

ПРОХОРОВ Владимир Валентинович

**Технология компонентных моделей
представления знаний
и ее приложения к построению гетерогенных
компьютерных сред**

Специальность 05.13.18 – математическое моделирование,
численные методы и комплексы программ

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
доктора физико-математических наук

Работа выполнена в Лаборатории визуальных систем Института математики и механики Уральского отделения Российской академии наук (ИММ УрО РАН).

Официальные оппоненты:

академик РАН, доктор физико-математических наук, профессор
Красовский Николай Николаевич,

доктор физико-математических наук, профессор
Мазуров Владимир Данилович,

доктор физико-математических наук, профессор
Куропатенко Валентин Федорович.

Ведущая организация:

Институт прикладной математики РАН им. М.В.Келдыша

Защита состоится 4 июня 2003 г. в 10 часов на заседании диссертационного совета Д 212.296.02 при Челябинском государственном университете по адресу: 454021, Челябинск, ул. Бр. Кашириных, 129.

С диссертацией можно ознакомиться в библиотеке Челябинского государственного университета.

Автореферат разослан 29 апреля 2003г.

Ученый секретарь
диссертационного совета
доктор физ.-мат. наук, профессор

Ухоботов В.И.

Общая характеристика работы

Актуальность темы

Для современных технологий характерен существенный рост сложности разрабатываемых и используемых систем. При этом для сложных систем соответствующая модель может включать хотя и тесно связанные, но весьма разнородные (как по внутренней сущности, так и по принятым формам представления) части. Для описания каждой из частей общей модели могут применяться весьма различные средства (например, дифференциальные уравнения, алгоритмические, логические описания, семантические сети, структурные диаграммы, описания структур данных, описания пространственных сцен и др.), что создает проблемы при описании системы в целом.

Задача описания моделей сложных систем значительно усложняется при необходимости отображения этих моделей на компьютерные средства (для информационных систем, систем имитационного моделирования и др.), когда язык описания модели должен отображаться на язык человеко-компьютерного взаимодействия. При этом возникает ряд значительных проблем, в частности – проблема *дистанции* между естественными для специалистов классами моделей, описывающими необходимые сущности предметной области в форме, привычной для этой области, и моделями, которые могут быть непосредственно «поняты» компьютерной средой. Преодоление этой проблемы всегда являлось одной из центральных задач компьютерных технологий, начиная с зарождения этих технологий по настоящее время. Значительные результаты в исследовании этой проблемы в нашей стране получили А.С.Нариньяни, А.П.Ершов, С.С.Лавров, Г.Е.Минц, Д.А.Поспелов, Г.С.Поспелов, Э.Х.Тыгу, Г.С.Цейтин, М.Р.Шура-Бура и др., изучавшие различные ее аспекты.

В связи с этой проблемой возрастает необходимость создания и развития компьютерно-ориентированных информационных моделей, ориентированных на конкретные предметные области, и соответствующих этим моделям средств человеко-компьютерного взаимодействия, обеспечивающих достаточную эффективность работы прикладного специалиста. С другой стороны, увеличивающаяся сложность систем так или иначе влечет возрастание сложности программных средств, что требует предпринятия специальных мер для обеспечения работы пользователя-специалиста с весьма крупными программно-аппаратными системами, включая мультикомпьютерные среды, как с *единым целым*.

Развитие методов создания средств описания моделей сложных многокомпонентных систем и их компьютерной реализации является актуальной задачей, которой и посвящена данная работа.

Один из основных подходов к построению компьютерно-ориентированных средств представления моделей связан с развитием методов сведения моделей различного типа к моделям некоторой единой метафоры и представлением этой монометафорной модели на некотором универсальном компьютерном языке (в частности – языке программирования, например, языке в основном алгоритмической метафоры С, языке функциональной метафоры Lisp, языке логической метафоры Prolog и т.д.). Такой подход исходно был вынужденным, единственно возможным, начиная с первого известного языка программирования реального компьютера «Plankalkül» (K.Zuse, 1945г.). Впоследствии, в связи со стремлением приблизить средства компьютерных моделей к языкам прикладных специалистов, возникли тенденции к постепенному увеличению мощи языковых компьютерных средств, однако подход «минимального языка» с минимумом метафор и сейчас многими исследователями видится, как наиболее разумный (Н.Вирт, Тьюринговская лекция), хотя современная «минимальность» уже весьма относительна.

Стремление найти наилучший язык представления компьютерных моделей для конкретной области привело к тому, что за каждый год компьютерной эры появлялось более 100 новых компьютерных языков; например, в перечне, опубликованном на <http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm> упоминается **2500** универсальных языков программирования (также: М. Зелковиц, Т. Пратт. Языки программирования: разработка и реализация. – СПб.: Питер, 2002 г. – 688 с.; Р.У.Себеста. Основные концепции языков программирования. – М: Изд. Вильямс, 2001). Весьма значимые достижения в области разработки языков программирования имелись в СССР – в качестве примеров можно упомянуть алгоритмический язык Альфа (А.П.Ершов и др., 1960), язык Рефал, базирующийся на концепции нормального алгорифма Маркова (В.Ф.Турчин, 1968), языки аналитических вычислений Аналитик (В.М.Глушков и др., 1968) и Сириус (1970), ряд языков работы с базами данных, язык концептуального программирования Утопист (Э.Х.Тыугу), язык Декарт (С.С.Лавров), и др.

Обилие разработанных языков стимулировало и многочисленные попытки создавать некие всеобъемлющие языки, уменьшить количество используемых языков подчас слиянием основных идей некоторых из групп известных языков («в идеале» – слить вместе почти все известное и получить некий универсальный многометафорный «всеобъемлющий язык», объединяющий возможности многих языков). К таким попыткам можно отнести, по-видимому, проекты PL/1, ADA. Подобные усилия не прекращаются и до сих пор (А.Андреев. Эволюция современных языков программирования. Мир ПК, №3, 2001). Однако попытки создания «всеобъемлющего языка» (или хотя бы нескольких таких языков) наталкиваются на трудности, связанные с весьма большим разнообразием областей сочетания различных типов моделей для сложных систем. Для таких языков характерны громоздкость, сложность модификаций, сложность изучения.

Другой распространенный подход связан с разработкой для каждой прикладной области своих специализированных средств, позволяющих представлять модели из этой области с, по возможности, наименьшими усилиями. Одна из технологий конструирования таких языковых сред связана с так называемыми «Пакетами прикладных программ» (ППП). Данный подход обычно связан с достаточно узкой специализацией; проработка интерфейсных средств работы с ППП как языков обычно незначительна; многие разработки ориентированы на сугубо местное использование. Массовость создания пакетов прикладных программ для частных областей такова, что количество их трудно поддается оценке (возможно, их количество достигает миллиона).

Огромное количество имеющихся и продолжающих появляться языков, многие из которых не очень значительно отличаются друг от друга, создает еще одну дополнительную задачу – их классификации.

Данные проблемы отражаются и в проблемах отображения представлений моделей на архитектуры программных средств и вычислительных систем. Наличие этих проблем было одним из стимулов для развития компонентных технологий построения программных средств. Значительным этапом в развитии компонентных технологий являлась разработка компанией Microsoft технологий OLE 1 (1992), OLE 2 (1993), приведших к архитектурам COM (1995) и DCOM (1996) (включая спецификации составных документов ActiveDoc и компонентных вычислений ActiveX, с механизмами удаленного вызова RPC и Object RPC) и архитектуры CORBA (1991), разработанной объединяющим более 800 компаний консорциумом OMG (в который впоследствии вошла и Microsoft). Разрабатываются и другие подходы (M.Griss, G.Pour, P.Brereton, D.Budgen). Компонентный подход широко используется при разработке приложений электронной коммерции (P.Fingar), банковской сферы (G.Baster, P.Konana, J.Scott), интеллектуальных бизнес-приложений (CHAMOIS) и др.

Развиваются подходы к построению программных средств и языковых средств взаимодействия с ними, основанных на знаниях, в том числе – иерархических (В.Н.Вагин,

А.С.Клещев, Л.В.Литвинцева, А.С.Нариньяни, Г.С.Поспелов, Д.А.Поспелов, Э.Х.Тыугу, В.Ф.Хорошевский), в том числе алгебраические (В.Э.Вольфенгаген, В.Я.Яцук, С.М. Ефимова, Е.В.Суворов) и фреймовые (В.Э.Вольфенгаген, М.Минский, В.Я.Яцук), моделировании поведения сложных систем с интеграцией различных парадигм (Г.С.Осипов).

Рассмотрению проблемы построения «единого универсального языка» посвящена **первая глава** данной работы, в которой предлагается «микроконтекстный» компонентный подход мультиметафорных «слоистых» языков, основанный на представлении языка как сложной системы *микроязыков*. Проблема компонентного языка рассматривается в единстве с компонентным построением программных средств и компонентным построением вычислительных систем.

Для построения компонентных моделей существенно наличие удобных микроязыков, соответствующих некоторым метафорам, для использования в качестве слоев мультиметафорной модели. Существенное значение для удобства работы с описаниями моделей имеет внешняя форма представления. При этом весьма удобны для использования в качестве слоев в компонентных языках (и особо – для компьютерной реализации подхода) графические языки.

Преимущества графических представлений перед текстовыми вызывают интерес исследователей к проблемам визуального представления знаний в различных областях (C.Chen, R.Paul, I.Foster, J.Insley, G.Laszewski, C.Kesselman, M.Thiebaut).

При докомпьютерном проектировании графические описания всегда широко использовались. Скажем, для описания **алгоритмов** широко применялись и применяются граф-схемы потоков исполнения, структурограммы Несси-Шнейдермана, Р-схемы (И.В.Вельбицкий), НПО-диаграммы IBM. Однако указанные представления обладают рядом недостатков, например, схемы потока исполнения ориентированы на представление последовательности команд, а не на отражение структуры, структурограммы ориентированы на представление структуры, но могут быть применены только для представления весьма простых алгоритмов.

Для графического представления **синтаксических описаний** наиболее широко используются «синтаксические диаграммы» (W.Taylor, L.Turner и R.A.Waychoff). Однако этот визуальный язык не ориентирован на структурные представления описаний, кроме того, он позволяет определять только контекстно-свободные грамматики; в то время как, например, почти все языки программирования являются контекстно-зависимыми. Известен также язык синтаксических Р-схем (И.В.Вельбицкий), достаточно близкий языку «синтаксических диаграмм».

Вторая глава работы посвящена построению нескольких новых графических языков, которые удобно применять как элементы предлагаемой в главе 1 технологии «слоистого языка» и в которых преодолены некоторые отмеченные недостатки других графических языков.

Реализация сложных компьютерных моделей может требовать привлечения высокопроизводительных вычислительных ресурсов. Достаточно резкое разделение возможностей компьютера пользователя и высокопроизводительных систем привело к развитию web-доступа к ресурсам высокопроизводительных компьютеров (напр., в отечественных системах WebWS, системе САТУРН, в проекте OGSA, позволяющей интегрировать службы и ресурсы распределенных гетерогенных динамических сред с помощью связывания технологий GRID и WEB, в системе Globus, где GRID-службы обеспечивают наполнение web-служб реальным содержанием для поддержки функционирования распределенных организаций), а также к формированию концепция «метакомпьютинга», включая технологию GRID (напр., в проекте EU DataGrid глобальной европейской инфраструктуры в об-

ласти физики высоких энергий, биологии и систем наблюдения над Землей). При этом компьютер пользователя выполняет основные функции по формированию вычислительной среды для конкретной модели, оказываясь «интегратором», объединяющим ресурсы мощных компьютеров, используемых для данного процесса, а также выполняет функции взаимодействия с пользователем. Взаимодействие компонент при такой интеграции удобно осуществлять с помощью средств, предоставляемых в рамках архитектур COM/DCOM и CORBA). Значительным шагом в развитии технологий метакомпьютинга являлась разработка открытой архитектуры Open Grid Services Architecture (OGSA), инициированная фирмами Globus Project и IBM.

При построении систем метакомпьютинга встает множество проблем: задания пользователем модели, реализуемой в такой среде, взаимодействия процессов в частях метасистемы, максимизации повторного использования компонент, включая старые разработки. Существенное значение имеет решение проблем навигации по доступным компонентам и интеграции метакомпьютинговой системы на уровне сопровождающих описаний компонентов. Весьма существенное значение имеет проблема надежности всего комплекса, состоящего из многих компонент, в условиях возможных отказов частей или линий связи. В отношении повторного использования здесь следует отметить и проблему переносимости программных средств, затрудняющую использование знаний, реализованных в виде текстов программ или откомпилированного машинного кода.

Третья глава работы посвящена разработке технологии использования в метакомпьютинговой системе вычислительных услуг удаленных вычислителей с помощью сервера-посредника (посредника между потребителем вычислительных услуг и вычислительной системой), предоставляющего средства накопления «актуализированных знаний». Технология способствует решению некоторых из отмеченных проблем интеграции высокопроизводительных систем в системы метакомпьютинга на базе компонентной технологии главы 1.

Применение компонентных подходов может быть плодотворным при реализации комплексов программ во множестве областей, например, в системах моделирования, метакомпьютинга, системах медиакommunikаций. При этом в числе актуальных элементов программных комплексов, построенных по компонентной мультиметафорной технологии, можно назвать системы интеграции компонент на базе визуальной сборки (близкие задачи решаются в системах HyperSignal, MathCad и др.), системы организации библиотек вычислительных модулей и удаленного доступа к вычислительным ресурсам (напр., система WebWS, обеспечивающая веб-доступ к высокопроизводительным ресурсам СПбГУ), системы медиавещания и медиаконференций (такие, как системы Microsoft MediaServer/MediaPlayer, RealVideoServer/RealVideoPlayer), модели «медиа по запросу» (W.Wright, G.Horn, P.Knudsgaard, S.Lassen, M.Lubi, J.Rasmussen), средства маршрутизации медиапотоков (H.Fahmi, M.Lstif, S.Sedigh-Ali, A.Ghafoor, P.Liu, L.Hsu), системы передачи видеопотоков научной визуализации (например, система ВизАЭффект с помощью технологии CORBA осуществляет часть визуализации векторных и скалярных полей на удаленных серверах и передает итоговую информацию через интернет на клиентские машины.

Четвертая глава посвящена примерам приложения описанной в диссертации технологии компонентных языков, а также методов визуальных представлений, методов доступа к ресурсам удаленных вычислителей, методов построения компонент мультипарадигменного программного комплекса. Компоненты связаны с областями визуального процедурного программирования, визуальной сборки распределенных приложений, предоставления вычислительных услуг удаленных вычислителей, видеообщения и передачи визуально интерпретированных результатов вычислений через интернет.

Объект исследования

Общие методы построения языков представления моделей сложных многокомпонентных систем, средства человеко-компьютерного взаимодействия с такими системами, архитектуры сложных многокомпонентных программно-аппаратных сред и гетерогенные программные комплексы.

Цель работы

Разработка технологии компонентного «слоистого» представления мультиметафорных компьютерно-ориентированных моделей сложных систем и построения соответствующих программных средств и вычислительных систем.

Разработка визуальных языков как слоев многокомпонентной модели. Разработка технологии интеграции вычислительных услуг удаленных компьютеров в компонентную вычислительную систему. Разработка архитектур программных комплексов для многокомпонентных гетерогенных систем на базе визуальной сборки и компонентной мультиметафорной технологии. Разработка архитектур систем интернет-видеокоммуникаций на базе мультиметафорной технологии.

Научная новизна

Результаты, приведенные в диссертации, являются новыми и имеют как теоретический, так и практический характер.

Для сложных систем разработан подход к организации человеко-компьютерного взаимодействия, языков представления знаний, программных и аппаратных средств как иерархии независимых «микроязыковых» элементов. Подход основан на рассмотрении программных средств/оборудования как сложной системы с применением соответствующей методологии. Предлагается декомпозировать решаемую задачу на иерархию микроконтекстов, каждый из которых может быть наилучшим образом поддержан своим собственным микроязыком (с соответствующими редакторами, компиляторами и интерпретаторами).

Разработаны новый визуальный структурный алгоритмический язык и новый визуальный структурный язык синтаксических описаний. Разработан новый метод порождения адаптируемого под класс задач визуального языка, новые методы структурных графических представлений. Разработан новый метод интеграции фазы редактирования и фазы исполнения программы созданием файловых контейнеров, нагружающих узлы структурной программы.

Предложена технология предоставления пользователю мощностей суперкомпьютера и интеллектуального алгоритмического продукта на базе первичного ресурса («вычислительной услуги») в виде скрытой от него **триады** <программные средства, аппаратные средства, информационная поддержка> с возложением на посредника – «вычислительный прокси-сервер» (ВПС) – выполнения ряда задач, связанных с решением проблем предоставления вычислительной услуги клиенту. Вычислительный прокси-сервер разбивает пару <пользовательский компьютер – вычислитель> на две пары: <пользовательский компьютер – ВПС> и <ВПС – вычислитель>. Изучен набор математических моделей взаимодействия вычислительного прокси-сервера с клиентом и вычислительной средой, исследован ряд возможных стратегий вычислительного прокси-сервера по оптимальному использованию ресурсов.

На основе общей технологии компонентного построения мультиметафорных гетерогенных систем разработан ряд частных технологий. При этом разработаны архитектуры компонент комплекса программных средств, включающего:

- среды визуального алгоритмического программирования – ПΥΘΑΓΩΡΑΣ и πJ ,
- среду визуальных синтаксических описаний ПΥΘΑΓΩΡΑΣ-S,
- систему визуального конструирования на базе функциональных схем HyperModeller,
- систему предоставления доступа к вычислительным услугам «Вычислительный прокси-сервер»,
- комплекс средств *ViDoors* для распределенной среды интернет-медиа-коммуникаций (как для межлического общения, так и для использования в составе многомашинных вычислительных систем).

Методы исследования

Методы исследования базируются на методах анализа больших систем, построения компонентных программных средств, включая концепции Corba, COM, DCOM, технологиях визуального программирования, технологиях метакомпьютинга, технологиях интернет-медиакоммуникаций.

Теоретическая значимость

Основные идеи, методы и утверждения, предложенные в диссертации, могут использоваться при разработке языков программирования и проведении сравнительного анализа существующих языков, при построении программных сред, связанных с большими системами, при построении сред визуального программирования, при разработке распределенных гетерогенных систем, в том числе включающих суперкомпьютеры, при создании систем накопления актуализируемых знаний, при разработке систем интернет-медиакоммуникаций.

Практическая значимость

Исследования по теме диссертации выполнялись автором в течение 1986-2003гг. в рамках следующих основных плановых тем Института математики и механики УрО РАН:

- «Управление объектами в условиях неполноты и искажений априорной и текущей информации с использованием методов обработки изображений и распознавания образов», 1986-1990гг. (приоритетное направление фундаментальных исследований РАН 1.12.7 (1.13.4) «Управление движением и навигация»). Руководитель Ю.С.Осипов.
- «Управление объектами в условиях неполноты и искажений априорной и текущей информации с использованием методов обработки изображений и распознавания образов». № гос. регистрации 01.9.10-028753, 1991–1995гг. (приоритетное направление фундаментальных исследований РАН 1.12.7 (1.13.4) «Управление движением и навигация»). Руководители В.Г.Гасилов, В.В.Прохоров.
- «Информационная технология и инструментальные средства мультязыкового программирования многокомпонентных вычислительных сред (модели организации высоко производительных вычислений в среде Интернет)», № гос. регистрации 01.9.60-002488, 1996–2000гг. (приоритетное направление фундаментальных исследований РАН 2.2.7 «Интегрированные информационные и

дований РАН 2.2.7 «Интегрированные информационные и телекоммуникационные системы и сети»). Руководитель В.В.Прохоров.

- «Технологии и программные средства многомашинных полиязыковых компьютерных систем и визуального человеко-компьютерного взаимодействия». № гос. регистрации 01.2.00-102382, 2001–2005гг. (приоритетное направление фундаментальных исследований РАН 2.2.8 «Архитектура, системные решения и программное обеспечение»). Руководители В.И.Бердышев, В.В.Прохоров.

Исследования автора были поддержаны грантами:

- Грант по федеральной программе Министерства науки и технологий РФ «Информатизация России»: «Разработка инструментальных средств многосредового (гипермедиа) объектно-ориентированного программирования на основе визуальных представлений». 1993г. Руководитель В.В.Прохоров.
- Грант по федеральной программе Министерства образования РФ «Информатизация образования в Российской Федерации»: «Создание программного комплекса для работы на основе иерархической информационной технологии и образных представлений». 1994-1995гг. Руководитель В.В.Прохоров.
- Грант РФФИ №93-01-00989 «Технология иерархического мультимедийного человеко-машинного взаимодействия и программный комплекс ее поддержки». 1993-1995гг. Руководитель В.В.Прохоров.
- Грант РФФИ №99-01-00468 «Технология применения вычислительных серверов». 1999-2001гг. Руководитель В.В.Прохоров.
- Грант РФФИ №02-01-96419 «Разработка методологических основ интеллектуальной составляющей информационно-аналитических систем поддержки принятия решений в управлении экономическими структурами». 2002-2003гг. Руководитель О.И.Никонов.
- Грант РФФИ №02-07-90361 «Разработка методов и программных средств сжатия информации для организации распределенных мультимедиа приложений с высоким качеством видео и для передачи больших объемов данных между персональным и высокопроизводительными компьютерами». 2002-2004гг. Руководитель В.В.Прохоров.

Кроме этого, исследования поддерживались федеральной целевой программой «Интеграция науки и высшего образования России» (2002–2006гг.), проектом «Разработка методик обучения и интегрированных курсов по суперкомпьютерным технологиям», государственным контрактом № У0036/2091, и рядом коммерческих контрактов, в частности: «МPEG-4 Coding JMF Plugins and Applications» с компанией YunHar (г.Сеул, Корея, 2000-2001), «Программные средства телерепортажного комплекса передачи видео-аудио потоков через спутниковые каналы связи» с компанией «Ин тек коммуникации» (г.Москва, 2002, 2003, комплекс демонстрировался на Международной выставке телевидеотехнологий IBC-2002 в г.Амстердаме), «Компьютерная система оптимизации финансовых и товарных потоков предприятия» с Уральской горно-металлургической компанией (г.В.Пышма, 2001).

На базе полученных по теме диссертации результатов построен спецкурс «Информационные модели и языки программирования», который преподается студентам УГТУ по специальности 220100 – «Вычислительные машины, комплексы, системы и сети» (1996-2003).

Полученные результаты использовались при разработке ряда программных комплексов, например, среды ПУΘΑΓΩΡΑΣ, использовавшейся для разработки системы медицинской диагностики (получен патент РФ), а также получившей гриф «Рекомендовано Отде-

лом информатизации Министерства образования Российской Федерации» для применения в сфере образования; среды ^πJ, использовавшейся при обучении студентов УГТУ, УрГУ и на факультативных занятиях школьников, среды HyperModeller, использовавшейся при разработке системы моделирования технологических процессов предприятия и системы конфигурирования и мониторинга интернет-узла Уральского отделения РАН, а также в учебном процессе при подготовке студентов. Полученные результаты по мультипарадигменной иерархической технологии и разработанные средства медиаинтернет использовались НПО УралСистем при разработке систем обеспечения работы законодательных органов, эксплуатируемых в настоящее время, например, в Законодательном собрании Челябинской области. Разработанная технология использована в системе «Мобильный репортерский спутниковый комплекс» фирмы «Ин тек коммуникации». Разработанные системы интернет-видеовещания использованы при трансляции и организации видеомостов на ряде общественных событий Екатеринбурга и России, эксплуатируются Российским фондом фундаментальных исследований, Уральским профсоюзным центром.

Апробация работы

Результаты диссертации докладывались на следующих конференциях и семинарах:

► IV, V, VII Всесоюзные семинары «Разработка и применение программных средств ПЭВМ в учебном процессе» (Симферополь 1988, Орджоникидзе 1989, 1991). ► VII, VIII, IX Сибирские школы по пакетам прикладных программ (Иркутск, 1989, 1992). ► Региональная конференция «Диалог «человек-ЭВМ»» (Свердловск, 1989). ► IV Всесоюзная конференция «Математические методы распознавания образов ММРО-IV» (Рига, 1989). ► IV научно-методическая конференция «Проблемы образования в области информатики, вычислительной техники и автоматизации» (Ленинград, 1990). ► Всесоюзная научная конференция «Интеллектуализация систем управления» (Баку, 1991). ► Международный симпозиум «Visual Analysis and Interface, VAI'91» (Новосибирск, 1991). ► Международная конференция «1-st Moscow International HCI'91 Workshop» (Москва, 1991). ► Международная конференция «Применение новых компьютерных технологий в образовании» (Троицк Моск. обл., 1991, 1992, 1994, 1996). ► Международная конференция «East-West Conference on Emerging Computer Technologies in Education» (Москва, 1992, Киев, 1994). ► Региональная научно-практическая конференция «Информатизация образования'93» (Екатеринбург, 1993). ► Международная конференция ICTE-93 (Киев, 1993). ► Выставка-конференция «Новые информационные технологии в школах и вузах» (Москва, 1993). ► IV Национальная конференция с международным участием «Искусственный интеллект» КИИ-94 (Рыбинск, 1994). ► Всероссийская конференция «Новые информационные технологии в образовании» (Москва, 1994). ► I Российско-турецкий семинар «Новые информационные технологии» (Москва, 1994). ► II Российско-турецкий семинар «New High Information Technologies» (Стамбул, Турция, 1994). ► 19-я международная конференция «Information Technologies and Programming» (София, Болгария, 1994). ► Международная конференция «East-West International Conference on Multimedia, Hypermedia and Virtual Reality. MHVR'94» (Москва, 1994). ► Международная научно-практическая конференция «Языки мозга и тела человека» (Орел, 1995). ► Международная конференция «East-West International Conference on Human-Computer Interaction, EWHCI», (Москва, 1995, 1996). ► Международная конференция «Second International Symposium on Knowledge Acquisition, Representation, and Processing, KARP-95» (Оборн, Алабама, США, 1995). ► Международная конференция «Expert Systems Applications & Artificial Intelligence, EXPERSYS-95» (Сан-Франциско, США, 1995). ► Международная конференция «Interactive Systems: The Problems of Human-Computer Interaction» (Ульяновск, 1995, 1999). ► Международный семинар «Искусственный интеллект в образовании» (Казань, 1996). ► Международная школа «Media communications on the Internet» (София, Болгария, 1998). ► Всероссийская научная конференция «Высоко-

производительные вычисления и их приложения» (Черноголовка, 2000). ► Всероссийская научная конференция «Научный сервис в сети Интернет» (Абрау-Дюрсо, 2001, 2002). ► Международная конференция «Parallel Computing Technologies PaCT-2001» (Новосибирск, 2001). ► Международная конференция «3rd International Workshop on Computer Science and Information Technologies» CSIT'2001 (Уфа, 2001). ► Первая научно-практическая конференция «Современные информационные технологии в образовании» (Ханты-Мансийск, 2002). ► Научно-техническая конференция «Проблемы создания единого информационного пространства в рамках СНГ» VII Российского экономического форума (Екатеринбург, 2002). ► Региональная молодежная конференция «Проблемы теоретической и прикладной математики» (пленарные доклады) (Екатеринбург, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003). ► II Межрегиональный семинар «Новые информационные технологии сопровождения заседаний и правотворческого документооборота властных органов» (Екатеринбург-Челябинск, 2002). ► Всероссийская конференция «Актуальные проблемы прикладной математики и механики», посвященная 70-летию со дня рождения академика А.Ф.Сидорова (Екатеринбург, 2003). ► Совещание комиссии РАН по высокопроизводительным вычислениям (Москва, 2002). ► Совещания по высокопроизводительным вычислениям во Всероссийском ядерном центре (Пленарные доклады) (Снежинск, 1996, 1997). ► Семинары в Институте прикладной математики РАН (Москва, 1997, 2000). ► Семинар в компании «Центр спарк-технологий» Б.А.Бабаяна (1998, 1999). ► Семинар в представительстве компании Microsoft в России (И.Р.Агамирзян, 1998). ► Заседание Президиума УрО РАН (2002). ► Семинары УрО РАН по технологиям высокопроизводительных вычислений. ► Заседания Ученого совета ИММ УрО РАН.

Публикации

По теме диссертации опубликованы 96 работ, в том числе приведенные в конце автореферата работы [1-28]. Получен 1 патент на изобретение [29].

Из совместных работ в диссертацию включены результаты, полученные лично автором и не затрагивающие интересов других соавторов.

Структура и объем работы

Диссертация состоит из введения, четырех глав, заключения, приложения и списка литературы. Объем работы – 297 страниц основная часть и 28 страниц приложения.

Краткое содержание и основные результаты диссертационной работы

Во *введении* обоснована актуальность направлений исследования, приведены постановки рассматриваемых задач и дан обзор работ, близких к теме диссертации.

Первая глава работы посвящена проблеме создания «наилучшего языка» для представления знаний в какой-либо частной области. Проблема обсуждается с точки зрения противоречия между богатством описательных средств языка и его компактностью, гибкостью и универсальностью. Рассмотрен подход к организации языков представления знаний, человеко-компьютерного взаимодействия, программных и аппаратных средств как иерархии независимых элементов. Этот подход основан на рассмотрении программных средств/оборудования как сложной системы с применением соответствующей методологии. Главное внимание уделяется примеру языков человеко-компьютерного взаимодействия, включая программирование.

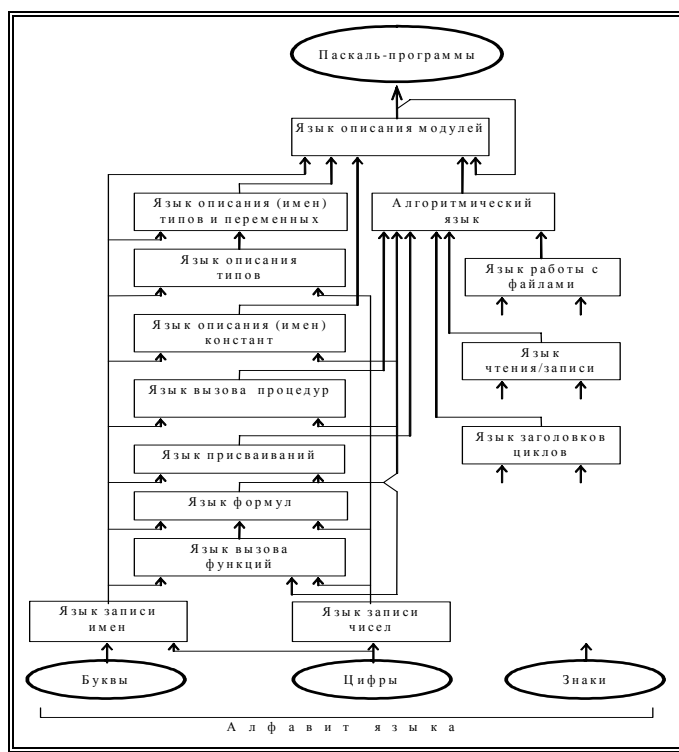
В *первом параграфе* главы 1 дается общий обзор проблемы.

Во втором параграфе главы 1 рассматриваются различные известные подходы к построению информационных описаний.

В третьем параграфе главы 1 проводится анализ сложной информационной среды, как иерархии. Обсуждаются статические микроязыковые структуры, разбиение сложных языков на меньшие части как первый шаг к открытой слоистой многоязыковой технологии. При этом *любой* компьютерный язык рассматривается как, прежде всего, средство представления некоторых информационных моделей, а не как средство задания процесса функционирования компьютера.

В качестве базиса, формирующего язык, рассматриваются пары {[набор базовых элементов], [набор средств композиции/декомпозиции]}. Примерами таких пар являются алгоритмические средства (базовые элементы – из некоторого набора действий, а средства композиции/декомпозиции – алгоритмические конструкции), функциональные (формульные) средства (базовые элементы – из некоторого набора функций, а средства композиции/декомпозиции – суперпозиция функций, условное определение функций, рекурсивное определение), логические (предикатные) средства (позволяющие строить сложные логические высказывания из примитивных), «геометрические» средства (позволяющие строить сложные графические элементы из простых посредством, скажем, сдвига-поворота, наложения с просвечиванием, и т.д.), правила создания сложных типов данных из простых (например, подязык Хоара в Паскале), группа средств для работы со звуками и речью, средства синтаксических определений (позволяющие строить сложный синтаксический объект из примитивных).

Каждая такая пара может быть сопоставлена некоторому специфическому виду информационных моделей. Можно считать, что микроязыки, соответствующие каждому из этих наборов, – это инструменты композиции/декомпозиции, связывающие некоторые сущности с производными от них.



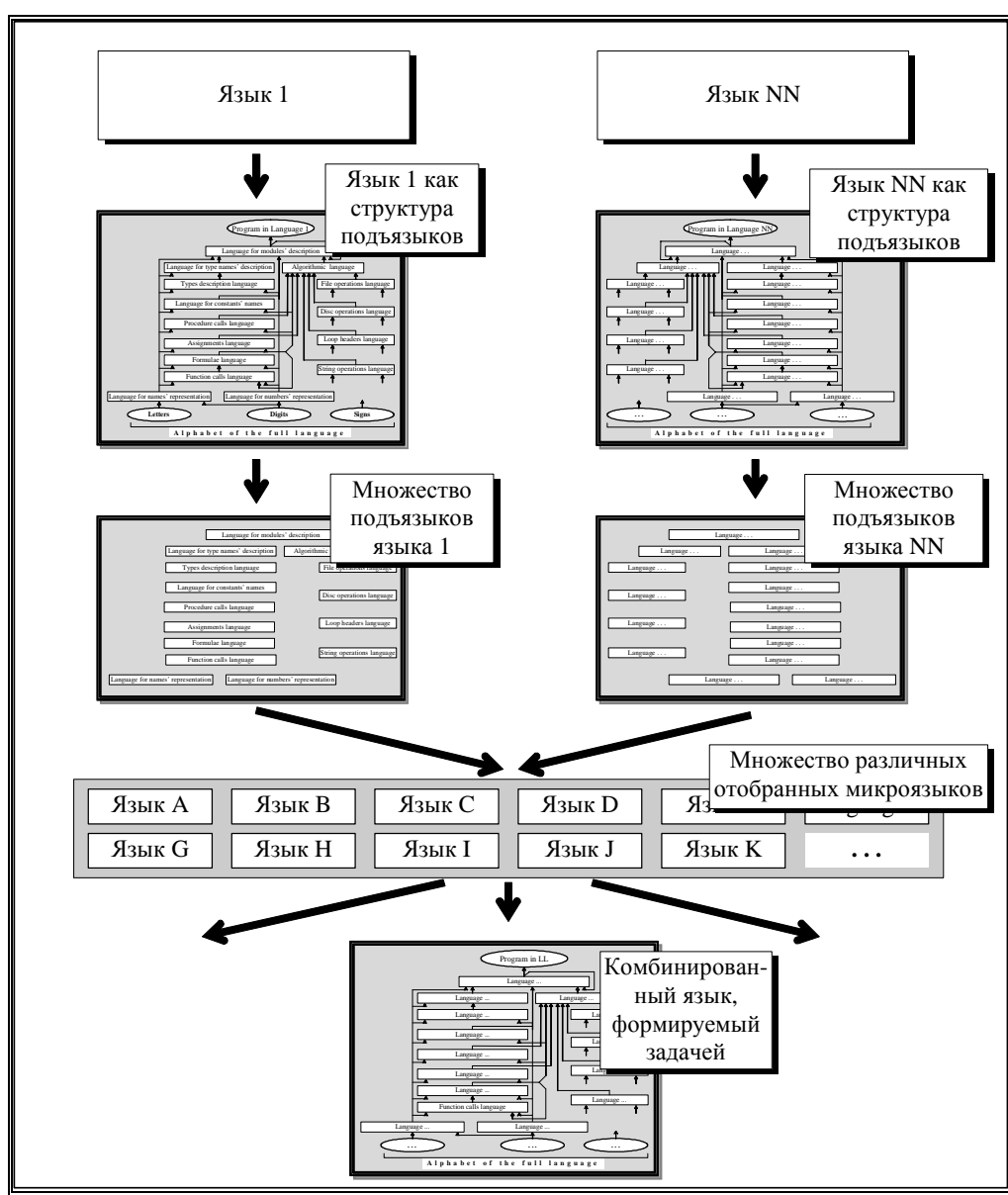
Фиг. 1. Пример выделения подязыков в языке Pascal

Плодотворным является рассмотрение сложных сущностей как композиции более простых с применением некоторых средств композиции. При повторении подобной процедуры можно получить представление сложной модели в виде структуры более простых

моделей. При этом можно рассматривать описание на любом языке как *сложную структуру* или иерархию *моделей*. Идея «анатомирования» сложных языков полезна как сама по себе, так и для определения наилучших экземпляров реализации в каждой группе правил (на Фиг. 1 приведен пример «анатомирования» языка Pascal).

В четвертом параграфе главы 1 вводится концепция мультипарадигменной микроконтекстной иерархической «π-технологии».

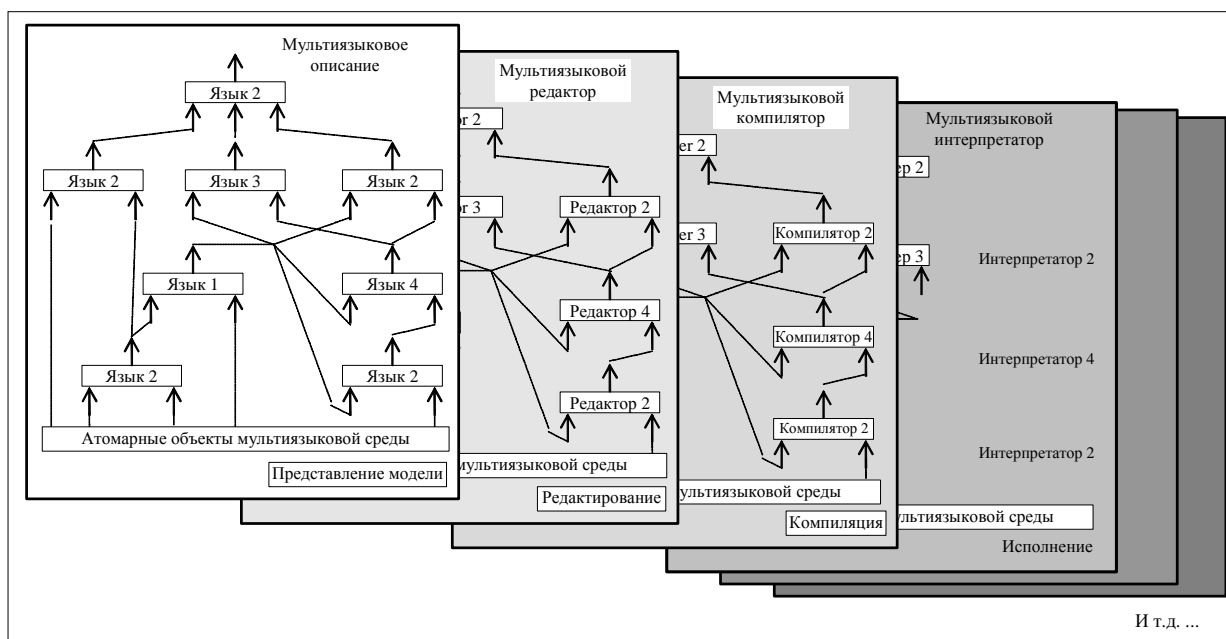
Описывается обобщение идеи разбора сложных языков, приводящее к микроконтекстному подходу. Прежде всего, дополнительно к описанному выше можно *освободить* связи между подязыками, выделенными при анатомировании различных языков. Тогда эти подязыки можно рассматривать как «сырье» для сборки новых языков. Это позволяет динамически строить ориентированный на задачу язык представления, наилучшим образом подходящий для конкретной частной проблемы (Фиг. 2). Кроме того, каждый из подязыков может обладать наилучшей формой внешнего представления, например, в одном комплексе могут быть смешаны текстовые и визуальные подязыки.



Фиг. 2

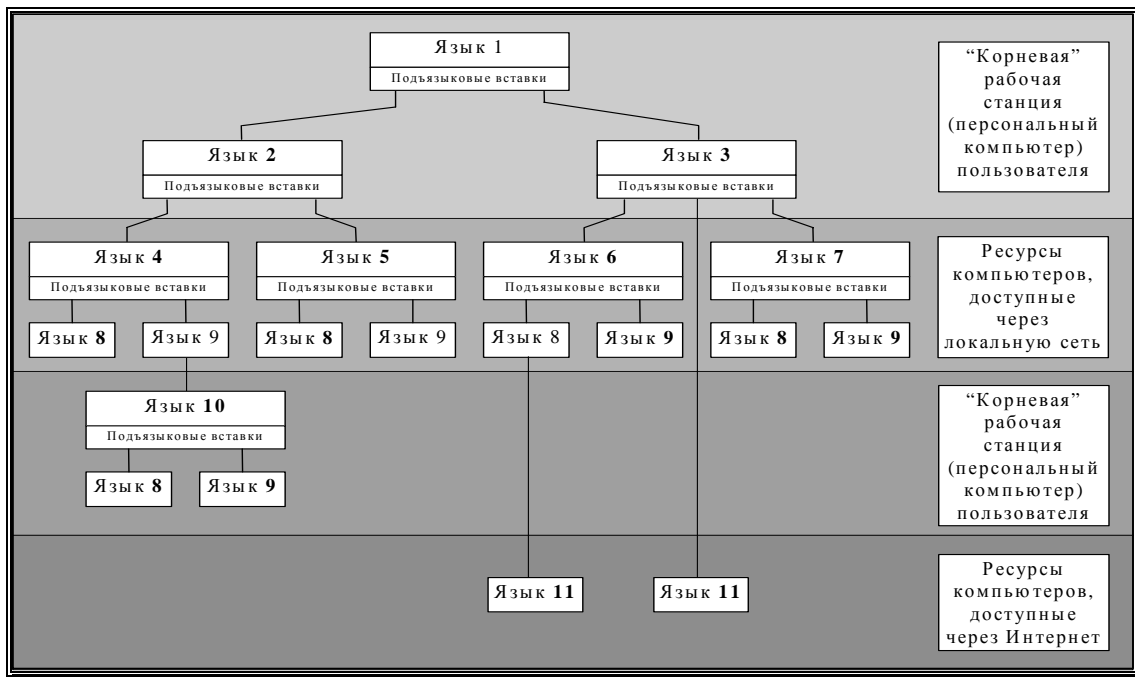
Затем рассматриваются различные «стадии существования» описания знания, включая связанные с использованием компьютерной техники. Например, можно выбрать ста-

дии создания, редактирования, компилирования, интерпретации и уничтожения описаний. При этом для каждого микроязыка желательно иметь программное обеспечение, поддерживающее каждую стадию существования описания, причем компоненты этой программной среды могли бы работать совместно в различных комбинациях, адаптивно формируя ориентированные на задачу редактор, компилятор, интерпретатор и т.д. (Фиг. 3).



Фиг. 3. Иерархия программных средств для стадий существования описания

Далее рассматриваются различные материальные среды («несущие» аппаратные вычислительные средства), поддерживающие каждую из стадий существования каждого из микроязиков. Таким образом, можно говорить о компьютерной архитектуре, динамически синтезируемой для каждой конкретной задачи (Фиг. 4).



Фиг. 4. Иерархия языков и иерархия оборудования

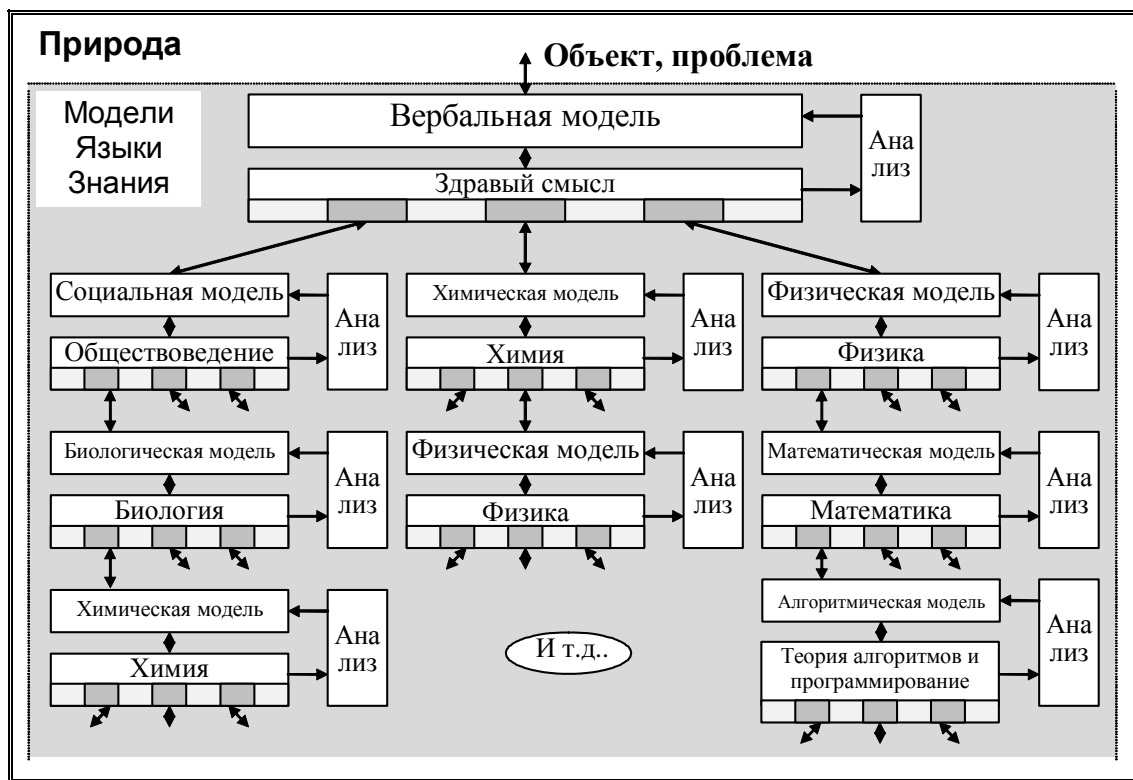
Разработана общая компонентная технология построения языков информационных описаний, организации программных и аппаратных средств, процесса создания про-

граммных продуктов. Предложен общий мультипарадигменный подход, основанный на декомпозиции решаемой задачи на иерархию микроконтекстов, каждый из которых может быть наилучшим образом поддержан своим собственным микроязыком и соответствующими сопряженными программными и аппаратными средствами.

Содержание технологии предполагает следующее.

1. Всякое информационное описание, исключая базисное, рассматривается как иерархическая структура, суперпозиция более простых информационных описаний (слоев), каждое из которых построено средствами своей информационной системы.
2. Каждая из используемых информационных систем ориентирована на весьма специализированный, узкий класс операций, множество ее правил весьма компактно.
3. Для каждой информационной системы имеются: (а) независимая внутренняя концепция языка описаний, (б) независимая форма внешних представлений спецификаций.
4. Существование слоя описания разделяется на фазы, такие как (а) разработка описания-модели, (б) ввод/редактирование, (в) вывод на экран и на печать, (г) интерпретация и/или конвертирование, (д) компиляция, и др.
5. Для каждого слоя и каждой фазы имеются независимые, вообще говоря, (а) компьютерное оборудование, (б) программная поддержка. (Эти программное обеспечение и оборудование могут быть созданы независимо, иметь, вообще говоря, свою независимую фирму-изготовитель.)
6. Существуют соглашения, поддерживающие многоцелевое использование каждой из компонент (5), неединственность пар слоев, для которых эти компоненты можно использовать совместно.
7. Имеется возможность самостоятельного комплектования иерархической программной системы из программных и аппаратных средств поддержки языковых слоев (5).

Компонентное представление знаний может рассматриваться и с существенно более общих, нежели компьютерные среды, позиций (Фиг. 5, Фиг. 6).



Фиг. 5. Диаграмма отношений наук для некоторой проблемы, как иерархия моделей



Фиг. 6. Пример диаграммы уровней в возможной иерархии моделей

Разработанный подход можно рассматривать как обобщение концепции модульного проектирования на ситуацию, где каждый из иерархически связанных модулей является контейнером-«конструктом», связывающим свои субмодули своими собственными средствами (включая и свои внешние формы представления), независимо от средств, используемых другими модулями структуры (причем для человеко-компьютерного взаимодействия в целом, а не только для области «программирования»).

В пятом параграфе главы 1 описываются основные характеристики предлагаемой технологии.

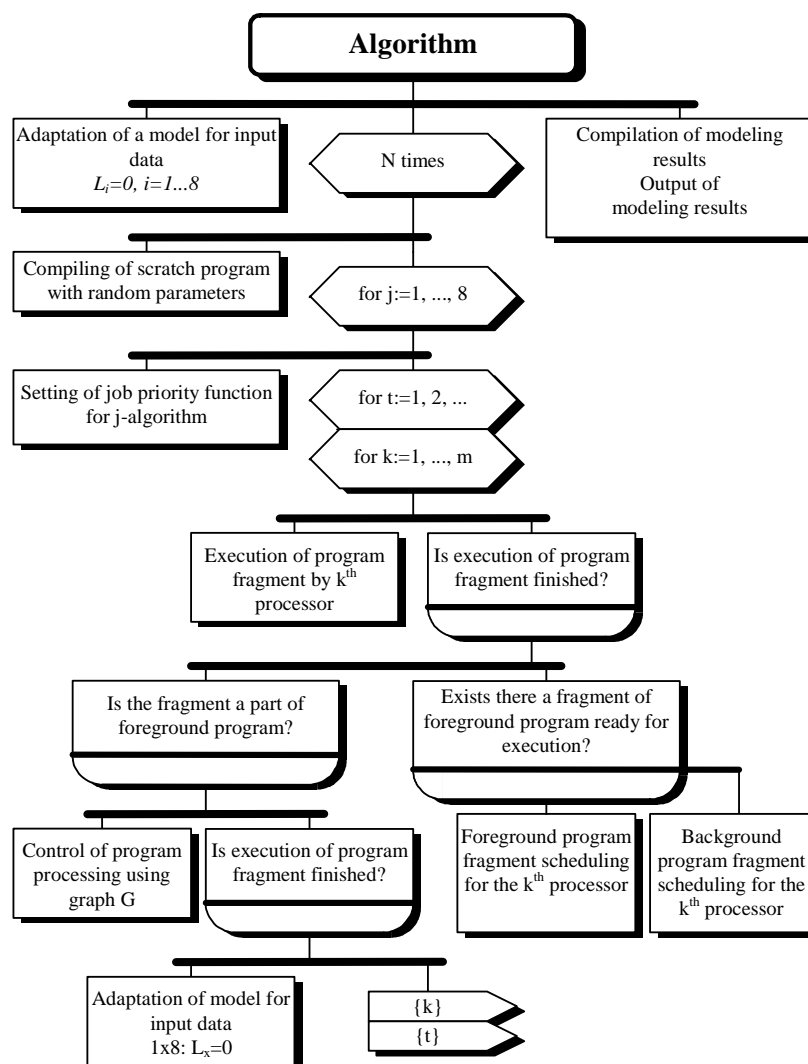
В шестом параграфе главы 1 рассматривается модель оценки и оптимизации затрат в многоуровневых компонентных средах на базе экспертных оценок. Предложен аппарат оптимизации на основе динамического программирования, позволяющий выбрать оптимальный для конкретной задачи набор языков, оборудование, специалистов в случае фиксированных ресурсов. Исследовано обобщение подхода на нечеткий случай для учета степени уверенности и квалификации экспертов. Изучены вопросы теоретико-вероятностной интерпретации подхода в условиях неопределенности ресурсов. Рассмотрен простейший пример, иллюстрирующий предложенный подход к оптимизации затрат.

Вторая глава работы посвящена вопросам реализации визуальных микроязыковых слоев для технологии, описанной в главе 1, соответствующих некоторым информационным парадигмам.

В первом параграфе главы 2 дается общий обзор проблемы.

Во втором параграфе главы 2 рассматриваются визуальные языки представления алгоритмов.

Предлагаемый автором графический язык, в отличие от наиболее распространенного графического языка схем потока исполнения, ориентирован на изображение структуры, устройства алгоритма, а не последовательности исполнения команд, хотя последовательность исполнения операторов он полностью описывает. Этим данный язык идейно близок к языку «структурограмм» Несси-Шнейдермана, однако, как представляется, удобнее для изображения алгоритмов на бумаге и экране компьютера и нагляднее представляет структуру алгоритма. Язык π -схем ориентирован исключительно на структурный подход к алгоритмизации (неструктурный принципиально невозможен), на разработку алгоритмов (и возможность их изучения) «сверху вниз»; при этом термин «сверху вниз» приобретает буквальный смысл (Фиг. 7).



Фиг. 7. Пример π -схемы алгоритма

Такие элементы языка, как <Подстановка>, <Синонимия>, <Склейка> позволяют визуализировать понятия «условное значение», «значение составного оператора», «условное имя», «условная операция» и т.п.

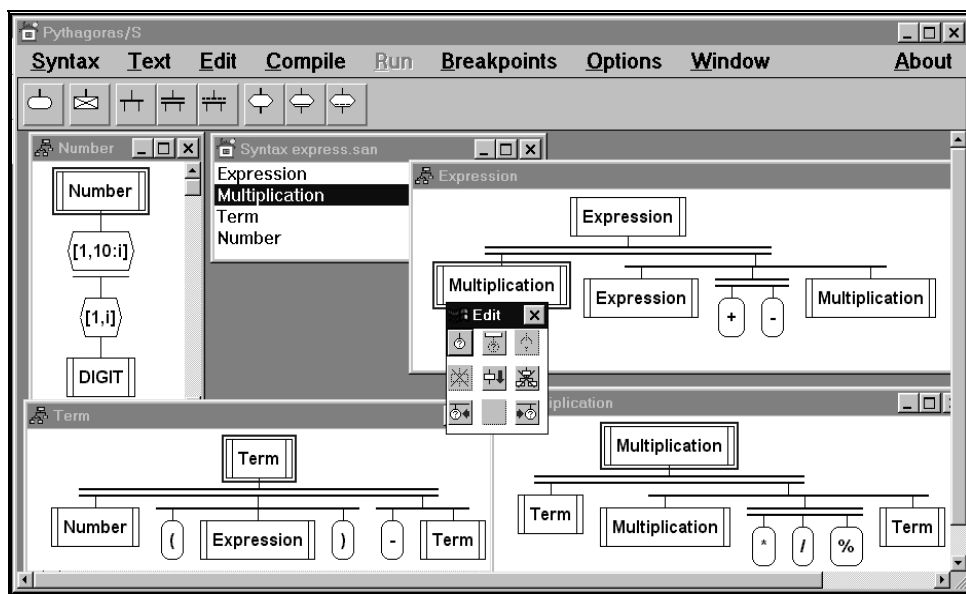
Язык π -схем может использоваться как средство наглядного описания алгоритмов при общении «человек-человек», скажем, алгоритмов технологических процессов, а также как язык описания алгоритмов в компьютерных системах (например, вычислительных algo-

ритмов, алгоритмов управления подключенным к компьютеру оборудованием, скажем, станком с ЧПУ и т.п.).

Алгоритмический язык π -схем удобен для использования как элемент структуры «слоистой технологии» и реализован в средах визуального программирования ПΥΘΑΓΩΡΑΣ и πJ (описанных в главе 4).

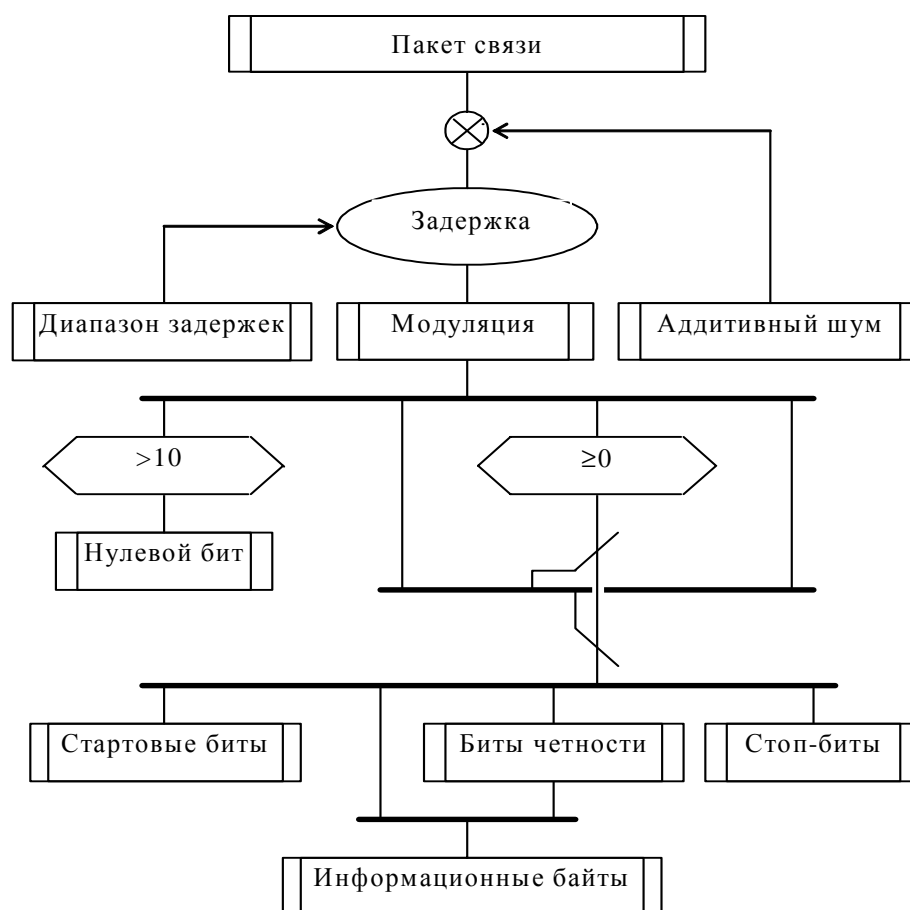
В *третьем параграфе* главы 2 рассматривается подход к синтаксическим описаниям, основанный на концепции синтаксической спецификации как описания множества допустимых объектов через множества возможных первичных объектов и параметров с использованием теоретико-множественных и функциональных операций. Для данного подхода предлагается графический структурный синтаксический язык, описания на котором имеют смысл «функциональной схемы» правил композиции, включающей граф структуры. Представленный язык отличается высокой наглядностью описаний, может использоваться как для описаний при информационном общении «человек-человек», так и при организации человеко-компьютерного интерфейса в компьютерных системах с элементами искусственного интеллекта (например, для описания языков программирования).

Синтаксический язык π -схем реализован в комплексе программных средств ПΥΘΑΓΩΡΑΣ-S (Фиг. 8), входящем в состав среды визуального программирования ПΥΘΑΓΩΡΑΣ. Реализован расширенный язык, позволяющий описывать как контекстно свободные, так и контекстно зависимые грамматики. Среда позволяет ввести набор синтаксических описаний для некоторого набора текстовых объектов (например, для понятий языка программирования) в виде синтаксических π -схем, а затем для любого текста определять является ли он одним из синтаксически допустимых объектов, и определять его структуру.



Фиг. 8. Пример синтаксических описаний в среде ПΥΘΑΓΩΡΑΣ-S.

В *четвертом параграфе* главы 2 описан графический язык формального структурно-лингвистического описания образов для задач распознавания, синтеза образов и организации человеко-компьютерного взаимодействия в системах с элементами искусственного интеллекта. Множество допускаемых объектов (образов) при этом описывается через множества возможных первичных объектов и параметров с использованием теоретико-множественных и функциональных операций (Фиг. 9). При этом как множество первичных объектов, так и операции их композиции и результирующее множество образов из описываемого класса рассматриваются как нечеткие в широком смысле (скажем, имеющие статистико-вероятностную или субъективно-вероятностную природу).



Фиг. 9. Пример описания класса сигналов на графическом языке описания образов

В пятом и шестом параграфе главы 2 обсуждаются некоторые новые подходы к визуализации сложных моделей.

В пятом параграфе главы 2 описана технология адаптируемого под пользователя базового визуального языка средствами того же базового языка – «язык на языке в языке». Эта технология позволяет создать средствами базового языка программирования (и программных средств его поддержки) языковые средства, ориентированные на конкретную область приложений (например, на работу с мультимикропроцессором МВС с одной стороны, на работу с матричной алгеброй – с другой, и т.п.). Подобные расширения в интерфейсном плане воспринимаются пользователем как элементы нового языка, однако эти средства расширяют базовый язык; в процессе работы развитие моделей может сопровождаться параллельным расширением языка. Данная технология позволяет упростить работу специалиста за счет адаптивного приближения возможностей среды к предметной области.

В шестом параграфе главы 2 описана разработанная, как развитие концепции объектно-ориентированного программирования, концепция интеграции объектно-ориентированных и процедурных компонент с файловыми – концепция наделения элементов программы (например, переменных, заголовков циклов, заголовков классов, методов, программы в целом и т.п.) внеязыковыми файловыми контейнерами, доступными как из редактора программы, так и из самой программы при ее исполнении. В рамках данного подхода возможна, в частности, реализация самодокументирующейся программы, различные элементы которой содержат «внутри себя» их описания-подсказки (в виде, например, html-страниц или звуковых речевых файлов), что может существенно облегчить разработку программы в условиях совместной работы программиста и прикладного специалиста-непрограммиста. Механизм удобен для реализации представлений данных в программе в естественном виде (например, матриц в виде матриц, функциональных зависимостей в ви-

де рисуемых графиков и т.п.) Возможность взаимодействия исполняемой программы с редактором программы позволяет создавать эволюционирующие программы.

Третья глава работы посвящена проблеме интеграции удаленных вычислителей в гетерогенную систему, основанную на объектной компонентной модели человеко-компьютерного взаимодействия и взаимодействия программно-аппаратных средств.

В *первом параграфе* главы 3 рассматриваются имеющиеся проблемы для построения гетерогенных систем с применением удаленных вычислений.

Во *втором параграфе* главы 3 рассматривается предлагаемая технология синтеза полиязыковых многомашинных программных систем и интеграции удаленных вычислительных ресурсов в гетерогенной системе на базе «вычислительных посредников».

В данном подходе пара <пользовательский компьютер – вычислитель> разбивается на две пары: <пользовательский компьютер – ВПС> и <ВПС – вычислитель>. Основными элементами этого подхода являются:

- концепция «вычислительной услуги», как ***триады***: (а) вычислительная система, (б) программные средства, (в) информационное сопровождение,
- концепция «вычислительного прокси-сервера» ВПС, предоставляющего некоторое множество вычислительных услуг, каждая из которых может реализовываться с помощью каких-то из доступных вычислительных систем и снабжается необходимым информационным сопровождением для использования вышележащим слоем компонентной системы в рамках π -технологии мультязыковых описаний и распределенной обработки.

При этом «вычислительная услуга» может использоваться слоем компонентной системы как «вычислительный атом». В соответствии с компонентной технологией должны обеспечиваться средства для связывания «вычислительной услуги» с объемлющим слоем-контейнером по крайней мере на фазах создания-редактирования и исполнения. Связь на фазе редактирования спецификации модели обеспечивается за счет «информационного сопровождения», включенного в понятие «вычислительной услуги», обеспечивающего навигацию по ресурсам из редактора объемлющего языка, а также регистрацию ресурса в контейнере для возможности последующего вызова на этапе исполнения.

Основные задачи ВПС таковы:

- ведение «справочного бюро» по функциям, предоставляемым клиенту,
- предоставление средств для «принятия заказов» на вычисления с передачей данных по протоколам CORBA и COM, с предоставлением от сервера данных для регистрации ресурса у клиента через веб-доступ,
- наиболее целесообразном выборе конкретного вычислительного сервера (или серверов при многомашинной работе) для проведения конкретных вычислений (например, по стоимости вычислений),
- активация задания на вычислителе
- передача данных,
- контроль хода вычислений,
- принятие решений о смене вычислителя при его отказе либо обнаружившихся плохих параметрах по производительности (динамическая реконфигурация),
- решение задачи накопления опыта по надежности вычислителей,
- регистрацию ресурсов, предоставляемых вычислительными серверами (ВС), и регистрацию программ для ВС для хранения в библиотеке ВПС,
- регистрацию пользователей и коммерческий учет потраченных ресурсов.

Использование системы-посредника позволяет решить проблемы по унификации способов обращения к вычислительной услуге независимо от особенностей конкретного вычислителя, на котором выполняется обработка задания, позволяет до некоторой степени избавить конечного пользователя от необходимости быть в курсе новостей в области вычислительных серверов и обновлений версий программных средств, решающих интересующие его задачи.

Одно из достоинств данного подхода – то, что выполнение лицензированной программы, хранящейся на удаленном вычислительном сервере, делает для обратившегося к этой программе невозможным извлечение какой-либо информации об этой программе, допускающей изготовление ее несанкционированной копии.

Наконец, технология позволяет наряду с библиотеками научных публикаций, библиотеками алгоритмов и программ создавать «библиотеки» вычислительных «агентов» (триад <«ноу хау», вычислительная система, информационное сопровождение>) для использования их как вычислительных атомов (со скрытыми деталями реализации) в гетерогенных системах.

Таким образом, технология направлена на поддержку индустрии и рынка вычислительных ресурсов с включением суперкомпьютеров в Интернет-технологии с целью массового использования их в науке и промышленности, с созданием отраслевых и национальных банков вычислительных услуг.

В третьем параграфе главы 3 рассмотрен ряд математических моделей, описывающих взаимодействие вычислительного прокси-сервера с клиентом и вычислительной средой. Исследована задача оптимального планирования загрузки вычислителей, мониторинга вычислений и расчета стоимости ресурса вычислительных серверов в рамках данных моделей взаимодействия.

Четвертая глава работы посвящена применению мультязыковой компонентной технологии и визуального подхода к построению конкретных сложных систем, являющихся компонентами программного комплекса для гетерогенной обработки.

В первом параграфе главы 4 описываются программные средства, реализующие полученные результаты, которые можно условно разделить на следующие взаимосвязанные группы:

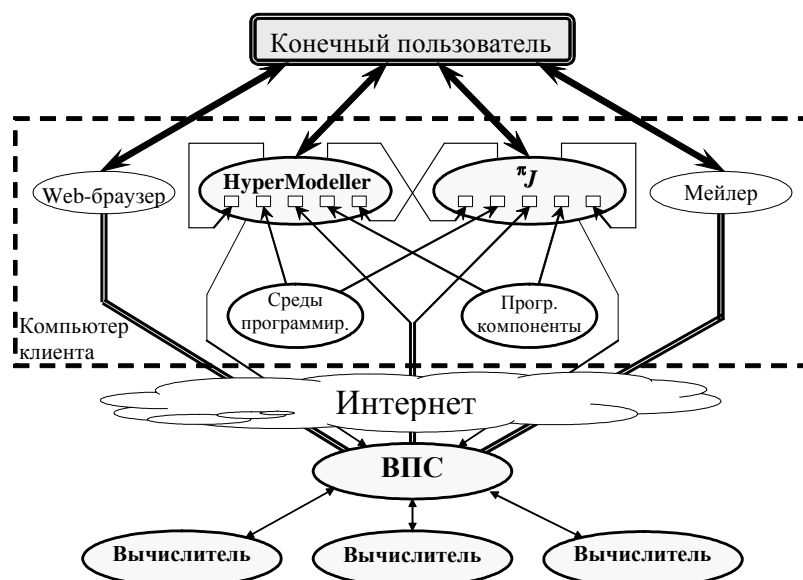
(а) комплекс решений по средам визуального процедурного программирования, включая технологию мультязыкового компонентного программирования с использованием удаленных вычислительных ресурсов, разработка методов адаптации языка к области задач пользователя; реализация результатов исследований в комплексе визуального процедурного программирования ⁷;

(б) комплекса решений для «вычислительного менеджера» – рабочего места прикладного специалиста, базирующегося на объектно-ориентированной визуальной сборке вычислительных и интерфейсных элементов; реализация результатов исследований в программной объектно-ориентированной среде HyperModeller; разработка средств визуального мониторинга и управления вычислительной сетью;

(в) вычислительный прокси-сервер;

(г) комплекс решений по интернет-видеотехнологиям.

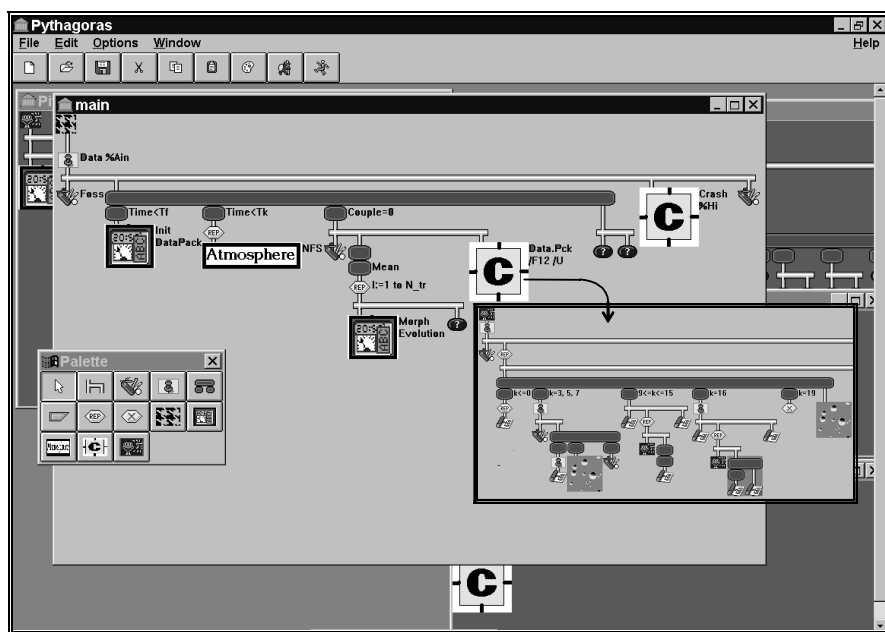
Возможное соотношение основных элементов разработанного комплекса при построении сложной многокомпонентной системы показано на Фиг. 10 (не изображенные средства видеотехнологий могут обеспечивать передачу визуальных данных от ВПС к пользователю).



Фиг. 10. Взаимодействие частей программного комплекса

Комплекс разработанных программных средства позволяет создавать на базе разработанных технологий сложные системы с использованием различных аппаратных средств (персональных компьютеров и ресурсов суперкомпьютеров), программных средств (графических языков и произвольных традиционных сред программирования и приложений) и технологических решений (напр., создания результирующих вычислительных сред в виде веб-страниц с удаленным доступом и др.). Одной из основных задач являлось обеспечение доступности элементов разработанного комплекса для работы прикладного специалиста, далекого от программирования, минимизация затрат на освоение систем.

Во *втором параграфе* главы 4 описывается среды визуального алгоритмического программирования ПΥΘΑΓΩΡΑΣ-Т (1991), ПΥΘΑΓΩΡΑΣ для MS DOS (1993), ПΥΘΑΓΩΡΑΣ для Windows (1995), позволяющие разрабатывать программные средства (например, системы моделирования) в соответствии с компонентной π -технологией, создавая мультимедийные многомашинные комплексы. При этом в качестве верхнего слоя языковой иерархии во всех системах мог применяться графический язык алгоритмических структурных π -схем.



Фиг. 11. Языковой слой системы ПΥΘΑΓΩΡΑΣ

Основными компонентами этих систем являются наборы {редактор, конвертор, компилятор, интерпретатор, ...} для алгоритмического, синтаксического и формульного подязыков с возможностью самостоятельно определять (в виде аналогичных триад) произвольные другие языковые слои, подключать произвольные программные средства. При этом редакторы комплекса обеспечивают запуск специфицированных «внешних» редакторов для каждого из определенных подязыков, компилятор – запуск специфицированных «внешних» компиляторов при прохождении процесса компиляции через подязыковые «вставки», а также обеспечивается запуск специфицированных интерпретаторов в соответствии с описанием при запуске программы в исполнение.

На Фиг. 11 приведен скриншот одной из подсистем комплекса ПΥΘΑΓΩΡΑΣ, где в качестве подчиненных слоев языковой иерархии использованы язык Си и графический вариант языка Си для программирования многопроцессорного вычислителя МВС.

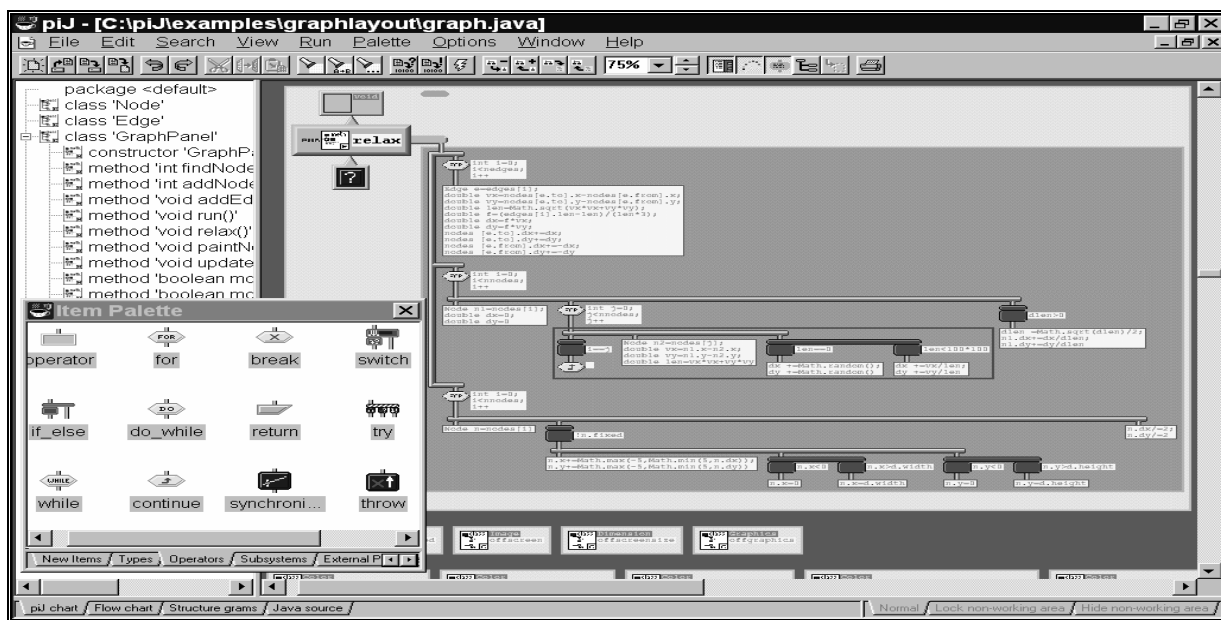
В *третьем параграфе* главы 4 описывается среда πJ визуального программирования на языке Java, базирующаяся на концепциях объектно-ориентированного программирования. При этом под «визуальностью» понимается как возможность работы с элементами графического интерфейса программы в их естественном визуальном представлении, так и возможность работать с алгоритмами в программе, представленными графически (на одном из 3-х графических языков со свободным выбором пользователем желаемого вида отображения для любой части программы).

Для сравнения в таблице приведены некоторые характеристики некоторых из распространенных сред визуального программирования и среды πJ :

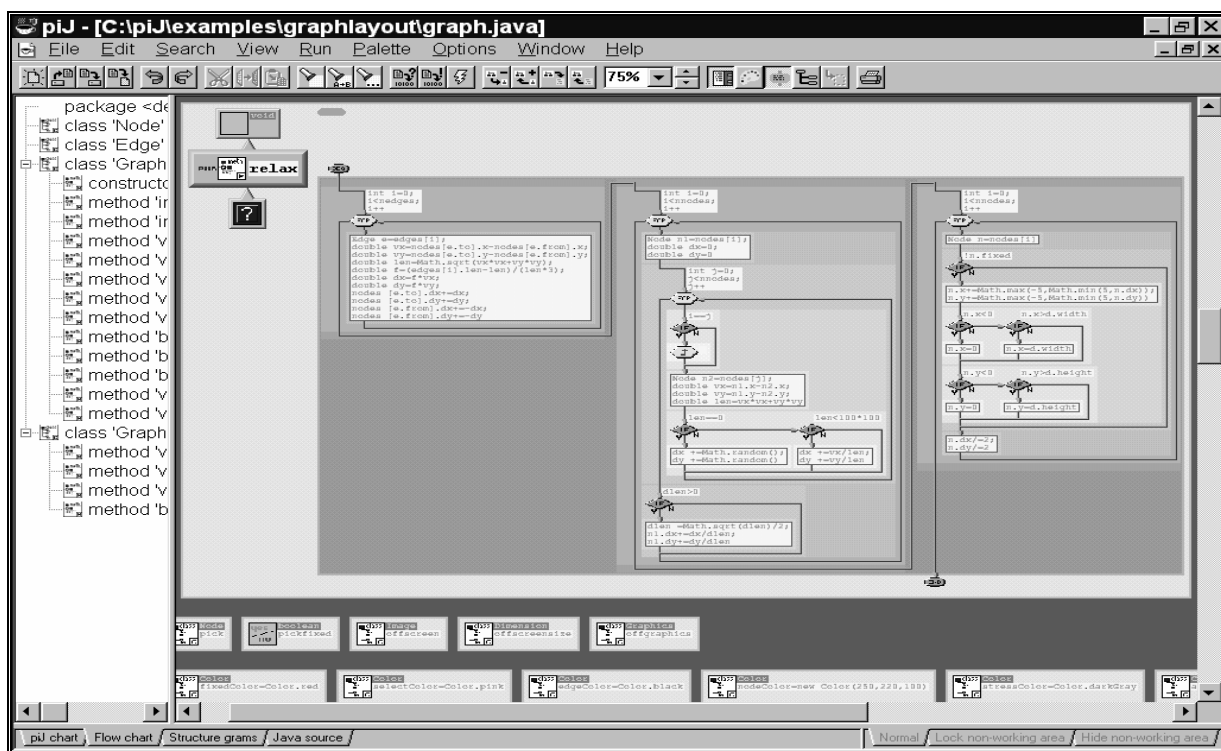
Характеристика		IBM Visual Age	MS Visual J++	πJ
Поддерживаемый язык	Java	✓	✓	✓
	JavaScript	—	—	✓!
Графическое представление структуры программы:	уровень пакетов	✓	—	—
	уровень классов	✓	✓	✓
	уровень методов и переменных-полей	✓	✓	✓
	уровень операторов (алгоритмов)	—	—	✓!
Возможность редактирования программы с графическим представлением структуры	уровень пакетов	✓	—	—
	уровень классов	✓	✓	✓
	уровень методов и переменных-полей	—	—	✓!
	уровень операторов (алгоритмов)	—	—	✓!
Отладка с использованием графического интерфейса		✓	✓	✓
Представление структуры наследования классов		✓	✓	✓
Импорт визуальных ресурсов		—	✓	—
Визуальное редактирование интерфейсных элементов		✓	✓	✓
Возможность адаптации пользовательского языка программирования под конкретную область приложений		—	—	✓!
Возможность подключения ресурсов суперкомпьютеров и внешних языков по иерархической технологии		—	—	✓!

Разработаны 2 основные версии среды πJ : (а) среда программирования на Java (функционирует как автономная система и как компонент среды HyperModeller), (б) добавление к MS FrontPage, для разработки фрагментов на JavaScript. Поддерживается программиро-

вание в графических версиях языков Java и JavaScript в виде структурных π -схем алгоритмов (Фиг. 12), структурных граф-схем алгоритмов (Фиг. 13) и структурограмм Несси-Шнейдермана. π включает инструментальные средства для конвертирования текста на стандартном языке Java в схему любого из форматов. Среда позволяет пользователю изменять текстовое представление любой структурной части программы на графическое и обратно. Разработан структурированный язык граф-схем алгоритмов схем с цветовой визуализацией структуры (Фиг. 13).



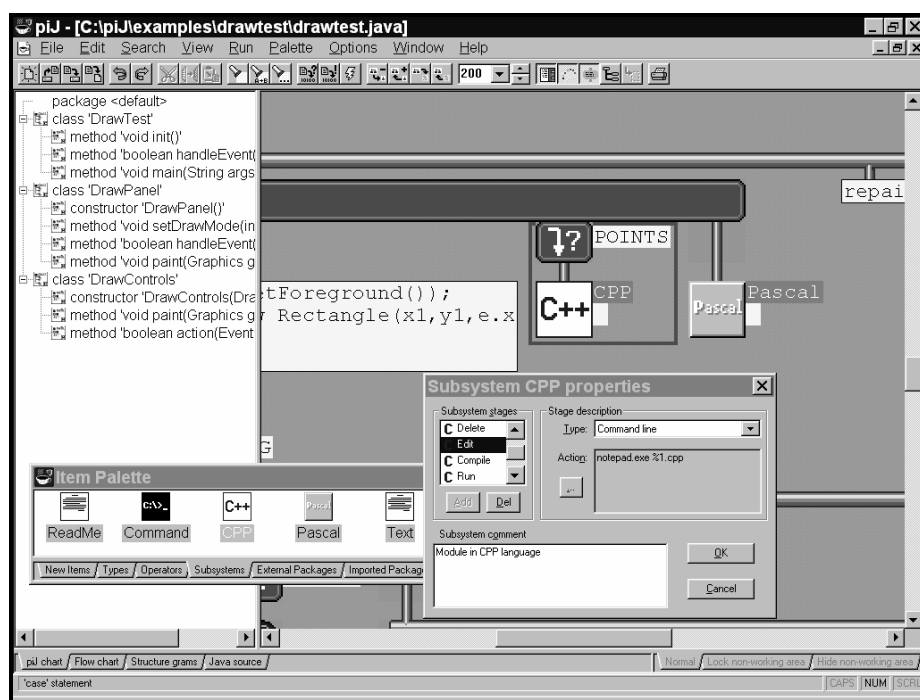
Фиг. 12. Отображение в режиме π -схем алгоритмов



Фиг. 13. Отображение в режиме граф-схем алгоритмов

В среде реализованы технология адаптируемого под пользователя базового языка средствами того же базового языка, концепция интеграции объектно-ориентированных и процедурных компонент с файловыми (описанные в параграфах 5 и 6 главы 2), технология

интеграции в языке ресурсов на произвольных других языках и ресурсов удаленных суперкомпьютеров и др. (Фиг. 14).



Фиг. 14. Настройка подсистемы удаленных вычислений

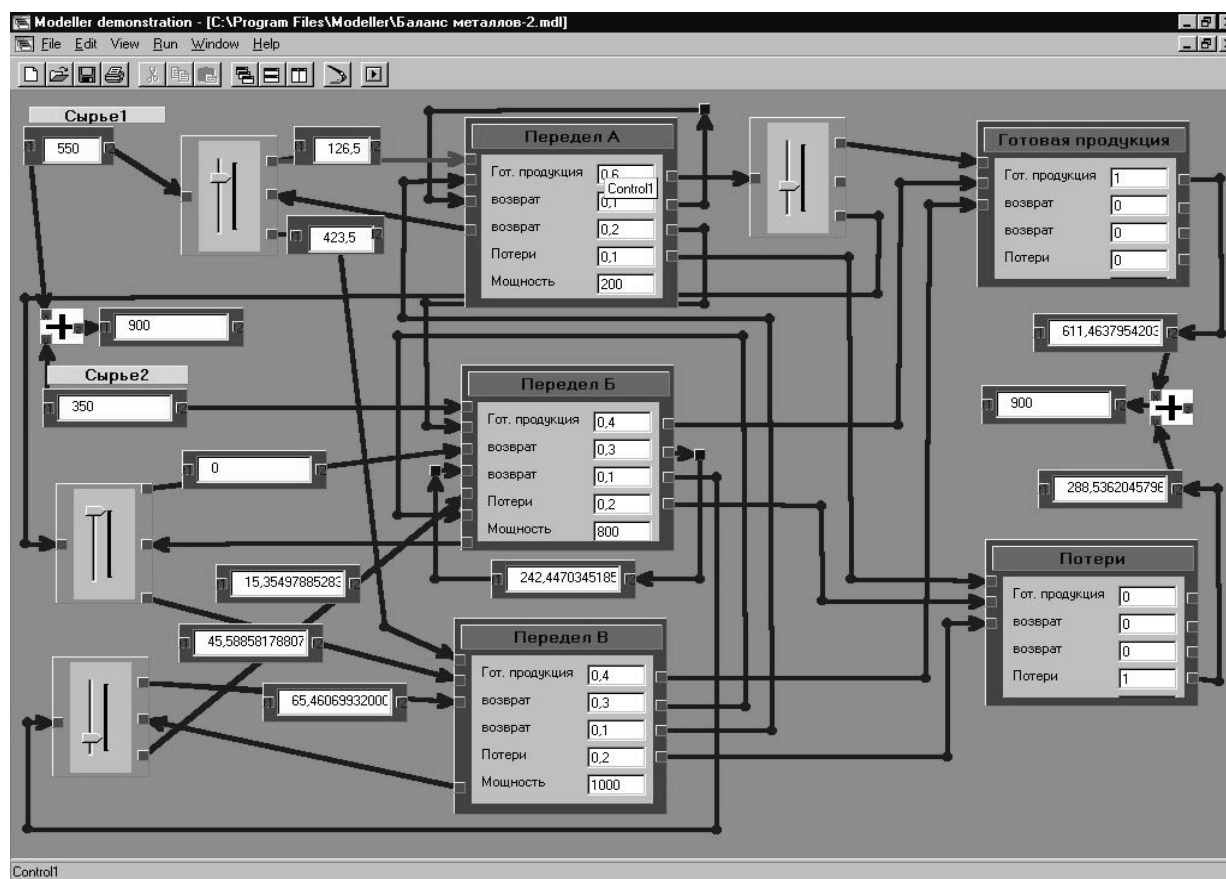
В четвертом параграфе главы 4 описывается программная среда HyperModeller, обеспечивающая рабочее место прикладного специалиста-«строителя компьютерных моделей».

Развита архитектура интегрирующего приложения, обеспечивающего полный цикл разработки распределенных приложений на основе объектно-ориентированной визуальной сборки вычислительных и интерфейсных элементов. При этом сборочными элементами могут быть модули, заданные на произвольных языках программирования и/или разрабатываемые в произвольных современных средах, а также суперкомпьютерные ресурсы, доступные через вычислительный прокси-сервер. Модули могут быть как объектами в смысле ООП, так и обычными процедурными программами. Определены следующие виды модулей: COM-объект, ActiveX-элемент управления, консольный модуль (приложение DOS или Win32, при генерации которого использованы старые компиляторы, умеющие создавать только DOS-приложения), Java-объект (может быть создан с использованием визуальной среды разработки Java-приложений πJ), модуль обращения к вычислительным услугам ВПС, а также произвольные программные средства, образующие «подсистемы» в смысле π -технологии. Элементы могут соединяться друг с другом посредством информационных (служащих для передачи данных) и управляющих (служащих для управления процессом передачи данных) связей.

Данная технология вместе с π -технологией и технологией комплекса πJ обеспечивает полный цикл чисто визуальной разработки распределенного программного обеспечения с использованием ВПС, в том числе – с применением языка Java. Особенностью среды HyperModeller является возможность совмещения фаз редактирования и исполнения, то есть, спецификация программы может меняться «на ходу», без ее останова.

Среда HyperModeller может применяться как визуальная среда разработки сложных иерархических динамических систем, систем управления и обработки информации. Она ориентирована на непрограммирующего профессионала и обладает интуитивно понятным, дружественным интерфейсом пользователя. Используя данную среду моделирова-

ния, можно получить интерактивный доступ к любой точке схемы, оперативно вмешаться в ход расчетов, изменить параметры динамической системы и др.

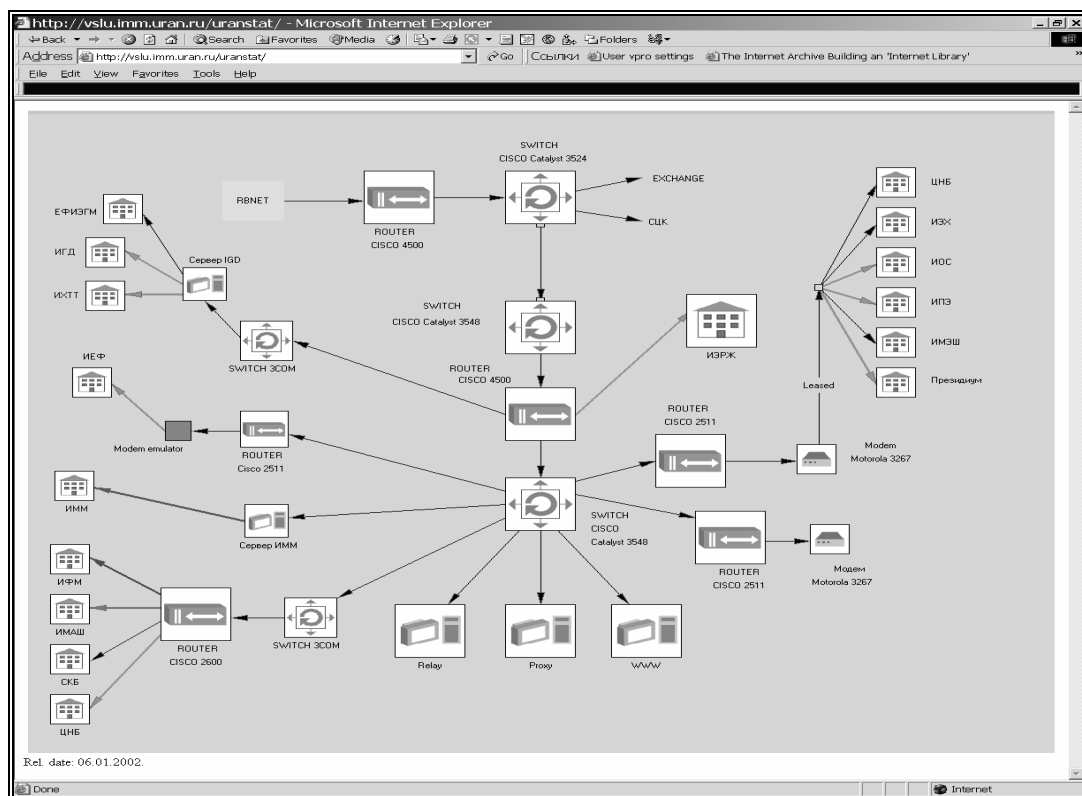


Фиг. 15. Модуль разработанной в среде HyperModeller модели баланса металлов на медеплавильном предприятии

Скажем, в экономических системах объектная модель, заложенная внутри данного программного комплекса, позволяет быстро разрабатывать и внедрять новые объекты, которые несут информацию не только о денежных и сырьевых потоках, но и о производственных процессах и их взаимосвязи. Объединение в единую систему всех процессов предприятия позволяет провести производственно-экономическое моделирование и стратегическое планирование деятельности. С помощью данных программных средств можно достаточно просто создать модель финансовой структуры предприятия, модель структуры управления предприятием или модель производственного процесса.

На Фиг. 15 показан пример компонента гетерогенной системы принятия управленческих решений, разработанной на базе сред HyperModeller и πJ (пример подсистемы баланса металлов). «Атомарные элементы» показанной схемы HyperModeller – на самом деле являются сложными модулями отображения, обрабатывающими модулями и ресурсами данных, части из которых могут соответствовать удаленные компьютеры. Эти модули реализованы в других подсредах, вход в работу по редактированию которых осуществляется двойным щелчком мыши на соответствующем элементе.

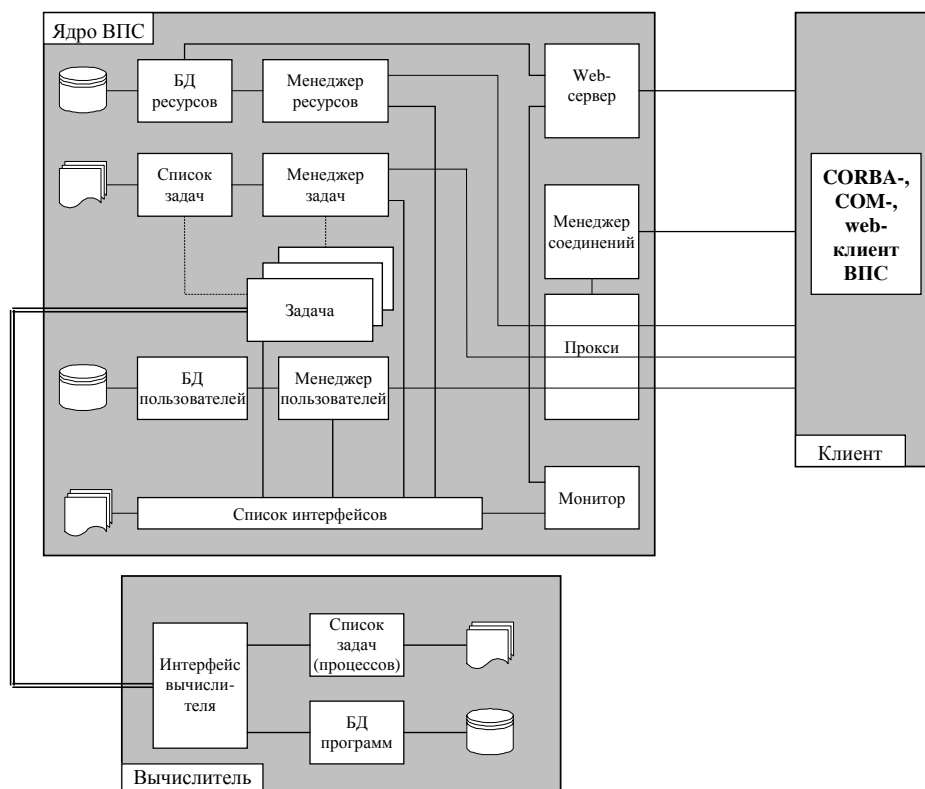
На Фиг. 16 показан пример системы администрирования и настройки вычислительной сети и предоставления информации об ее работе, разработанной на базе сред HyperModeller и πJ (<http://vpro.imm.uran.ru/uranstat>). Данная система также является гетерогенной мультязыковой многомашинной средой – за «атомарными элементами» скрываются программные средства, функционирующие на удаленных компьютерах, переход на работу с которыми возможен по двойному щелчку мыши на элементе.



Фиг. 16. Система администрирования и настройки вычислительной сети в системе HyperModeller

В *пятом параграфе* главы 4 описывается система «Вычислительный прокси-сервер», реализующая технологию, описанную в главе 3 диссертации.

Основные компоненты программных средств ВПС и их взаимосвязь показаны на Фиг. 17.



Фиг. 17. Структура программного комплекса ВПС

В ИММ УрО РАН была запущена в опытную эксплуатацию экспериментальная версия вычислительного портала под ОС MS Windows NT. Данный вычислительный портал поддерживает доступ к вычислительным услугам из приложений по стандартам Corba и COM, а также информационную интеграцию через веб-доступ. Разработаны соответствующие программные средства и осуществлено экспериментальное подключение к ВПС вычислителей МВС ИММ УрО РАН.

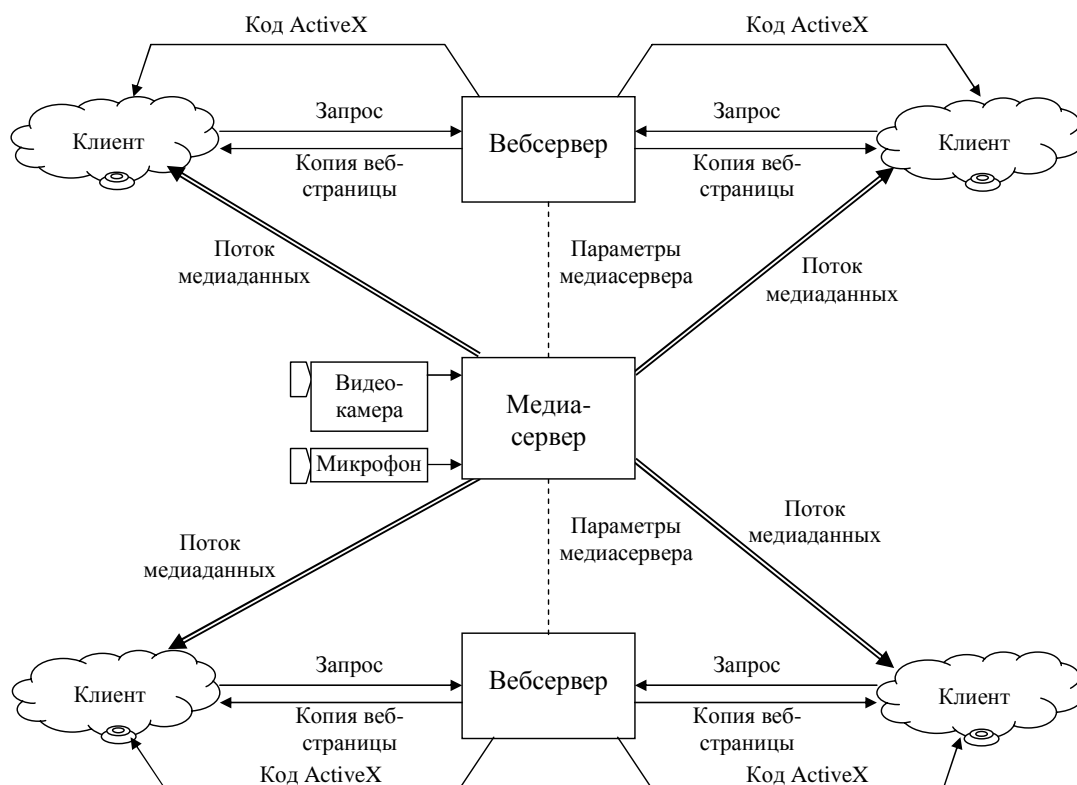
Вычислительные услуги ВПС могут использоваться из средств программирования с применением их как вычислительных «атомов». При этом могут использоваться, например, среды визуального проектирования *HyperModeller* (функциональный конструктор) и πJ (алгоритмический конструктор). Услуги ВПС могут использоваться через протокол CORBA или COM и из программ пользователя, разработанных в других средах, при этом информационная интеграция в среду разработки верхнего уровня (включая навигацию, документирование, передачу шаблонов вызова) осуществляется с помощью веб-доступа.

В шестом параграфе главы 4 рассматриваются технологии передачи медиаданных, связанные как с разными сервисами межчеловеческого общения, так и с применением удаленных суперкомпьютерных вычислений, базирующиеся на применении компонентной π -технологии создания сложных программных систем. Описаны приложения для аудиовизуальной связи через компьютерную сеть, входящие в комплекс *ViDoors*:

- «*Веб-камера*» для наблюдения через Интернет в реальном времени, например, докладов, выступлений, общественных событий. Обеспечивает одностороннюю передачу аудиовизуальных данных в реальном времени от одного передающего компьютера множеству клиентских компьютеров зрителей. Вариант программных средств может быть использован для односторонней передачи для удаленного наблюдения визуальных результатов суперкомпьютерных вычислений.
- «*Медиаконференция*» для проведения через Интернет, например, научных семинаров и совещаний. Обеспечивает двунаправленную и многонаправленную аудио-видео связь в реальном времени между многочисленными участниками. Вариант программных средств предназначен для осуществления только аудиосвязи.
- «*Медиа по запросу*» для создания в цифровой форме, например, научных докладов и лекций, которые в дальнейшем можно будет просматривать через Интернет. Обеспечивает после однократного создания аудио/видео материала его передачу и воспроизведение на удаленных компьютерах по запросу пользователей в любое время.
- «*Видео по расписанию*» для просмотра заранее записанных видеоматериалов в соответствии с некоторым расписанием (составляемым вручную или автоматически).
- «*Мобильный репортерский спутниковый видеокомплекс*» – комплексная система, разработанная на базе полученных технологий. Система содержит полный комплект программных средств для поддержки технологического процесса подготовки законченного телевизионного новостного сюжета на компьютере выездной журналистской бригады. Это средства для записи отснятых видеоматериалов на жесткий диск, монтажа и формирования "плейлиста" с включениями "живой камеры", передачи видеосюжета в телекомпанию через спутниковый канал связи в соответствии с "плейлистом", включая вставки "живой камеры". Система включает также программные средства и для принимающего компьютера, формирующие из принимаемых со спутника данных видеосигнал, включаемый телецентром в "прямой эфир".

Разработанная технология организации медиасети с применением ретрансляторов и применения по возможности вещательных протоколов может позволить качественно уменьшить потоки данных при массовом вещании в сравнении с другими используемыми ныне системами. При этом нагрузка на сервер вещания не зависит от количества одновре-

менно принимающих клиентов, количество которых может составлять, скажем, миллионы.



Фиг. 18. Структура системы медиавещания комплекса ViDoors

Использование разработанной технологии с осуществлением *визуализации суперкомпьютерных вычислений* на стороне суперкомпьютера при применении вычислительного прокси-сервера позволяет осуществить более гибкий мониторинг вычислений в реальном времени, повышая производительность труда специалиста, проводящего удаленные вычисления, снизить расходы на вычислительный эксперимент и ускорить получение конечного результата.

Разработанные программные средства представлены на сервере webtv.uran.ru.



Фиг. 19. Одна из комнат сервера медиаконференций

Заключение

Для сложных систем разработан подход к организации человеко-компьютерного взаимодействия, языков представления знаний, программных и аппаратных средств как иерархии независимых «микроязыковых» элементов. Подход основан на рассмотрении программных средств/оборудования как сложной системы с применением соответствующей методологии. Предлагается декомпозировать решаемую задачу на иерархию микроконтекстов, каждый из которых может быть наилучшим образом поддержан своим собственным микроязыком (с соответствующими редакторами, компиляторами и интерпретаторами). Построен и изучен пример модели оценки и оптимизации затрат в многоуровневых компонентных средах.

Разработаны новый визуальный структурный алгоритмический язык и новый визуальный структурный язык синтаксических описаний. Разработан новый метод порождения адаптируемого под класс задач визуального языка, новые методы структурных графических представлений. Разработан новый метод интеграции фазы редактирования и фазы исполнения программы созданием файловых контейнеров, нагружающих узлы структурной программы.

Предложена новая технология предоставления клиенту ресурсов вычислительной системы для включения в метапрограмму в виде *триады* <программные средства, аппаратные средства, информационная поддержка> с возложением на посредника – «вычислительный прокси-сервер» (ВПС, «вычислительный портал») – выполнения ряда задач, связанных с решением ряда проблем предоставления вычислительной услуги клиенту. Вычислительный прокси-сервер разбивает пару <пользовательский компьютер – вычислитель> на две пары: <пользовательский компьютер – ВПС> и <ВПС – вычислитель>. При этом, кроме всего прочего, выполнение лицензированной программы, хранящейся на удаленном вычислительном сервере, делает для обратившегося к этой программе невозможным извлечение какой-либо информации об этой программе, допускающей изготовление ее несанкционированной копии.

Рассмотрен набор математических моделей взаимодействия вычислительного прокси-сервера с клиентом и вычислительной средой, исследован ряд возможных стратегий вычислительного прокси-сервера по оптимальному использованию ресурсов вычислительных серверов.

На основе общей технологии компонентного построения мультиметафорных гетерогенных систем разработан ряд частных технологий. При этом разработаны архитектуры элементов комплекса программных средств, поддерживающего разработанную компонентную технологию, включающего:

- среды визуального алгоритмического программирования – ПУΘΑΓΩΡΑΣ и ^πJ,
- среду визуальных синтаксических описаний ПУΘΑΓΩΡΑΣ-S,
- систему визуального конструирования на базе функциональных схем HyperModeller,
- систему предоставления доступа к вычислительным услугам «Вычислительный прокси-сервер»,
- комплекс средств *ViDoors* для распределенной среды интернет-медиакоммуникаций (как для межчеловеческого общения, так и для использования в составе многомашинных вычислительных систем).

Основные публикации автора по теме диссертации

- [1] Прохоров В.В. Структурный язык визуальных представлений. В кн.: Задачи моделирования и оптимизации. – Свердловск: УрО АН СССР, 1991. – С. 151-174.
- [2] Прохоров В.В. π -схемы – язык графического представления алгоритмов. // «Кибернетика и системный анализ», №2, 1992 (ISSN 1060-0396). – С.93-107.
- [3] Prokhorov, V.V. Hypermedia programs and scenarios in images // Proceedings of the International Conference on Computer Technologies in Education. Kiev, 1993. – P.78-82.
- [4] Прохоров В.В. О языке структурно-графических синтаксических описаний // Материалы IV Национальной конференции с международным участием «Искусственный интеллект» КИИ-94. – Рыбинск, 1994. – С. 118-121.
- [5] Прохоров В.В. PYTHAGORAS and piLOGO: Software on the Graph Approach and the pi-Technology Basis // Proceedings of the 19th International Conference «Information Technologies and Programming». – Sofia: Math Institute of Bulgarian Academy of Sciences, 1994. – С. 147-152.
- [6] Prokhorov, V.V. On contradictions Between Wealth and Compactness of Languages. In: M.Purvis, Ed. «Software Education Conference» SRIG-ET'94, Los Alamitos: IEEE Computer Society Press, 1994. – P.289-296.
- [7] Prokhorov, V.V. Hierarchic representations in software environment // Proceedings of 5th Hellenic Conference on Informatics, Athens, Greece, 1995. – P.57-62.
- [8] Prokhorov, V.V. PYTHAGORAS: Multienvironment Software. // Proceedings of the 5-th East-West International Conference on Human-Computer Interaction (EWHCI'95), Vol.II, ICSTI, Moscow, Russia, 1995. – С.160-172.
- [9] Prokhorov, V.V. Hierarchic Multilanguage Reuse Technology and Knowledge Engineering. In: Arbeitspapiere der GMD Gesellschaft fur Mathematik und Datenverarbeitung mbH, Vol.903. Sankt Augustin, Germany, 1995. – P. 122-126.
- [10] Prokhorov, V.V. PYTHAGORAS: Multienvironment Software. In: B.Blumenthal, J.GornostaeV, and C.Unger, Eds. Human-Computer Interaction. Lecture Notes in Computer Science (LNCS), Vol.1015. Springer Verlag, Berlin, Germany, 1995. (ISBN 3-540-60614-9). – P.135-148.
- [11] Prokhorov, V.V. To Graph Representation of Syntax Definitions. In: J.G.Chen (ed.). Expert Systems Applications & Artificial Intelligence. Technology Transfer Series. Gournay-Sur-Marne, France: PTT International, 1995 (ISBN 2-90766931-1). – P.53-58.
- [12] Prokhorov, V.V. Graph Approach and Hierarchical Technology in Representation of Knowledge and Programs. // Second International Symposium on Knowledge Acquisition, Representation, and Processing (KARP-95): Abstracts. Auburn University, Auburn, AL, USA, 1995. – P. 103-105.
- [13] Прохоров В.В. Об иерархической технологии организации программных средств и человеко-компьютерного взаимодействия. // Труды Института математики и механики УрО РАН, т.3. Екатеринбург: УрО РАН, 1995. – С.274-294.
- [14] Прохоров В.В. О мультиметафорном подходе к построению языков и программных средств. // Труды международного семинара «Искусственный интеллект в образовании», часть II. – Казань: КГТУ, 1996. С.49-54.
- [15] Prokhorov, V.V. Model of Microcontext User-Adaptable HCI for Multi-Computer Complexes. // Международная конференция «EastWest HCI'96». – Москва, ICSTI, 1996. – С.113-120.
- [16] Prokhorov, V.V. On graphic structured languages. // Proceedings International Conference on Information Visualisation IV'97, IEEE Computer Society Press, Los Alamitos, CA. 1997. – P.219-225.

- [17] Prokhorov, V.V. The PYTHAGORAS Media: Multilanguage Technology and Graphic Descriptions. In: P.Commers, A.Dovgiallo, V.Petrushin, and P.Brusilovsky, Eds. New Media and Telematic Technologies for Education in Eastern European Countries. Twente University Press, Enschede, Nederland, 1997. – P.377-385.
- [18] Прохоров В.В. О микроконтекстном подходе к построению языков представления знаний и человеко-компьютерного взаимодействия. // Известия РАН. Теория и системы управления, 1997, №5 (ISSN 1064-2307). – С.5-16.
- [19] Прохоров В.В. Технология использования вычислительных ресурсов в Интернет на основе вычислительных прокси-серверов. // В кн.: Алгоритмы и программные средства параллельных вычислений. Вып.2. – Екатеринбург: УрО РАН, 1998, с.256-267.
- [20] Прохоров В.В. Вычислительный портал: средства удаленного доступа к ресурсам суперкомпьютеров. // Труды Всероссийской научной конференции «Высокопроизводительные вычисления и их приложения» (г.Черноголовка). М.: МГУ, 2000. С.161-165.
- [21] Prokhorov, V.V. Computational Portal: Remote Access to High-Performance Computing. In: V.Malyshkin, Ed. Parallel Computing Technologies. Lectures Notes in Computer Science (LNCS). Vol.2127. Springer Verlag, Berlin, Germany, 2001. P.308-313.
- [22] Prokhorov, V.V. Computational Proxy-Servers and Visual Assembling Technology in Distributed Software Engineering. Proceedings of the 3rd International Workshop on Computer Science and Information Technologies CSIT'2001. Ufa, Yangantau, Russia, 2001. – С. 97-105.
- [23] Прохоров В.В. Об оптимальном планировании использования ресурсов вычислительным прокси-сервером // Алгоритмы и программные средства параллельных вычислений. Вып.6. – Екатеринбург: ИММ УрО РАН, 2002. С.242-288.
- [24] Прохоров В.В. Комплекс интернет-медиасредств на базе компонентной π -технологии // Алгоритмы и программные средства параллельных вычислений. Вып.6. – Екатеринбург: ИММ УрО РАН, 2002. С.289-356.
- [25] Прохоров В.В., Косарев В.А., Копелев А.Б., Смирнов Д.В. Визуальные мультязыковые средства для разработки распределенных сред. // Управляющие системы и машины. – 1999. №5. – С.75-83.
- [26] Прохоров В.В., Смирнов Д.В. Модель оценки и оптимизации затрат в многоуровневых нечетких средах. // Программирование, 2001, №5, с. 35-44.
- [27] Prokhorov, V.V.; Kosarev, V.A. Environment π J for Visual Programming in Java. // Proceedings International Conference on Information Visualisation IV99, IEEE Computer Society Press, Los Alamitos, CA, 1999. – P.535-542.
- [28] Прохоров В.В., Смирнов Д.В., Копелев А.Б., Береснева Н.П., Кучина Е.П., Яковлев А.В. ПИФАГОР. Графическая среда программирования. – Екатеринбург: УрО РАН. 1998. – 136 с.
- [29] Бердышев В.И., Козлов В.А., Кондратьев В.П., Прохоров В.В., Л.А.Хоменко. «Метод диагностики нарушений кровенаполнения поджелудочной железы». Патент России RU2098011, опубл. 10.12.1997.