

УДК 519.83+115.330

К концепции сетевой библиотеки прикладных вычислимых моделей

А. В. Воротынцев

*Отдел математического моделирования экономических систем
Вычислительный центр им. А.А. Дородницына РАН
ул. Вавилова, д. 40, ВЦ РАН, Москва, 119333, Россия*

Обсуждаются элементы архитектуры системы Нива для поддержки сетевой библиотеки прикладных вычислимых моделей.

Ключевые слова: сетевая библиотека, распределённые вычислительные системы, математическое моделирование.

1. Введение

Задача прикладных моделей — стать инструментом конструирования, принятия решений и просвещения для прикладных специалистов. Их число растёт. Число научных журналов удваивается каждые 10–15 лет [1]. Поэтому представляется насущным создание в сети Интернет сетевых библиотек для хранения множества рецензированных численных методов и прикладных вычислимых моделей для их расчёта и использования прикладными специалистами и другими пользователями. Будучи и потребителем, и распространителем моделей библиотеки включают реализации моделей в информационное поле науки. Для многих моделей включение в библиотеку — единственный способ сделать видимым их поведение для таких пользователей. Несомненно, таким сетевым библиотекам предстоит длительный путь неизбежного становления и развития. Но уже сейчас можно реализовать библиотеку с наглядным лёгким доступом, представляющую в сети широкий спектр численных методов и многих полезных вычислимых моделей, решаемых итерациями.

2. Система Нива

В качестве системы для поддержки такой библиотеки рассмотрим систему Нива [2, 3] на рис. 1. Рис. 3 изображает пример графа расчётного задания, сконструированного пользователем, а рис. 5 — пример полученных им результатов. Задача Нивы — предоставить удалённому пользователю, не обладающему полным знанием модели, наглядный универсальный интерфейс для работы с большим числом разнообразных моделей сетевой библиотеки. В меру заданного объёма статьи поясним рисунками элементы архитектуры Нивы, имеющей объёмный программный код около 4000 стр.

Нива состоит из 3 программ: а) сервера моделей `nvServerMod`, на котором хранится банк компонент моделей в исполняемом коде; б) сервера `nvServerDB` трёх баз данных, на котором помещаются база данных `nvParamsDB` стандартных параметров компонент моделей, база данных `nvUserHyModelsDB` параметров компонент моделей, отредактированных пользователем, и база данных `nvUserInfoDB` с информацией о пользователях Нивы, а также в) из программы `nvClient`, помещённой на компьютере удалённого пользователя (рис. 1).

Программа пользователя `nvClient` и программа каждого сервера, размещённые на различных удалённых друг от друга компьютерах, связаны взаимным обменом данными по сетевым коммуникациям Интернет. На рис. 1 сетевые коммуникации

Статья поступила в редакцию 28 ноября 2009 г.

Автор благодарит Ю.Н. Павловского, А.А. Петрова, И.Г. Поспелова, В.А. Серебрякова, А.М. Тарко, Ю.А. Флерова и других сотрудников ВЦ РАН, а также Л.Н. Столярова (МФТИ УНПК), Л.М. Рекса и Ю.П. Добрачева (ВНИИГиМ) за внимание, критику и советы.

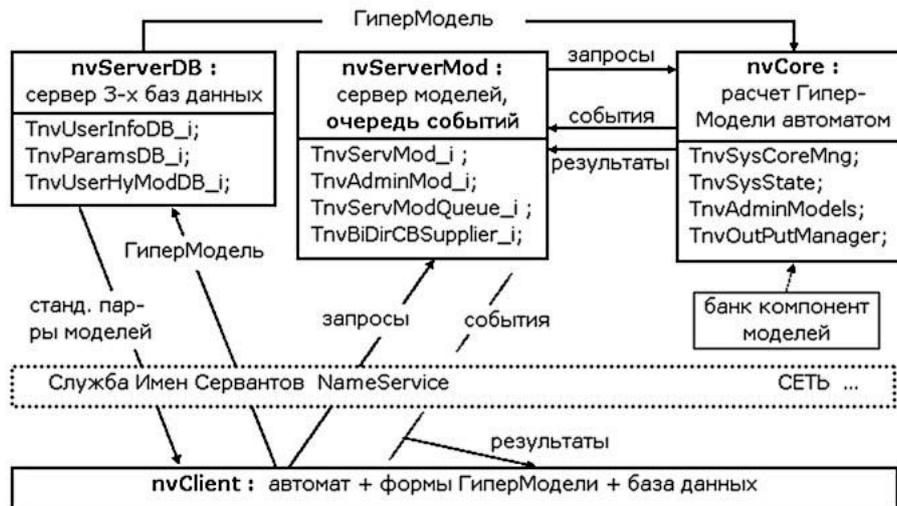


Рис. 1. Обобщённая структура системы Нива

изображены прямоугольником 'СЕТЬ...'. Сетевой обмен данными осуществляется специальными «сетевыми» функциями, реализованными в сервантах. Имена последних отмечены на рис. 1 окончанием '_i'. Особенность сетевой функции в том, что она реализована и исполняется сервантом на одной стороне, а вызывается в программе на другой стороне сетевой коммуникации, задающей параметры и получающей результат исполнения функции. На рис. 1 сетевые потоки данных указаны стрелками.

В Ниве имеется служба `NameService`, которая выдаёт ссылки на серванты и их «сетевые» функции, если пользователь `nvClient` обратился к службе по её известному IP-адресу. Служба `NameService` даёт возможность расширить Ниву на несколько серверов, а также работать нескольким пользователям — в перспективе одновременно.

Ключевыми для концепции Нивы являются понятие МодОбъекта (МО) и понятие ГиперМодели (ГМ). ГиперМодель представляется композицией именованных МодОбъектов `[modName][objName]`, где `[modName]` — имя модели, функциональность которой обозначим через F_i , `[objName]` — имя набора D_i данных, достаточных для расчёта модели F_i . Таким образом, содержательно i -й МодОбъект — это пара $M_i = \{D_i, F_i\}$. ГиперМодель графически представляется графом работ с её МодОбъектами. Под работой понимается исполнение программой функции МодОбъекта. Ребра графа показывают последовательность работ. Граф скрывает обмены данными, а также другие детали реализации. Работа может не иметь содержательной модельной интерпретации. МО может иметь программный код неограниченной сложности и быть не только моделью, но и, например, численным методом, интерпретатором, графором и т.д.

На рис. 2 слева показаны иерархия C++ классов МодОбъекта, а пунктирные стрелки справа и сверху указывают на функции абстрактных интерфейсов. Точка в скобках (.) означает наличие аргументов. Окружение МО — ядро `nvCore` Нивы — получает доступ к объекту класса `TRnvHyModObj`, реализующего компоненту МО ГиперМодели через интерфейсы, вызывая их функции. Вызвав функцию `getIFace(.)` с именем абстрактного интерфейса, ядро получает указатель на этот интерфейс, например на `TnvObject`, а затем исполняет его абстрактные функции, приведённые справа на рис. 2. Название функции поясняет содержание её работы. Например, работа каждого узла графа ГМ на рис. 3 заключается в исполнении функции `DoWork(.)` интерфейса `TnvDoWork`.

ГМ на рис. 3 содержит 11 МО с `[objName]=[standard]`. Всегда присутствующие в ГМ два МО `[Start]` и `[End]` начинают и заканчивают расчёт ГМ. В них

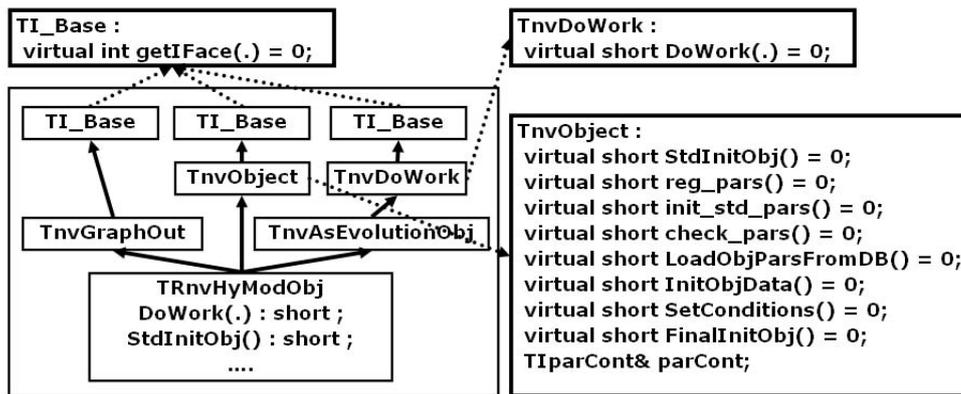


Рис. 2. Иерархия классов МодОбъекта

задаётся интервал времени и число циклов итераций расчёта остальных МО графа. Результаты передаются пользователю в начале каждого цикла. 9 остальных МО содержат объёмную эколого-экономическую модель животноводства, включающую МО [Economy] экономической деятельности домашних хозяйств и государства, МО [MrktLabour] рынка труда, МО [Bank] финансовой деятельности, МО [Biota] продукции травяных ценозов и т.д. МО [LiveStock] организует расчёт модели методом Эйлера с шагом 1 год, последовательно вызывая функцию DoWork(.) каждого МО, которая рассчитывает (т.е. совершает работу) соответствующую МО часть системы уравнений всей модели. Естественная абстракция в виде графа работ обычной блочной вычислительной программы позволила здесь ясно показать информационную структуру модели и экспоненциально увеличить число возможных сценариев её использования.

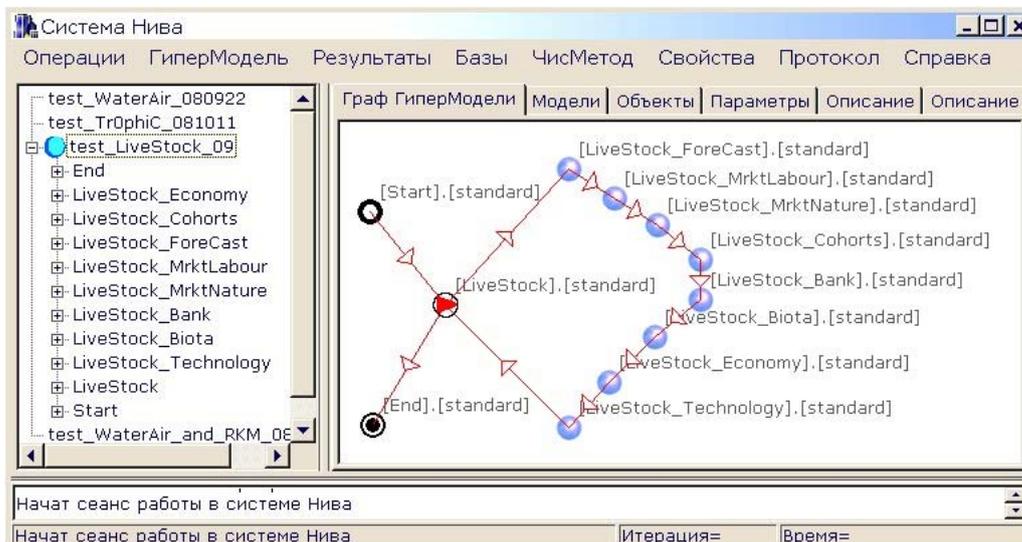


Рис. 3. Пример ГиперМодели, сконструированной пользователем

Графы работ ГМ восходят к диаграммам активностей и классов известного графического языка UML [4]. На рис. 1 показан сравнительно простой граф, в других случаях графы сложнее. Описание вычислимых моделей графами работ — задача исследований. В Ниве есть примеры ГМ, состоящих из 1 изолированного МО; из 2 МО, представляющих пару «численный метод-модель», например

расчёт модели динамики популяций или модели динамики биомассы дерева методом Рунге-Кутты. Есть примеры ГМ типа «численный метод-последовательность МО», представляющие сценарии расчёта переноса в многослойной среде, например одномерного переноса влаги в пористой многослойной почве на основе дифференциально-алгебраической системы уравнений в частных производных, в которых пользователь конструирует профиль как последовательность слоев. Есть пример ГМ для расчёта переноса реагирующей смеси, представленной деревом подграфов. Заметим также, что деревом подграфов работ можно представить дерево арифметического выражения.

Граф ГМ конструируется удалённым пользователем из МО, заранее откомпилированных и хранимых в базе nvParamsDB сервера. Запросив каталог хранимых МО, пользователь мышкой просто перетаскивает ярлыки нужных МО к себе на форму рис. 3 примерно так же, как он копирует файлы в компьютере. При этом описание каждого МО, включая стандартные параметры, копируется в локальную базу пользователя. Подграфы ГМ не являются произвольными и определяются специальными подграфами-грамматиками [3], которые пользователь может наблюдать на вкладке 'Модели' формы рис. 3. Затем можно отредактировать стандартные значения параметров наборов данных D_i МО. Граф ГМ автоматически проверяется на сервере подграфами-грамматиками и сохраняется в базе nvUserHyModelsDB для последующих расчётов.

Такие МО можно иначе называть ситуациями использования модели или её компоненты, а ГМ — сценарием расчёта, или расчётным заданием. Нива позволяет хранить миллионы проверенных ситуаций использования моделей, что напоминает хранение обычных функций в табличной форме. Представление модели сценариями её расчёта совместно с традиционным текстовым описанием как математической модели кажется надёжным и достаточным для пользователей, не обладающих полнотой знания модели и поэтому не имеющих обоснованной возможности изменять её уравнения. Заметим, что к таким пользователям относятся все, за исключением автора, использующего свою модель.

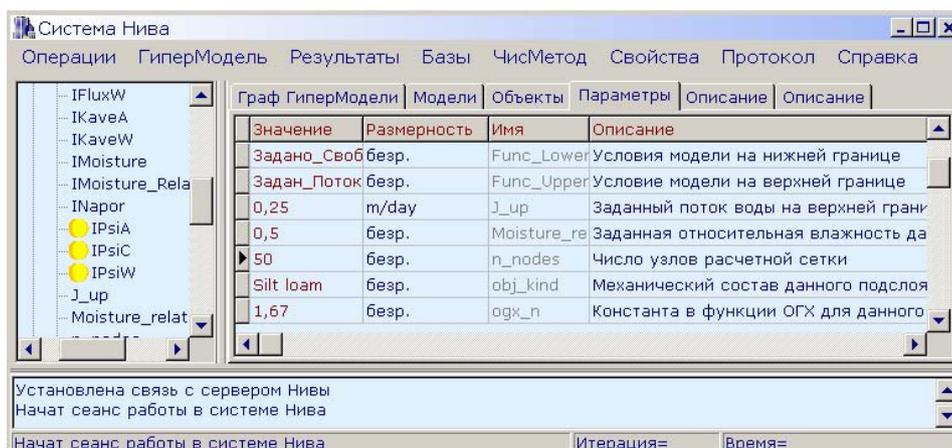


Рис. 4. Форма для изменения параметров МодОбъекта ГиперМодели

Работа пользователя в Ниве состоит из 10 основных стадий: 1) регистрация или проверка регистрации; 2) получение списка ГМ пользователя; 3) получение нужной ГМ или её создание, если ГМ нет в списке (рис. 3); 4) проверка ГМ на сервере nvServerMod с помощью специальных грамматик; 5) ввод своих данных вместо стандартных значений параметров МО, если это было разрешено автором (рис. 4); 6) указание переменных для вывода на графор/текст/слайды (рис. 4); 7) сохранение ГМ на сервере в базе nvUserHyModelsDB; 8) сборка и инициализация ГМ в памяти nvServerMod; 9) расчёт ГМ циклами итераций, наблюдение результатов расчёта указанных переменных, рис. 5; 10) закрытие сеанса и программы nvClient.

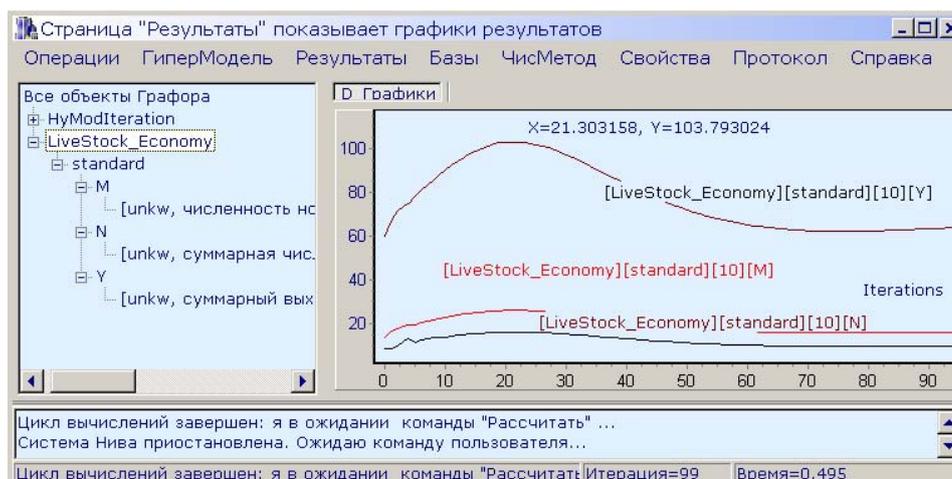


Рис. 5. Результаты расчёта ГиперМодели на рис. 3

Отметим, что форма на рис. 4 позволяет пользователю не только вводить свои значения переменных МО в столбце 'Значение', указывать переменные для вывода на дереве в левой части рис. 4 (указаны IPsiA, IPsiC, IPsiW для модели переноса почвенной влаги), но и получать краткое описание МО в столбце 'Описание', а также в другом столбце 'Заметки' получать привычное текстовое описание МО и модели в форматах txt, pdf или html файла. Таким образом, Нива не только позволяет пользователю библиотеки наблюдать поведение модели на своих данных, но и предлагает сетевую публикацию модели в обычном формате математической модели.

Пользователи библиотеки весьма отличаются в своих потребностях и представлениях от авторов моделей. Поэтому библиотека должна иметь спектр интерфейсов для разных пользователей. Графы работ вместе с текстовым описанием моделей дают, видимо, достаточное представление моделей для большинства пользователей, скрывая при этом детали. Так, полезно скрывать детали, которыми такие пользователи не могут управлять. Например, ГМ на рис. 2 скрывает обмены данными между МО. Также не все авторы захотят публиковать в сети полное описание своих моделей. В таких случаях Нива даёт возможность наблюдать поведение модели как «чёрного ящика».

Напротив, для авторов моделей требуются иные интерфейсы (например, на основе системы символьных вычислений), показывающие все детали и позволяющие создавать МО. Нива стремится включить такие интерфейсы.

3. Заключение

Нива — весьма объёмная и сложная многопоточная распределённая система, управляемая событиями, которые возбуждаются меню пользователя, а также сетевыми коммуникациями и вычислительными процессами на сервере. Чтобы синхронизировать состояния программы nvClient пользователя, программы nvServerMod сервера и состояния сети, используются иерархические конечные автоматы, встроенные в nvClient и сервер моделей nvServerMod. Синхронизация состояний автоматов осуществляется обменом событиями между nvClient и nvServerMod. К событиям, наряду с пересылкой констант, относятся также получение и пересылка nvClient результатов расчёта (рис. 1). Автоматы также существенно структурируют программный код и повышают надёжность управления Нивой, например, пропуская ошибочные команды меню пользователя, не отвечающие состоянию системы.

Литература

1. Земсков А. И., Евстигнеева Г. А. Роль библиотек на мировом рынке научных публикаций // Вестник РФФИ. — 2005. — № 4. — http://www.rfbr.ru/default.asp?doc_id=219494.
2. Воротынцев А. В. К концепции сетевых информационно-вычислительных библиотек // Моделирование и обработка информации. Сборник научных трудов МФТИ. — М., 2008.
3. Воротынцев А. В. О графическом интерфейсе сетевых информационно-вычислительных библиотек // Моделирование и обработка информации. Сборник научных трудов МФТИ. — М., 2008.
4. Кознов Д. В. Основы визуального моделирования. — М.: Интернет-Университет Информационных Технологий. БИНОМ Лаборатория знаний, 2008.

UDC 519.83+115.330

Network Libraries of Computable Models as the Reactive Distributed Systems

A. V. Vorotyntsev

*Department of Mathematical Modelling of Economic Systems
Dorodnicyn Computing Centre of the Russian Academy of Sciences
40 Vavilova, Computing Centre RAS, 119333, Moscow, Russia*

For computable models and numerical methods on the basis of the reactive distributed systems with a graphic interface the concept and the architecture of network object-oriented information libraries are discussed. Calculation of modelling situations is driven by the events and workflow constructed by the user using components saved in library and their data.

The library is intended for saving, easy access and the calculation of wide range set of models solved by iterations, including complex models on the basis of systems of equations of private and ordinary derivatives.

The example of calculation of complex system of mass transfer model and population dynamic model by means of working system Niva is shown.

Key words and phrases: network library, distributed systems, mathematical modeling.