

РАЗРАБОТКА ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ РАСПРЕДЕЛЕННОГО МОДЕЛИРОВАНИЯ¹

Белотелов Н.В., Бродский Ю.И., Павловский Ю.Н.

Учреждение Российской академии наук Вычислительный центр им. А.А. Дородницына
РАН, Россия, 119333, г. Москва, ул. Вавилова 40, ВЦ РАН
Тел.: (499)135-62-04, факс (499)135-61-59
e-mail: belotel@mail.ru, brodsky@ccas.ru, jpvlsk1@rambler.ru

Работа обобщает многолетний опыт авторов в области моделирования сложных многокомпонентных организационно-технических систем. Делается попытка ввести формализованное описание определенного класса сложных систем. Ключевыми понятиями этой формализации являются понятия компонент, которые могут образовывать комплекс и комплекса, состоящего из компонент, но на некотором более высоком уровне абстракции могущем восприниматься как единая компонента. В результате сложную систему можно рассматривать, начиная с одной компоненты на самом высоком уровне абстракции, и кончая фракталом компонент, на уровне максимально подробного моделирования. Степень подробности ограничивается лишь желанием того, кто составляет модель. Предлагаются методы анализа и синтеза многокомпонентных моделей. Устанавливается связь объектно-событийного моделирования и моделирования средствами классических динамических систем.

DEVELOPING OF AN INSTRUMENTAL SYSTEM FOR DISTRIBUTED SIMULATION

Belotelov N.V., Brodsky Yu.I., Pavlovsky Yu.N.

The paper is devoted to elaboration of the concept of the instrumental software for distributed simulation of complex systems.

Введение.

Одним из направлений развития имитационного моделирования традиционно является моделирование сложных организационно-технических систем — совокупность технических средств и организации, управляющей этими средствами и реализующей некоторую иерархию целей (ценностей, предпочтений). В связи с вышесказанным возникает задача формализации понятия сложной системы. Например, Н. П. Бусленко [1] дает следующее определение: «Сложная система — составной объект, части которого можно рассматривать как системы, закономерно объединенные в единое целое в соответствии с определенными принципами или связанные между собой заданными отношениями.». Авторы согласны с таким определением, однако представляется уместной попытка большей степени формализации, как понятия сложной системы, так и процесса синтеза таких систем.

Что касается проблемы синтеза сложных систем, в той же работе [1] Н. П. Бусленко указывает, что части сложной системы синхронизируют свою деятельность

¹ Работа выполнена при поддержке РФФИ (грант 07-07-00071-а)

посредством обмена сигналами и сообщениями. С этим принципом синтеза сложных систем также трудно не согласиться, более того, нами в конце 80-х – начале 90-х годов была реализована инструментальная система имитационного моделирования MISS [2], основанная на подобных принципах, а позднее, во второй половине 90-х годов эти принципы легли в основу спецификации HLA/DEVS (High Level Architecture/Discrete Event System specification), разработанной в США.

В настоящей работе, кроме задачи синтеза сложной системы в некоторую метамодель ставится также задача распределения в вычислительной сети компьютерной имитации этой метамодели, а также в определенном смысле двойственная задача интеграции в сети частей моделей, возможно выполненных разными разработчиками, в единую метамодель. Поэтому важно выбрать концепцию синтеза этой метамодели так, чтобы она накладывала минимум ограничений на характер входящих в нее компонент, другими словами, чтобы можно было использовать в качестве таких компонент максимально широкий класс моделей.

Динамические системы

Динамические системы – основной язык естественных наук посленьютоновской эпохи. В основе моделей представленных динамическими системами, лежит гипотеза о замкнутости. Мы изучаем влияющие друг на друга характеристики, или как их еще называют, внутренние переменные X_1, X_2, \dots , некоторого явления. Наконец, на некотором X_n нам начинает казаться, что выбранных нами характеристик достаточно чтобы описать данное явление полностью, или же с интересующей нас точностью, или хотя бы то, что интересует нас в этом явлении (например, Дж. Форрестеру [3] хватило пяти внутренних переменных для описания в некотором приближении всей мировой динамики). Утверждение, что характеристик X_1, X_2, \dots, X_n достаточно, чтобы полностью описать интересующее нас явление, и есть гипотеза о замкнутости нашей модели. Из нее непосредственно следует, что если теперь нас интересует изменение выбранных нами внутренних переменных во времени, - то ему не от чего зависеть, кроме как от самих этих переменных, и мы получаем обыкновенные дифференциальные уравнения:

$$\frac{dX_i}{dt} = F(X_1, \dots, X_n), i = 1, \dots, n.$$

Гипотеза о замкнутости лишь говорит, что правые части нашей системы зависят от тех же самых переменных (в силу этой гипотезы, больше зависеть им просто не от чего!), но не может помочь нам найти вид этих функциональных зависимостей. Определить вид правых частей – одна из задач построения модели. При определении правых частей обычно оказывается, что хотя мы и принимаем гипотезу о замкнутости нашей модели, полностью независимой от внешнего по отношению к ней мира она не может быть, а указанная зависимость ее от внешнего мира проявляется во внешних переменных модели или иначе – ее параметрах. Внешние переменные – это то, что в данной модели мы не моделируем явно (потому что не хотим, или быть может, не умеем), а принимаем как заданные извне параметры, постоянные или же меняющиеся во времени (в том числе это могут быть и внешние управления). Внешние переменные

– это вся информация о внешнем по отношению к ней мире, известная нашей замкнутой модели.

Отношение между внутренними и внешними переменными модели устанавливает гипотеза об инвариантности. Она утверждает, что внутренние переменные модели зависят от внешних, но не наоборот. Принимая эту гипотезу, мы еще раз четко разграничиваем, что мы хотим явно моделировать, а что принимаем, как заданные извне параметры.

Приняв гипотезу об инвариантности, мы на каждом i -м шаге моделирования зная состояние системы (т. е. ее внутренние переменные) \vec{X}_i а также внешние переменные \vec{a}_i и интервал моделирования Δt , можем вычислить следующее $i+1$ состояние нашей модели: $\vec{X}_{i+1} = \vec{F}(\vec{X}_i, \vec{a}_i, \Delta t)$. Вычислив очередное состояние модели, и узнав из внешнего по отношению к нашей модели мира \vec{a}_{i+1} , мы можем продолжить процесс моделирования далее.

Поскольку моделирование средствами динамических систем – один из наиболее распространенных способов моделирования, непременным требованием к разрабатываемой инструментальной системе будет способность интегрировать модели, описанные динамическими системами.

Объектный анализ.

Наряду с динамическими моделями, где в каждый момент времени, вообще говоря, все внутренние переменные зависят от всех, еще с античных времен существовал атомистический подход, где реальность описывалась как синтез действий отдельных относительно независимых друг от друга объектов – атомов, лишь время от времени, в моменты событий, взаимодействующих между собой. Через всю историю науки проходят эти два способа описания явлений – как взаимосвязь некоторых глобальных характеристик, и как совокупность отдельных агентов, взаимодействующих между собой. Очередная волна интереса к объектному представлению явлений возникла в 80-х годах XX века, как отклик на потребности с одной стороны моделирования сложных систем, а с другой – автоматизации проектирования таких систем. Появились объектно-ориентированные языки программирования, в которых класс объектов определяется как:

- набор характеристик (данные, связанные с объектами этого класса);
- методы (умения, функциональности объектов этого класса);
- события (некоторые сочетания характеристик, на которые объекты этого класса должны реагировать определенным образом, поскольку они «так устроены»).

Кроме того, в этих языках реализованы такие идеи как наследование объектами свойств и полиморфизм, облегчающие задачу построения новых объектов на основе готовых конструкций, а также инкапсуляция, открывающая пользователю функциональность объекта и скрывающая детали реализации этой функциональности.

Отметим, что объект в указанном выше смысле есть не только конструкция из области информатики, но вполне может рассматриваться как классическая математическая конструкция. Действительно, описание класса объектов, так, как это

делается в объектном анализе, является практически полным аналогом описания класса математических объектов, которое в бурбаковском формализме называется «род структуры» [4], [5].

Замечание. Когда Н. Бурбаки разрабатывали свои формальные системы, претендующие на формализацию математических рассуждений, компьютеры еще не были изобретены, формальных грамматик, языков программирования, средств объектного анализа не существовало. Здесь имеет место иногда встречающаяся ситуация, когда в разных областях деятельности одна и та же «сущность» имеет разные названия и разные формы записи, а специалисты в соответствующих областях не имеют об этом никакого понятия.

Как и в случае динамических систем, заметим, что поскольку объектное описание – один из наиболее распространенных способов моделирования, естественным требованием к нашей инструментальной системе распределенного моделирования будет способность интегрировать «чужие» объекты.

Связь между динамическими и объектными моделями.

В системе дифференциальных уравнений в каждый момент времени каждая переменная зависит от каждой. Покажем, что при определенных условиях, с любой степенью точности такую систему можно приблизить совокупностью объектов, взаимодействующих друг с другом лишь в конечное число моментов времени – событий, а в промежутках между событиями независимых друг от друга.

Рассмотрим задачу Коши для динамической системы

$$\dot{\vec{x}} = f(\vec{x}, t), \quad \vec{x}(0) = \vec{x}_0, \quad t \in [0, T] \quad (1)$$

Пусть функция f в правой части достаточно «хорошая», например, удовлетворяет условию Липшица по совокупности своих переменных. Будем называть системой событий $\{t_n\}$ разбиение отрезка $[0, T]$ точками $0 = t_0 < t_1 < \dots < t_n = T$. Будем называть разбиением $\{A\}$ вектора \vec{x} некоторое произвольное разбиение его компонент на два подвектора \vec{x}^1 и \vec{x}^2 . Через $\vec{x}_0(t)$ будем обозначать решение задачи Коши (1). Через $\vec{x}_{\{A\}, \{t_n\}}(t)$ будем обозначать склеенное решение следующих задач Коши на отрезках $[t_i, t_{i+1}]$, $i = 0, \dots, n-1$:

$$\begin{aligned} \dot{\vec{x}}^1 &= f^1(\vec{x}^1, \vec{x}^2 \equiv \vec{x}^2(t_i), t), \quad \vec{x}^1(t_i); \\ \dot{\vec{x}}^2 &= f^2(\vec{x}^1 \equiv \vec{x}^1(t_i), \vec{x}^2, t), \quad \vec{x}^2(t_i). \end{aligned} \quad (2)$$

Очевидно, $\vec{x}_{\{A\}, \{t_n\}}(t)$ - непрерывная функция, так как ее производная кусочно-непрерывна. Таким образом, мы разбили исходную систему (1), где все переменные всегда зависят от всех, на два объекта \vec{x}^1 и \vec{x}^2 , которые взаимодействуют между собой лишь в конечное число моментов времени – событий. В промежутках же между событиями эти объекты независимы друг от друга. Приведем доказанное в [6] утверждение о приближенной декомпозиции системы (1) на два подобъекта:

Пусть функция f в правой части (1) удовлетворяет условию Липшица по совокупности своих переменных. Тогда для любого разбиения $\{A\}$ компонент вектора \vec{x} на два подвектора \vec{x}^1 и \vec{x}^2 , и для любого числа $\varepsilon > 0$, найдется конечная система

событий $\{t_n\}$, такая что если $\bar{x}_0(t)$ - решение исходной задачи Коши (1), а $\bar{x}_{\{A\},\{t_n\}}(t)$ - решение задач Коши (2), то для этих решений будет справедлива оценка:

$$\max_{t \in [0, T]} |\bar{x}_0(t) - \bar{x}_{\{A\},\{t_n\}}(t)| \leq \varepsilon.$$

Приведенное утверждение иллюстрирует переход от динамических систем к объектному анализу, показывает, откуда возникают события, и оговаривает условия, при которых удастся обойтись конечным числом событий для достижения любой точности приближения динамической системы объектами. Заметим, что потребованное от правой части условие Липшица, по-видимому, является сильным достаточным условием, – оно позволило нам разбить исходную динамическую систему на два объекта произвольно. В жизни же, и в ее моделировании, объекты потому и объекты, что их связи внутри себя гораздо сильнее и их гораздо больше, чем связей с внешним миром. Этот факт и позволяет выделять их в составе сложной модели, как нечто относительно самостоятельное.

Компонента - основная составляющая сложной модели

Перейдем теперь к описанию основной конструкции предлагаемой в данной работе концепции моделирования сложных систем. Мы назвали ее компонентой. С одной стороны компонента – это в некотором смысле «элементарная» сложная система. Здесь слово «элементарная» означает не простоту ее внутреннего устройства, а то, что с другой стороны она может быть кирпичиком для построения еще более сложных систем. Компоненты являются предметами импорта-экспорта при построении распределенных моделей сложных систем. Основой конструкции компоненты будет объект объектного анализа (как говорилось выше, аналог рода структуры в бурбаковском формализме), но понятия связанные с динамическими системами мы также будем использовать.

Одна из самых важных и нетривиальных задач, связанных со сложными системами – это их синтез из более простых компонент. Обычно бывает достаточно ясно, какие функциональности содержит сложная система и что эти функциональности «умеют делать» по отдельности. Гораздо сложнее заставить их работать вместе так, чтобы поведение системы правильно реагировало на изменения состояния ее самой и внешнего мира. Описанию концепции такого синтеза также будет посвящен данный раздел.

Предлагаемая в данной работе концепция синтеза сложных систем разработана применительно к компоненте, однако, как выяснится в дальнейшем, она является универсальной, т. е. применимой и к более сложным системам, построенным из множества компонент.

Опишем теперь «устройство» компоненты. Устройство компоненты похоже на объект классического объектного анализа. Также есть характеристики, методы и события, только эти понятия нуждаются в некоторой структуризации и детализации.

Характеристики.

Компонента, как и объект, имеет характеристики. Как и в случае динамических систем, эти характеристики мы будем разбивать на внутренние и внешние. Внутренние характеристики – это то, что компонента моделирует, внешние – это то, что она знает о внешнем мире. Предполагается справедливой гипотеза замкнутости – знаний о внешнем мире (значений внешних переменных) и собственного состояния (значений внутренних переменных) достаточно для вычисления внутренних переменных в следующий момент времени.

Методы.

Компонента, как и объект, имеет методы. Излагаемая концепция компоненты различает два типа методов: один тип реализует функциональности, т. е. то, что компонента умеет делать, эти методы будем называть элементами. Другой тип методов прогнозирует наступление событий, об этих методах поговорим в следующем пункте. Методы должны быть написаны разработчиком модели на одном из разрешенных языков программирования. Методы являются самой мелкой единицей экспорта-импорта в инструментальной системе распределенного моделирования.

Функциональность компоненты удобно структурировать следующим образом: Считается, что компонента реализует один или несколько параллельно выполняющихся процессов. Процесс состоит в чередовании элементов – алгоритмически элементарных методов. По отношению к модельному времени некоторые элементы выполняются мгновенно, это сосредоточенные или быстрые элементы. Выполнение других элементов занимает определенное время. Если при этом элемент для любого промежутка времени Δt выдает некий результат, такой элемент называется распределенным или медленным. Может оказаться и так, что выполнение элемента занимает определенное модельное время, но результат его действия наступает лишь в конце, после полного выполнения элемента, т. е. никаких промежуточных результатов за время меньше полного времени выполнения нет. Такие элементы называются условно-распределенными. Вообще говоря, условно-распределенные элементы можно не рассматривать как отдельный класс, а моделировать парой: пустой распределенный элемент, за которым идет сосредоточенный, выдающий результат. Условно-распределенные элементы, например, возникают при моделировании информационных процессов (имитация выполнения вычислительных процессов в системе управления организационно – технической системы).

Порядок чередования элементов в процессе определяется наступлением тех или иных событий.

События.

Содержательно, события – это то, что нельзя пропустить при моделировании динамики системы – точки синхронизации различных ее функциональностей. Например, поезд доехал до станции – должна быть остановка, вне зависимости от того, что стандартный шаг моделирования по времени еще не закончился.

Формально событие – функция внутренних и внешних переменных (следствие гипотезы о замкнутости компоненты). Что делает компоненту по-настоящему сложной

системой – это необходимость рефлексировать – прогнозировать наступление событий. Если в предметной области, которой касается модель, с последним высказыванием, возможно и поспорить, то в области моделирования, без прогнозирования событий не обойтись. Поэтому в конструкции компоненты есть специальные методы, прогнозирующие то или иное событие. Эти методы, используя внутренние и внешние характеристики компоненты, вычисляют время наступления соответствующего события. Если это время равно нулю – значит, событие уже наступило. Например, событием может являться окончание выполнявшегося элемента.

Автоматные функции перехода.

Для каждого текущего (выполняющегося в текущий момент времени) элемента каждого процесса заранее задаются реакции на различные сочетания произошедших в компоненте событий. Эти реакции могут состоять либо в переходе к выполнению некоторого другого элемента процесса или же в продолжении выполнения того же самого элемента. Можно считать, что с каждым процессом связан конечный автомат, состояниями которого являются элементы процесса, а входным алфавитом некая часть множества всех подмножеств множества событий.

Моделирование с переменным шагом времени.

Опишем теперь, как компонента осуществляет синтез функционирования сложной системы. Модель сложной системы должна быть готова к моделированию с переменным шагом времени. Иначе поезд не доедет до станции или проедет ее, ракета пролетит мимо цели и т. д. Считается, что для компоненты задан некий шаг моделирования Δt по умолчанию.

1. В момент t в начале каждого шага моделирования вызываются все методы прогноза событий для всех процессов компоненты.
2. Если есть наступившие события, в соответствии с автоматными функциями перехода выбираются новые текущие элементы.
3. а) Если среди текущих элементов есть мгновенные (не занимающие модельного времени) – они выполняются. В этом случае считается, что выполнен шаг моделирования нулевой длины. Повторяется вызов прогнозов наступления событий (т. е. возвращаемся к пункту 1).
4. б) Если мгновенных элементов нет, повторяется вызов прогнозов событий, (возвращаемся к пункту 1).
5. Если наступивших событий нет – выбирается минимальное значения прогноза события t^* . Если $t^* \geq \Delta t$ - моделирование идет со стандартным шагом Δt , если же нет, в качестве шага моделирования берется $t^* - t$. С выбранным шагом времени вызываются все распределенные элементы. Этим заканчивается очередной шаг моделирования.

Может возникнуть вопрос, а не окажется ли у множества событий точек накопления, при предлагаемой системе выбора шага моделирования? Отчасти на него отвечает утверждение предыдущего раздела, где рассматривалась связь динамических и объектных моделей. Например, если элементы компоненты описаны

дифференциальными уравнениями и все правые части этих дифференциальных уравнений удовлетворяют условию Липшица – точек накопления не будет.

Комплексы компонент.

Будем обозначать действие компоненты формулой

$$\vec{x}_{i+1} = \vec{F}(\vec{x}_i, \vec{a}_i, \Delta t), \quad (1)$$

Это означает, что, зная внутренние переменные \vec{x}_i , внешние переменные \vec{a}_i предыдущего шага и задав шаг моделирования по умолчанию Δt , мы получаем внутренние переменные следующего шага моделирования \vec{x}_{i+1} , при этом про интервал модельного времени τ между шагами i и $i+1$ можно лишь сказать, что он лежит в пределах $0 \leq \tau \leq \Delta t$.

Как говорилось выше, мы предполагаем замкнутость модели (1), т.е., некоторую ее изолированность от внешнего мира, состоящую в том, что, зная векторы \vec{x}_0 и \vec{a}_i , $i = 0, 1, \dots, n$, мы можем рассчитать траекторию системы на любом шаге модельного времени. На самом же деле, все в этом мире связано, в том числе характеристики \vec{x}_i и \vec{a}_i . Вопрос лишь в том, на каких характерных временах эти связи столь существенны, что их нужно учитывать, имея в виду необходимую с практической точки зрения точность, с которой дается прогноз значений внутренних характеристик. Предположение о замкнутости модели (1) означает, что на временном интервале $(t_0, t_0 + \Delta t \cdot n)$ характеристики \vec{a}_i с точностью, определяемой практическими соображениями, не зависят от \vec{x}_i . Увеличение точности и глубины прогнозов, доставляемых существующими и эксплуатируемыми имитационными моделями, может привести к необходимости учитывать эту зависимость. Наиболее общей является ситуация когда \vec{a}_i и \vec{x}_i взаимосвязаны. Это означает, что характеристики \vec{a}_i также определяются динамической системой, в которой внешними характеристиками являются \vec{x}_i :

$$\vec{a}_{i+1} = \vec{\Psi}(\vec{a}_i, \vec{x}_i, \Delta t), \quad (2)$$

Системы (1) и (2) можно, конечно, рассматривать как единую систему. Однако, во-первых, компьютерная реализация такой системы нерациональна, поскольку из предыдущего текста вытекает, что рассматривается ситуация, когда взаимное влияние \vec{a}_i и \vec{x}_i нужно учитывать не на каждом шагу Δt , а реже, в соответствии с соотношением характерных времен, определяющих изменение значений характеристик рассматриваемых процессов. Во-вторых, и это самое главное, может случиться, что над моделью $\vec{a}_{i+1} = \vec{\Psi}(\vec{a}_i, \vec{x}_i, \Delta t)$ уже давно и успешно работает некий другой исследователь, для которого характеристики \vec{a}_i являются внутренними, а \vec{x}_i - внешними и которому для повышения глубины и точности прогноза также стало необходимо учитывать взаимное влияние \vec{a}_i и \vec{x}_i . Возникает задача объединения уже существующих и эксплуатируемых моделей (1) и (2) с наименьшими затратами как человеческих, так и машинных ресурсов. Ситуация, когда взаимосвязанными из-за необходимости повышать глубину и точность прогноза оказываются только два процесса, является

простейшей. В следующем разделе рассматривается ситуация, когда необходимо учитывать взаимное влияние друг на друга не двух процессов, а их произвольное количество.

Пусть имеется N процессов, описываемыми динамическими системами $\vec{x}_{i+1}^k = \vec{F}^k(\vec{x}_i^k, \vec{a}_i^k, \Delta t)$, $k = 1, \dots, N$. Эти системы будут трактоваться как подмодели единого процесса, называться компонентами и обозначаться A_k , $k = 1, \dots, N$. Рассматривается ситуация, когда при необходимой точности и глубине прогноза компоненты уже нельзя рассматривать как независимые друг от друга. В то же время для достижения необходимой точности достаточно учесть факт зависимости некоторых из внешних характеристик некоторых компонент от некоторых внутренних характеристик некоторых других компонент. При этом мы располагаем информацией, необходимой для такого учета.

Для каждой из $N(N-1)$ пар $(A_i, A_j), i \neq j$, можно построить матрицу коммутации $Q_{i,j}$ размера $n_i \times m_j$, где n_i - размерность вектора \vec{a}^i компоненты A_i , а m_j - размерность вектора \vec{x}^j компоненты A_j . На пересечении p -й строки ($1 \leq p \leq n_i$) и q -го столбца ($1 \leq q \leq m_j$) этой матрицы стоит 1, если внешняя переменная a_p^i компоненты A_i зависит от внутренней переменной x_q^j компоненты A_j ; и 0 - в противном случае. Задание матриц $Q_{i,j}$, $1 \leq i, j \leq N; i \neq j$, которые будут называться матрицами коммутации, полностью решает вопрос информационного синтеза комплекса из компонент. Здесь следует заметить, что при фиксированном i , и при $1 \leq j \leq N; i \neq j$, наличие единицы в фиксированной строке более чем в одной из матриц $Q_{i,j}$, также как и наличие в одной строке более одной единицы, свидетельствовало бы о моделировании одной величины несколькими способами, и, следовательно, ставило бы вопрос о согласованности таких моделей. Стало быть, подобная ситуация должна вызывать если не сообщение о фатальной ошибке при коммутации компонент комплекса, то по крайней мере, очень серьезное предупреждение с призывом тщательно проверить согласованность компонент. С другой стороны, наличие нескольких единиц в столбце матрицы $Q_{i,j}$, а также, при фиксированном j , и при $1 \leq i \leq N; i \neq j$, наличие единицы в фиксированном столбце более чем одной из матриц $Q_{i,j}$, говорит о том, что внутренняя переменная одной из компонент используется в качестве внешней переменной более чем одной компоненты, что не противоречит нашей концепции, и должно быть разрешено. Следует также заметить, что поскольку, как следует из сказанного выше, единица - достаточно «редкий гость» в матрицах коммутации, то при описании комплекса на формальном языке, будет проще специальным оператором языка описывать только существующие коммутации, вместо выписывания всех матриц коммутации, которые должны автоматически строиться и проверяться на этапе компиляции описания комплекса.

В заключение данного раздела следует отметить, что комплекс, изнутри состоящий из многих компонент, вовне может проявляться в качестве единой компоненты. Введем следующую операцию объединения компонент комплекса.

1. Внутренними переменными комплекса объявляется объединение внутренних переменных всех его компонент.
2. Методами комплекса объявляется объединение всех методов его компонент.
3. Событиями комплекса объявляется объединение всех событий его компонент.
4. Внешними переменными комплекса объявляется объединение всех внешних переменных его компонент, из которого исключаются все те переменные a_p^i , для которых P -я строка одной из матриц коммутации $Q_{i,j}$, $1 \leq i, j \leq N; i \neq j$ содержит единицу.

Мы видим, что тогда как изнутри комплекс имеет сложный многокомпонентный состав, извне он, в соответствии с данным в начале работы определением, вполне может быть воспринят как единая компонента. Так или иначе, определяющим остается то, что задавая для комплекса, рассматриваемого как единая компонента, начальные данные и значения внешних переменных, можно однозначно определить его внутренние переменные на любом шаге моделирования.

Таким образом, любая модель может быть представлена с одной стороны агрегировано, как единая компонента, а с другой стороны подробно, причем с произвольной степенью подробности, как комплекс компонент, в котором каждая из компонент тоже в свою очередь допускает представление в виде комплекса, тем самым, позволяя реализовывать концепцию мультимоделирования, т.е., построения семейства моделей разной степени подробности, для изучения одного и того же явления.

Язык описания комплексов и компонент (ЯОКК).

Для формального описания компонент и комплексов разработан специальный непроцедурный язык ЯОКК. На его основе создается программное обеспечение рабочей станции инструментальной системы распределенного моделирования. Подробно язык ЯОКК описан в работах [7], [8].

Архитектура инструментальной системы распределенного имитационного моделирования

Система распределенного имитационного моделирования представляется пиринговой [см., например, <http://www.ip2p.ru/articles/>] сетью средней степени централизации.

Несколько десятков серверов глобальной сети имитационного моделирования должны выполнять следующие функции обслуживания рабочих станций (количество рабочих станций зависит от количества компьютеров, на которых содержатся модели компонент, предназначенных для генерации распределенного моделирующего комплекса; при достаточном развитии предлагаемой технологии распределенного имитационного моделирования их может быть очень много - до нескольких тысяч):

- Регистрация клиентских рабочих станций и предоставляемых ими в сеть разделяемых ресурсов, т.е., моделей, компонент моделей, данных, алгоритмов и т.д.
- Хранение и постоянное поддержание баз данных модельных ресурсов в сети, т.е. какими рабочими станциями предоставлены в сеть те или иные модели,

компоненты моделей и прочие ресурсы. Какие из рабочих станций реально присутствует в сети, кто из них и когда был доступен в последний раз, возможно, расписаний присутствия рабочих станций в сети.

- Поиск ресурсов в сети, т.е. уже существующих моделей, компонент моделей, необходимых данных и алгоритмов (примерно так, как осуществляется поиск различных файлов в современных пиринговых файлообменных сетях).
- Осуществление функций посредников стандартных запросов рабочих станций.
- Осуществление некоторых организационных функций, например, возможности нескольким рабочим станциям договориться о проведении совместной сессии имитационных экспериментов определенной продолжительности, с началом в определенное время, или же спланировать время проведения такой сессии в соответствии с их расписаниями присутствия в сети.

Клиентское программное обеспечение рабочих станций такой сети вполне могло бы базироваться на описанных выше принципах анализа и синтеза модели. А именно, основу программного обеспечения рабочей станции составляет «система программирования» на упомянутом выше языке ЯОКК. Эта система программирования должна включать редактор (в том числе желательно и с графическим интерфейсом) описания моделей и их компонент. При описании компонент модели той или иной конструкции, должна быть возможность ссылки на найденные уже существующие в сети компоненты, подходящие на роль компонент описываемой конструкции. Далее, система программирования клиентской части должна включать средства отладки описаний отдельных компонент и всей модели в целом. Далее, в клиентской части должен присутствовать сборщик модели, создающий базу характеристик компонент модели, проверяя при этом ее целостность.

Блок клиентской части инструментальной системы, ответственный за организацию имитационных экспериментов, должен с помощью существующих интеграционных технологий вызывать как локальные, так и удаленные компоненты модели по их известным адресам, передавая им необходимые характеристики из базы данных модели и принимая от них обратно в базу обработанные характеристики. Служебные методы, предоставляемые системой, также должны быть рассчитаны как на локальное, так и на удаленное использование. На этапе имитационного эксперимента рабочая станция, проводящая эксперимент, работает напрямую, минуя сервера, с рабочими станциями, содержащими используемые в модели компоненты. Как и в современных пиринговых сетях, серверы здесь нужны лишь на этапе поиска партнеров.

Уже отлаженные, работающие модели и их компоненты могут быть объявлены доступными для всеобщего использования, и пополнить фонд доступных в сети моделей и их компонент. При этом автоматически становятся доступными также и описания модели и ее компонент на языке описаний, по которым можно судить о ее адекватности тем или иным способам применения в других моделях. Таким образом, каждая рабочая станция может предоставлять в общее пользование, как свой вычислительный потенциал, так и существующие наработки в области моделирования.

Заключение

По мнению авторов, предложенная концепция могла бы применяться и более широко: не только как средство объединения в единый моделирующий комплекс

различных имитационных моделей, описывающих слабо связанные части единого процесса, но и как универсальное средство совместного использования в сети разнородных и разноплатформенных информационных и алгоритмических ресурсов, что могло бы существенно расширить круг ее потенциального применения. Тем самым у инструментальных систем имитационного моделирования появился бы шанс превратиться из экзотических программных продуктов предназначенных для узкого круга специалистов в области имитации, в массовое и универсальное средство объединения ресурсов Интернета для совместного выполнения широкого круга задач.

В настоящее время выполняется разработка макета инструментальной системы, концепция которой описана в данной статье. Планируется, что макет будет состоять из системы программирования, базирующийся на языке описания комплексов и компонент. Эта система программирования должна включать редактор с графическим интерфейсом описания моделей и их компонент, компилятор, генерирующий, в частности базу данных модели, средства отладки. Испытания проектных решений, касающихся разработки инструментальной системы, выполняются с помощью модели, имитирующей экономические, демографические, экологические процессы в системе из нескольких стран. Эта модель описана в [8], а ее компьютерная реализация доступна по адресу <http://simul.ccas.ru/>

Литература

1. Бусленко Н. П. Сложная система //Статья в Большой Советской Энциклопедии, 3-е изд., М.: Советская энциклопедия, 1969-1978.
2. Бродский Ю.И., Лебедев В.Ю. Инструментальная система имитации MISS. М.: ВЦ АН СССР, 1991, 180с.
3. Форрестер Дж. Мировая динамика. М.: Физматгиз. 1978. 168 с.
4. Бурбаки Н. Теория множеств. М.: Мир. 1965. 456 с.
5. Павловский Ю.Н., Смирнова Т.Г. Введение в геометрическую теорию декомпозиции. М.: Фазис, 2006, 169 с.
6. Павловский Ю.Н., Белотелов Н.В., Бродский Ю.И. Имитационное моделирование. М.: Издательский центр "Академия", 2008, 236 с.
7. Бродский Ю.И. Описание, компоновка и работа модели в инструментальной системе распределенного моделирования //Моделирование, декомпозиция и оптимизация сложных динамических процессов, М.: ВЦ РАН, 2008, С. 24-46.
8. Белотелов Н.В., Бродский Ю.И., Павловский Ю.Н. Компьютерное моделирование демографических, миграционных, эколого-экономических процессов средствами распределенных вычислений. М.: ВЦ РАН, 2008, 123 с.