

## WEB-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ КАК ЧАСТЬ РАСПРЕДЕЛЕННОЙ МОДЕЛИРУЮЩЕЙ СРЕДЫ

**Потапенко В.А.**

кафедра ЭВМ ДонНТУ  
eline@cs.dgtu.donetsk.ua

### **Abstract**

*Potapenko V. Web-oriented simulation as a part of distributed simulation system. The simulation systems present our days are mostly language or platform dependent. They didn't develop to organize to the complex robust distributed system. This article is about the new methods for design framework for the distributed simulation.*

### **Введение**

Существующие на сегодняшний день системы компьютерного моделирования зачастую либо платфо-рмо-, либо языково-зависимые. Обычно они разработаны с использованием одного конкретного языка программирования под определенную операционную систему (ОС). В большинстве случаев результат их работы является удовлетворительным, но концепции, заложенные при проектировании этих систем, не соответствуют современным информационным требованиям, а именно:

- Отсутствует возможность переноса (портирования) на другой язык программирования. Зачастую компоненты системы моделирования невозможно использовать во взаимодействии с другими языками программирования.
- Системы моделирования и их компоненты реализованы на различных языках для различных платформ, что затрудняет их совместное использование для создания более мощных и сложных моделей.
- Отсутствует возможность работы в локальных сетях и Интернет. В связи со стремительным ростом числа пользователей Интернет способность функционировать в глобальной сети будет приобретать весомое значение.
- Динамическое представление объектов. Даже те системы, которые способны функционировать в сети зачастую не имеют средств представления своих компонент для использования сторонними моделирующими средами. Это также сужает область применения уже готовых разработанных компонент.

В данной статье рассматривается концепция построения Web-ориентированной распределенной моделирующей среды на базе современных технологий распределенных вычислений.

### **Задачи Web-ориентированного моделирования**

Мировая паутина (World Wide Web, WWW) и Интернет несет потенциально огромные возможности в создании распределенной компьютерной инфраструктуры, которые могут быть использованы, в частности, для организации сервисов распределенного моделирования [1]. Широкомасштабное внедрение таких сервисов позволит перейти на совершенно новый качественный уровень моделирования. На сегодняшний день видятся следующие наиболее актуальные сферы применения Web-ориентированного моделирования:

- Моделирование как часть гипермедиа (hypermedia). Возможность моделировать в среде web-браузера несет новые потенциальные возможности в процесс традиционного обучения и тренинга. Объединение традиционного гипермедиа и моделирования позволит создавать современные, функционально-законченные, мощные и удобные web-порталы.
- Возможность быстрого распространения моделей, результатов и публикаций в Интернет открывает новые подходы в процесс научных исследований, в частности в области моделирования. Количество документов в электронном виде доступных через Интернет растет с каждым днем. Интернет становится средой мирового интеллектуального хранилища.
- Web-ориентированный доступ к программам моделирования. Включает в себя удаленное исполнение существующих (legacy) систем моделирования в окне браузера посредством гипертекста (HTML) и CGI-скриптов. Также существуют варианты с использованием апплетов Java.
- Распределенное моделирование. К данной области относятся системы, использующие WWW и Web-ориентированные технологии (CORBA, Java, RMI) в качестве инфраструктуры для организации распределенного моделирования [2-4].
- Моделирование WWW. Моделирование и исследование WWW с целью оценки производительности и возможностей оптимизации.

Web-ориентированные средства моделирования должны быть портируемыми (переносимыми) на множество компьютерных платформ. Данные средства должны быть легко используемыми, как и традиционные системы моделирования, иметь возможность расширяться за счет объектов пользователя. Средства Web-ориентированного моделирования должны предоставлять переносимую и расширяемую инфраструктуру с поддержкой интероперабельности для распределенных объектов реализованных на различных языках программирования.

### **Распределенная моделирующая среда**

Распределенным объектом является такой объект, который размещен в любом месте в сети и является доступным, как будто выполняется на локальном вычислительном узле. Фактически, распределенные объекты взаимодействуют в среде гетерогенных платформ. Клиент, получая доступ к распределенному объекту, имеет минимум информации о реальном размещении ресурса. Рассматривая подобное взаимодействие можно говорить о создании некой распределенной моделирующей среды (PMS). Каждый участник данного взаимодействия решает определенный, в некоторых случаях достаточно узкий, круг задач. Среда является открытой и стандартизированной по внешним

интерфейсам, так как основана на принципах объектно-ориентированного программирования. Распределенная моделирующая среда, в зависимости от инфраструктуры сети, может быть организована различными путями. Ядром РМС являются моделирующие компоненты, осуществляющие моделирование конкретных моделей с заданными начальными условиями. Участниками распределенной моделирующей среды могут быть любые объекты, заинтересованные в получении сервиса, предоставляемого РМС. Кроме того, эти объекты могут оказывать непосредственное влияние на общий процесс работы РМС. Например, можно реализовать графический интерфейс пользователя (Graphics User Interface, GUI), посредством которого пользователь сможет контролировать процесс моделирования. Применяя технологию Java applets (апплеты Ява) можно использовать стандартный Интернет-браузер, как средство доступа к распределенной моделирующей среде (рис. 1):

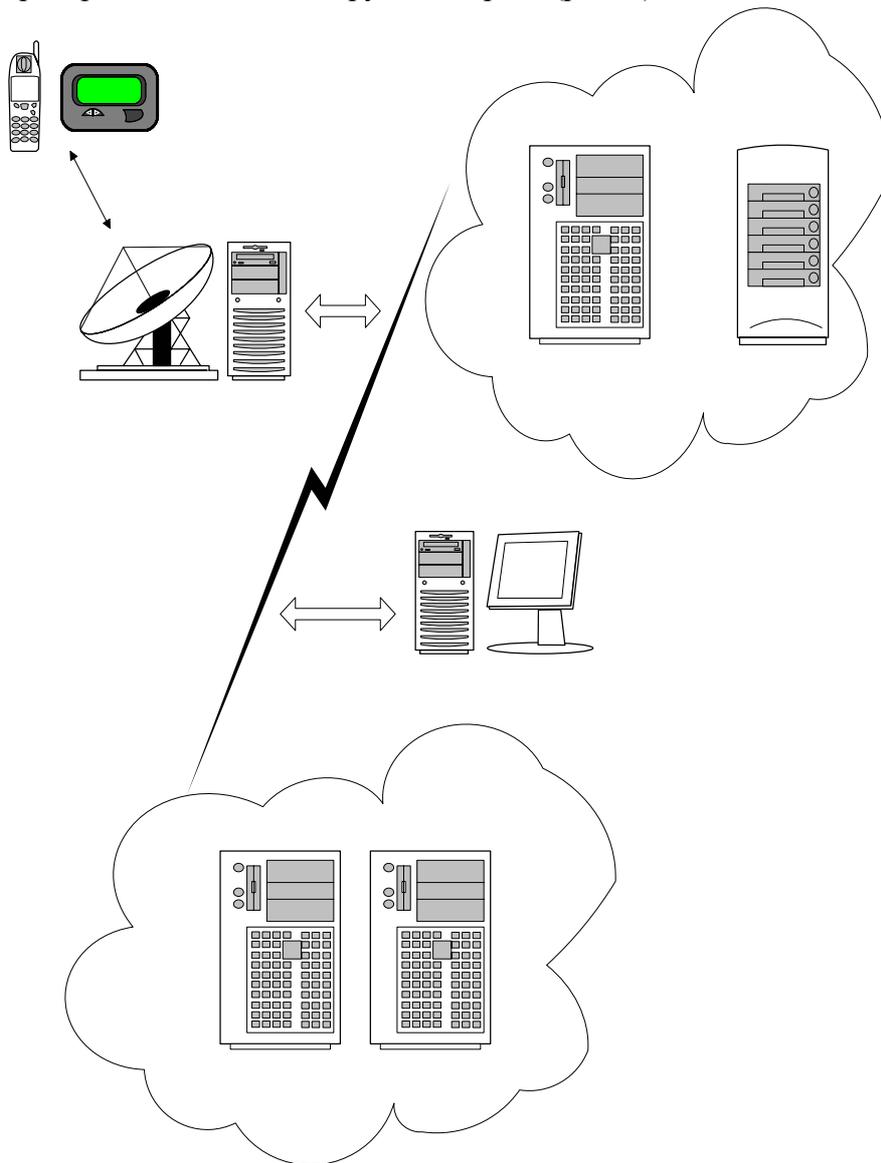


Рисунок 1. – Архитектура  
распределенной моделирующей

### ***DIVA как объект организации распределенного моделирующего узла***

В качестве примера организации распределенного моделирующего узла возьмем систему моделирования сложных технологических процессов DIVA.

Система моделирования DIVA была разработана в 1987-1990 гг. в институте системной динамики и управления Штутгартского университета на базе исследовательских работ, которые проводились в институте в области разработки моделирующих систем сложных химических процессов и моделирования отдельных технологических аппаратов и установок [5].

Система DIVA реализована на языке программирования FORTRAN для операционной системы UNIX. Первые варианты системы представляли собой тестовые варианты несвязных программных модулей.

Современная реализация DIVA позволяет моделировать химические процессы в широком спектре разнообразных задач. С помощью DIVA возможно принятие решения по выбору реализации, как технологического процесса, так и построения технологического аппарата в целом.

Моделирование в системе DIVA является блочно-ориентированным. Модель в DIVA задается в виде автономных блоков, которые имеют набор входных и выходных параметров. В модели можно использовать как стандартные (базовые) блоки, имеющиеся в достаточно большой библиотеке боков, так и специфические элементы, параметры и принципы работы которых задаются пользователем. Математическая модель блока представляется системами алгебраических или обыкновенных дифференциальных уравнений, а также системами уравнений в частных производных с заданными начальными и граничными условиями. В системе DIVA каждому блоку соответствует откомпилированный FORTRAN модуль.

Полное описание модели содержится во входных файлах с расширением .anl и .dat. Файл с расширением .anl содержит описание связей между различными блоками, образуя полное описание модели. Перед началом процесса моделирования выполняется инициализация всех блоков, образующих модель устройства или процесса. Для этого используется описание входных и выходных переменных блоков, их граничные и начальные значения, находящиеся в файле с расширением .dat.

Непосредственная работа с системой DIVA осуществляется в режиме командной строки. Пользователь последовательно вводит различные команды DIVA позволяющие получить/изменить значения переменных, установить время моделирования, получить/установить тип интегратора, стартовать процесс моделирования и др. Для автоматизации процесса ввода команд предусмотрен командный файл, имеющий расширение .do. В данном файле команды DIVA располагается в той последовательности, в которой их необходимо передать системе.

Результат моделирования можно либо получать на консоль, либо сохранять в специальных файлах с расширением .ddf и .dsf. Последующий анализ этих файлов осуществляется либо визуализатором MatLab, либо собственным модулем DivaGraphics.

**Методы организации распределенной моделирующей среды на базе DIVA**

Изначально система DIVA не ориентированна на работу в сетевой среде и не имеет объектных интерфейсов. В общем виде можно построить следующую модель (рис. 2):

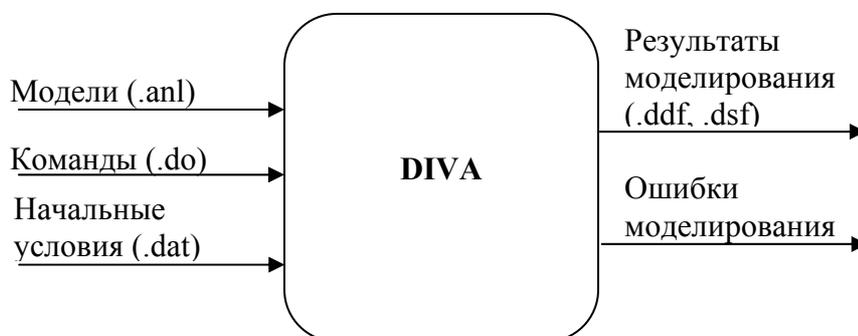


Рисунок 2. – Внешние интерфейсы взаимодействия системы DIVA

Таким образом, можно сказать, что мы имеем вектор входных моделей:

$$\bar{M} = (m_1, m_2, \dots m_n) \tag{1}$$

где  $m_1, m_2, \dots m_n$  предопределенный набор моделей описанных в файле <имя модели>.anl,  $n$  – количество моделей. Как было сказано выше, модель представляет собой набор систем алгебраических и обыкновенных дифференциальных уравнений, а также систем уравнений в частных производных с заданными начальными и граничными условиями.

Для каждой модели существует вектор начальных условий:

$$\bar{N} = (n_1, n_2, \dots n_n) \tag{2}$$

где  $n_1, n_2, \dots n_n$  предопределенный набор описаний начальных условий для соответствующих моделей,  $n$  – количество файлов. Файл должен иметь название <имя модели>.dat.

Набор всех команд системы DIVA можно представить вектором команд:

$$\bar{C} = (c_1, c_2, \dots c_n) \tag{3}$$

где  $c_1, c_2, \dots c_n$  все реализованные команды DIVA,  $n$  – количество всех команд. Команды можно разделить на управляющие, информационные и команды получения помощи. Управляющими командами задается тип модели, параметры моделирования и т.д. При помощи информационных команд можно получить значения текущих параметрах моделирования. Для получения справки по командам DIVA используются команды помощи.

Результаты моделирования сохраняются в файлах <имя модели>.ddf и <имя модели>.dsf.

В процессе работы системы DIVA могут возникать разнообразные ошибки. Их можно описать вектором ошибок:

$$\bar{E} = (e_1, e_2, \dots, e_n) \quad (4)$$

где  $e_1, e_2, \dots, e_n$  стандартный набор ошибок системы DIVA,  $n$  – количество всех возможных ошибок. Ошибки могут быть исправимыми и фатальными. Примером исправимых ошибок может быть ситуации с ошибочным вводом команд, неправильным заданием параметров моделирования и т.д. При этом система DIVA продолжает функционировать. При возникновении фатальных ошибок выполнение программы прерывается.

Для того чтобы создать объектное представление системы DIVA необходимо реализовать объектно-ориентированные интерфейсы к входным и выходным векторам DIVA. Язык FORTRAN, на котором написана DIVA, не является объектно-ориентированным. Среди современных объектно-ориентированных языков программирования, способных взаимодействовать с языком FORTRAN, наиболее рациональным является использование языка C++. В результате получим следующую модель (рис.3):

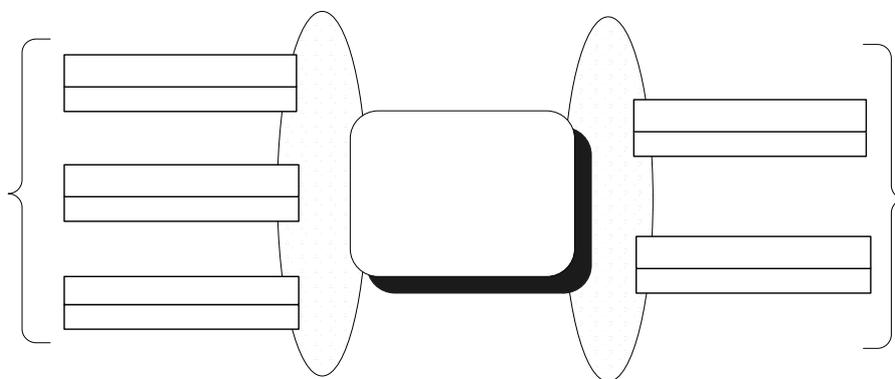


Рисунок 3. – Объектная модель системы DIVA

Таким образом, мы получаем объектное представление системы DIVA. Класс **ModelLoader** предназначен для операций загрузки конкретной модели. В зависимости от потребностей пользователя система предоставляет возможность выбрать ту или иную модель. Класс **CommandsExecuter** предоставляет объектный интерфейс ко всем командам DIVA. Методы этого класса соответствуют набору имеющихся команд. Задачи начальной инициализации покрывает класс **Initializer**. Для доступа к результатам моделирования используется класс **ResultsData**. DIVA имеет семь типов возвращаемых значений, из которых самыми важными являются: Real Parameters, Integer Parameters и String Parameters. Если в ходе моделирования возникают ошибки, класс **DIVAExceptions** предоставляет подробную информацию о возникшей проблеме.

### ***Технология CORBA как средство создания распределенной моделирующей среды***

Имея объектное представление DIVA можно воспользоваться всей мощью технологии CORBA, как средства организации распределенной моделирующей среды.

CORBA (Common Object Request Broker Architecture) является индустриальным стандартом консорциума OMG (Object Management Group) [6].

Данный стандарт описывает взаимодействие между приложениями, написанными на различных языках программирования и выполняющихся в средах отличных платформ.

Основой CORBA является брокер объектных запросов (Object Request Broker, ORB). ORB является той средой, в которой приложения взаимодействуют друг с другом. Клиент использует ORB для того, чтобы сделать вызов объекта, находящегося в любом месте в сети. Когда клиент вызывает метод удаленного объекта, ORB перехватывает вызов и пытается найти подходящий объект. В случае успеха ORB передает методу удаленного объекта список параметров и получает обратно результат работы. Для клиента вызов удаленного объекта полностью идентичен вызову локального. Клиенту не приходится беспокоиться о сетевых протоколах.

Для того чтобы обеспечить возможность удаленного вызова сервисов новой объектной модели DIVA необходимо добавить процедуру инициализации и регистрации DIVA объектов на сервере CORBA. В результате получим следующую схему взаимодействия (Рис 4.):

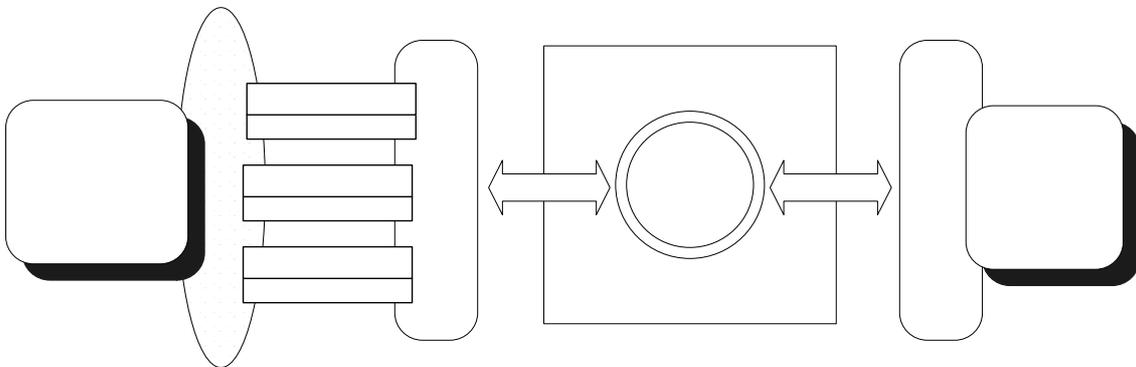


Рисунок 4. – Распределенное взаимодействие на базе CORBA

Первым шагом является определение объектных классов. Для этого используется язык определения интерфейсов (Interface Definition Language, IDL). IDL представляет собой средство указания того, как объекты представляются своим потенциальным клиентам, какие операции допустимы и как они будут вызываться. IDL определяет типы объектов, их атрибуты, экспортируемые методы и их параметры. После этого типичный CORBA-совместимый компилятор обрабатывает IDL-файлы и создает скелетоны (Skeletons) для реализации серверных классов на соответствующем языке программирования. Созданные скелетоны связываются с реализующим кодом. На стороне клиента, на основе IDL-файлов генерируются соответствующие стабы (Stubs). Далее, посредством объектной ссылки, которая может быть получена либо через сервис именования CORBA (NameService), либо через локальный файл осуществляется распределенное взаимодействие.

### ***Практическая реализация распределенной моделирующей среды***

В рамках научных исследований в области организации распределенных моделирующих сред была реализована описанная выше схема взаимодействия. Исходная система моделирования сложных технологических процессов DIVA

посредством реинжиниринга была модифицирована, с целью обеспечить возможность ее работы в распределенной гетерогенной среде. Объектные интерфейсы, добавленные в систему, обеспечивают аналогичную функциональность доступную в исходном консольном режиме работы. В качестве клиента к новой системе был реализован Java апплет, способный работать в окружении современного Интернет-браузера. В ходе тестовых испытаний предложенная методика организации распределенной моделирующей среды себя полностью оправдала.

### **Выводы**

Инфраструктура Интернет в современных условиях является по сути мощной массивно-параллельной средой. Использование описанной методики построения распределенных вычислительных систем, в частности распределенных моделирующих сред, позволит практически на порядок повысить функциональность и быстродействие процесса моделирования. Использование Web как среды развертывания распределенной моделирующей системы позволило перейти на качественно новый уровень организации вычислительной среды, в том числе для целей обучения и тренинга.

### **Литература**

1. Paul A. Fishwick. December 1996. Web-Based Simulation: Some Personal Observations. Proc. 1996 Winter Simulation Conf., pp. 772-779.
2. Klein, U. S. Stra\_burger, and J. Beikrich. (1998). \Distributed Simulation with JavaGPSS based on the High Level Architecture," In: Proceedings of the 1998 SCS International Conference on Web-Based Modeling and Simulation, pp. 85-90, San Diego, CA, 11-14 January.
3. Nance, R.E. (1994). \The Conical Methodology and the Evolution of Simulation Model Development," Annals of Operations Research, 53, Special Volume on Simulation and Modeling, O. Balci (Ed.), pp. 1-45.
4. Page, E.H., Moose, R.L. and Gri\_n, S.P. (1997). \Web-Based Simulation in SimJava using Remote Method Invocation," In: Proceedings of the 1997 Winter Simulation Conference, pp. 468-474, Atlanta, GA, 7-10 December.
5. Аноприенко А. Я., Кинле А., Святный С. Н., Осипова Т. Ф. Моделирование реактора синтеза уксусной кислоты на базе моделирующей среды DIVA // Сборник научных трудов ДонГТУ. Серия "Информатика, кибернетика и вычислительная техника". Выпуск 1 (ИКВТ-97). - Донецк: ДонГТУ. - 1997. - С. 16-21.
6. Object Management Group. The Common Object Request Broker: Architecture and Specification. Object Management Group , Framingham, M.A., Revision 2.2, July 1998
7. Забровский С.В., Мацак С.А., Потапенко В.А., Сисюкин В.С. Опыт использования сети Интернет в качестве распределенной среды моделирования. // Сб. научных праць Донецького Державного Технічного Університету, серія: «Проблеми моделювання та автоматизації проектування динамічних систем», випуск 29, Донецьк, 2001. – С. 254-259.