен агентом – главным менеджером, который осуществляет взаимодействие с пользователем, создание целевых агентов по TSDL-спецификации имитационной модели и трансляцию управляющих команд на средний уровень.

б. Средний уровень представлен периферийным менеджером, или менеджером узла, который осуществляет ретрансляцию управляющих команд целевым агентам вычислительного узла.

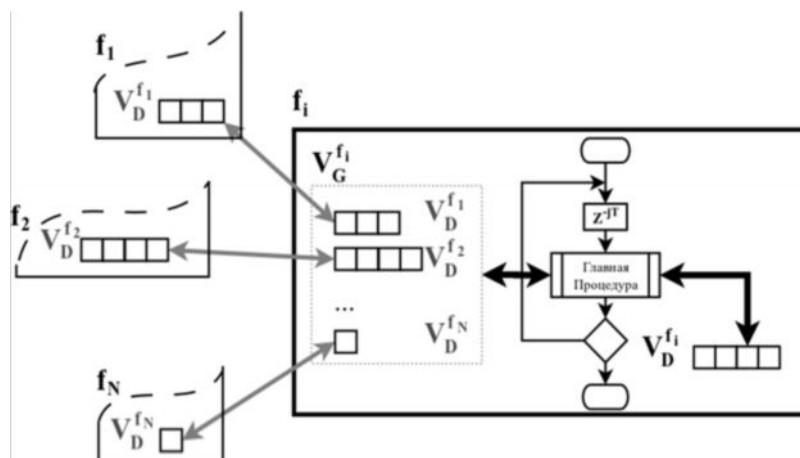


Рис. 2. Структура целевого агента

с. Нижний уровень представлен механизмами целевого агента, который при поступлении от среднего уровня управляющих команд запускает или приостанавливает выполнение своей главной процедуры, либо самоуничтожается.

Платформа распределенной обработки данных, критичных ко времени ABSynth

Разработанная программная архитектура реализована на языке Java с использованием агентной платформы и библиотеки JADE в виде программной платформы распределенной обработки данных ABSynth. Разработанная программная платформа включает в себя классы, реализующие механизмы и принципы предложенной программной архитектуры [5].

Для построения распределенных имитационных моделей на базе программной платформы ABSynth для библиотеки компонент были разработаны типовые звенья, традиционно используемые для имитационного моделирования технических систем:

источники модельных сигналов: текущее модельное время, константа, гармонический сигнал, генератор случайного сигнала, ступенчатый сигнал;

статические звенья: математические функции и векторно-матричные операции;

динамические звенья: линейное динамическое звено (может быть задано как коэффициентами передаточной функции, так и коэффициентами уравнений состояния);

блоки визуализации результатов: временные диаграммы, плоские графики.

В качестве примера использования предложенного подхода рассматривалась задача распределенного моделирования элементов газотурбинного двигателя (ГТД). На основе известной кусочно-линейной многорежимной динамической модели была разработана имитационная модель газогенератора ГТД. Была поставлена задача компьютерного моделирования газогенератора с разными параметрами поступающего воздуха на режимах от запуска до режима земного малого газа на базе небольшой корпоративной ЛВС, состоящей из персональных компьютеров (CPU Intel i3, RAM 2048 Mb), объединенных в сеть Ethernet 100 М. Узлы сети находились под управлением как ОС Windows XP, так и CentOS 6.3. На трех узлах вычислительной сети были запущены экземпляры имитационной модели газогенератора с разными параметрами, с четвертого (главного) производилось управление моделированием, а также сбор и отображение результатов. Результаты моделирования (скриншот рабочего стола четвертого узла) показаны на рис. 3.

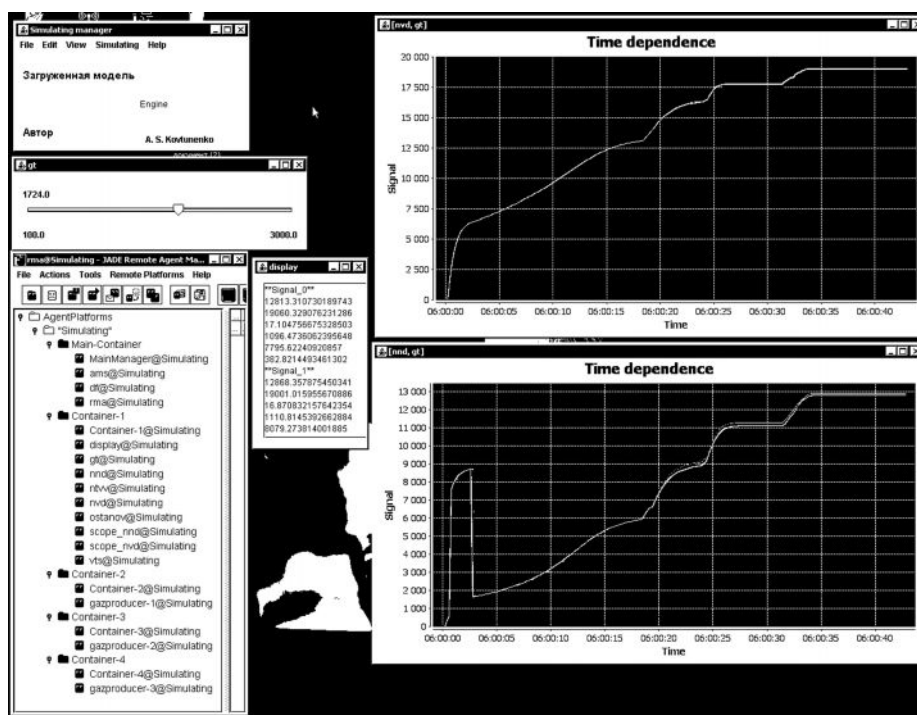


Рис. 3. Результаты моделирования, отображаемые на экране главного узла вычислительной сети

Результаты и выводы

Полученные в результате проведенных исследований и разработок результаты обладают существенной практической значимостью, поскольку позволяют организовать виртуальное прототипирование сложных ИУС, что, в свою очередь, позволяет снизить затраты на проектирование и построение программно-аппаратных средств за счет использования сквозных и гибких имитационных моделей. Научная новизна полученных результатов заключается в предложенной обобщенной архитектуре программно-аппаратного комплекса имитационного моделирования, которая позволяет создавать распределенные имитационные модели, используя метапрограммирование на разработанном языке TSDL. Благодаря основанному на XML синтаксису языка построение распределенных имитационных моделей может осуществлять эксперт, не имеющий навыков программирования и разработки распределенных программных систем.

Благодаря агентно-ориентированному подходу и технологии, использованной при создании программной платформы ABSynth, создаваемые на ее базе распределенные имитационные модели обладают живучестью и гибкостью, что позволяет обеспечить динамическое и распределенное проектирование сложных ИУС.

Литература

1. Fujimoto, Richard M. Distributed Simulation Systems. – Wiley, 2000.
2. Миков, А. И., Замятина, Е. Б., Козлов, А. А. Программные средства оптимизации распределенного имитационного эксперимента // Тр. Всерос. суперкомпьютер. конф. «Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность». – 2009. – С. 275–282.
3. Окольников, В.В. Разработка системы распределенного обработки данных для различных операционных сред // Сб. докл. Второй всероссийской науч.-практ. конф. по

имитационному моделированию и его применению в науке и промышленности «Имитационное моделирование. Теория и практика» (ИММОД 2005). – СПб., 2005. – Т. 1. – С. 256–260.

4. **Валеев, С.С., Ковтуненко, А.С., Масленников, В.А.** Проектирование промежуточного программного обеспечения на основе агентно-ориентированных технологий для САУ сложными техническими объектами // Параллельные вычисления и задачи управления: материалы VI международной конф. Т. 1. – М.: ИПУ РАН, 2012. – С. 250–259.

5. Свидетельство о государственной регистрации программы для ЭВМ № 2013616395. Программная платформа для распределенного имитационного моделирования (ABSynth-base) / А. С. Ковтуненко, С. С. Валеев, В. А. Масленников. – М.: Роспатент, 2013.