

Использование сборочной технологии для построения пользовательских интерфейсов сетевой информационно-вычислительной системы¹

С.В. Пискунов, С.В. Кратов, М.Б. Остапкевич, А.В. Веселов

Институт вычислительной математики и математической геофизики СО РАН

630090 Россия, г. Новосибирск, пр. ак. Лаврентьева, 6

Реферат: Рассматривается подход к построению пользовательских интерфейсов сетевой информационно-вычислительной системы, в рамках которого сначала строится высокоуровневое описание интерфейса, его модель, а затем по ней генерируется программный код с использованием технологии сборки. Описание выполняется в виде двух согласованных онтологий: предметной области и графического представления интерфейса. Такой подход позволяет а) снизить временные затраты на разработку графического интерфейса и упрощает его последующую модификацию, потому что разработчик имеет дело только с декларативным описанием, б) использовать для сборки интерфейса как собственные разработки, так и заемные компоненты. Реализация предлагаемого подхода обеспечивает поддержку многоаспектной инновационной деятельности и позволяет членам сетевого коллектива вести инновационный проект с использованием ресурсов географически удаленных кластеров для выполнения математического моделирования отдельных звеньев инновационного проекта, не выходя за пределы своей корпоративной среды и понятий своей предметной области.

Ключевые слова: сетевая информационно-вычислительная система, сборочная технология, моделиориентированный подход, онтология, пользовательские интерфейсы

Abstract: An approach to the construction of users' interfaces is considered in the paper. Within this approach a high level interface description is done first. Then a source code is generated on its basis using an assembly technology. The description is represented by two coordinated ontologies, one of which describes a problem domain and the other represents user's

¹ Работа поддержана грантом Президиума РАН 13.6

interface of this domain. The approach makes it possible to 1) accelerate the development of user's interface and ease its further modifications because a developer deals only with a declarative descriptions, b) use both own and borrowed components for an interface construction. The implementation of this approach provides support of a multi-aspect innovation activity and permits members of a network team to develop an innovation project using resources of remote clusters to perform the mathematical modeling of individual phases of the innovative project, without going outside of their corporate environment and concepts of their problem domain.

Keywords: network informational computational system, assembly technology, model-oriented approach, ontology, users' interfaces

Введение

Для инновационной разработки типичным является следующий путь. Для апробации потенциально перспективной («технологичной») идеи формируется техническое задание, создается коллектив, например временная лаборатория, организуется научно-исследовательская и опытно-конструкторская работа (НИОКР). Представляется целесообразной информационная поддержка такой цепочки на современном уровне с использованием пространства сети Интернет. В этом случае возможно существенное расширение как административных, так и географических рамок временного коллектива, привлечение как зарубежных инвесторов, так и зарубежных ученых, экспертов и т.п., обеспечение высокой скорости информационных обменов между участниками коллектива, широкий доступ к важной для тематики НИОКР информации.

Общеизвестно, что математическое моделирование является важной составляющей инновационного процесса, обеспечивающей успешную и быструю реализацию инновационного проекта. Супер-ЭВМ превратились в действенный инструмент математического моделирования как научных, так и прикладных задач больших размеров. Потребность в высокопроизводительных вычислениях постоянно растет, расширяется область применения супер-ЭВМ, вовлекаются все новые и новые пользователи, занимающиеся исследованиями в самых разных предметных областях. Для эффективного использования супер-ЭВМ в инновационной деятельности необходимо создание сетевой информационной инфраструктуры, включающей не только системные и прикладные программные средства, обеспечивающие математическое моделирование на супер-ЭВМ, но также программные средства, обеспечивающие интерактивное общение разработчиков

инновационного проекта, быстрый доступ к огромным объемам общей и профессиональной информации и т.д. Поэтому актуальной является задача создания сетевой системы, содействующей извлечению из фундаментальных научных исследований и передаче в наукоемкую промышленность «технологичных» идей посредством активной поддержки решения сложных задач НИОКР созданными в сети Интернет (виртуальными) коллективами.

В работах [1, 2, 3] сформированы требования к сетевой информационно-вычислительной системе (СИВС), описаны ее архитектура, назначение и вариант реализации. Основу системы составляет ядро, состоящее из библиотеки Динамически Конфигурируемая Модульная Система (ДКМС) [4] и набора модулей, образующего каркас системы (см. рис. 1). Этот набор реализует минимальный набор функций, который требуется для использования системы: унифицированный доступ к наборам функций ОС (для файлового ввода/вывода, сетевых коммуникаций, работы с потоками, процессами, разделяемыми библиотеками и виртуальной памятью), поддержка различных структур данных, учет и авторизация пользователей, обеспечение безопасности данных, возможность для ввода, редактирования, поиска и просмотра разнотипных данных в системе через браузер. С помощью технологии сборки над ядром надстраиваются модули расширения. Были разработаны модули, обеспечивающие формирование информационного пространства системы, взаимодействие пользователей, организацию виртуального коллектива пользователей для совместного выполнения проекта. Информационное пространство представлено репозиторием, хранящим пользовательские фрагменты данных, называемые карточками, и классификаторами, позволяющими упорядочивать множество карточек по разнообразным рубрикам. Формирование системы выполнялось силами ее разработчиков. Вместе с тем сборочная технология, шаги построения которой представлены в [5, 6, 7] обеспечивает возможность пополнения системы модулями, созданными не только разработчиками системы, но и ее пользователями.

В данной работе основное внимание уделяется развитию системы как продукта со свободным доступом. Под таким развитием, в первую очередь, подразумевается равноправное участие и разработчиков, и пользователей в развитии системы за счет формирования внешних по отношению к ядру специализированных библиотек из модулей или завершенных приложений.

Цель работы – представить комплексную технологию развития системы, которая дает как разработчику, так и пользователю возможность расширения ее функциональности на всех уровнях модульной структуры системы и, в первую очередь,

Web и GUI интерфейсов, что может существенно расширить круг пользователей СИВС, а, значит, сделать ее развитие более динамичным и многоплановым.

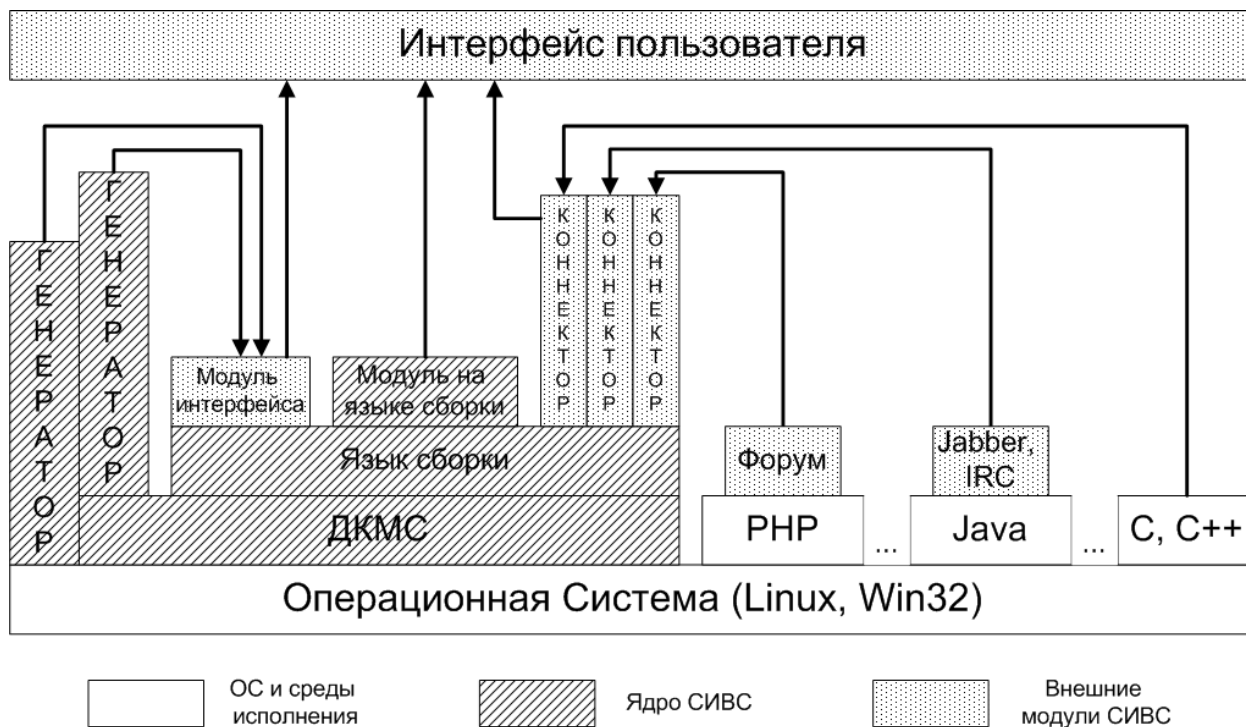


Рисунок 1. Модульная архитектура СИВС

Выбор пути развития СИВС

Все современные системы строятся как модульные с открытой архитектурой, т.е. такие, которые обеспечивают переносимость на различные платформы, масштабируемость, интероперабельность (кооперативную работу с другими открытыми системами) и дружелюбность пользовательского интерфейса. Есть два пути их построения как программного продукта.

Первый путь. Сложную систему создает подходящий по численному составу и квалификации коллектив разработчиков всю: от начала и до конца. То, что система проектируется одним коллективом и при этом используется открытая архитектура, обеспечивает повторную используемость модулей, облегчают возможность тестирования, отладки, сопровождения, логическую целостность системы с точки зрения пользователя и возможность ее модернизации.

Второй путь. Небольшой коллектив создает только ядро системы и делает доступной технологию и инструменты построения ее модулей. Ядро содержит необходимые инфраструктурные функции для наполнения системы модулями.

Ограниченный набор стандартных и часто используемых модулей строится самими разработчиками системы. Помимо полезного набора функций эти модули представляют пример того, как подобные модули могут разрабатываться пользователями, которые нуждаются в некоторой функции и хотят реализовать ее в рамках подобной системы. Если система оказывается привлекательной для достаточного числа пользователей, ее набор функций начинает пополняться такими пользователями, без усилий со стороны разработчиков. Это, в свою очередь может привлечь еще больше пользователей. Для удачных систем возникает эффект снежного кома. Именно этот путь выбран разработчиками для дальнейшего развития СИВС.

Чтобы развитие системы было успешным, мы считаем, что:

1. Пользователи должны быть допущены ко всем уровням развития модульной структуры системы.
2. При разработке модулей пользователям должен быть доступен широкий арсенал языковых и инструментальных средств: собственный язык сборки и библиотека ДКМС, общепринятые языки (C, C++, Java, Perl, PHP), такие средства, как Protégé.
3. Допускается участие в развитии системы пользователей самой разной квалификации от программистов, до экспертов в конкретной предметной области, которые могут только точно ответить на вопрос, что модуль должен делать.

Дальнейшее изложение содержит развитие сформулированных выше положений.

Описание комплексной технологии развития модульной архитектуры СИВС

Фундамент сборочной технологии, обеспечивающей развитие СИВС как системы с открытым доступом, составляют три варианта (и их комбинации) построения внешних модулей. Схема технологии и варианты ее применения пользователями представлены на Рис.2.

Первый вариант полностью основан на библиотеке ДКМС. Она является надстройкой над операционной системой (Win32, Linux) и состоит из процедур. Все модули, построенные на ее основе, содержат только обработчики событий. Все межмодульные взаимодействия происходят на основе событий. Событийный интерфейс обеспечивает возможность как встраивания новых, так и замещения старых модулей без необходимости модификации исходных текстов ядра. Первый вариант содержит два пути построения модулей, обозначенные на рис. 2 цифрами 1 и 2. Модули, строящиеся на

основе ДКМС, представлены динамически компокуемыми библиотеками, написанными на C/C++ (модуль В на рис. 2). Модули, написанные на языке сборки (на рисунке используется сокращение ЯС), представлены скриптами (модуль А на рис. 2). Добавление новых модулей может производиться любым пользователем, знакомым с языками C/C++ или языком сборки ДКМС. Модули могут быть как прикладными, так и системными. Например, путь А может быть использован для решения следующей задачи. Некоторый пользователь, применяющий приложение KOrganizer для хранения контактов, хочет обеспечить возможность их переноса в информационное пространство СИВС. Для этого он строит на языке С модуль поддержки формата данных KOrganizer для хранения контактной информации, который состоит из обработчика события, возникающего в момент, когда происходит считывание объекта данных в соответствующем формате. При этом событийный интерфейс ДКМС позволяет интегрировать данный модуль в СИВС без какой либо модификации её модулей за пределами этого нового модуля. Примеры системных модулей, которые добавляются к ИВС - поддержка сетевого протокола для взаимодействия с удаленным сервером, стыковка с языковой средой Java для возможности исполнять в ИВС процедуры, написанные на Java, реализация генератора текстов фрагментов кода на VBScript для улучшения интерактивности и дизайна Web страниц, передаваемых клиенту с MS Internet Explorer, и ряд других.

Второй вариант - присоединение к ИВС заемных модулей (форум, Jabber сервер, Wiki и др.). Этот вариант также содержит два пути (см. рис 2., цифры 3, 4). И в том, и в другом случае правка исходных текстов заемных модулей не требуется. Рассмотрим третий путь. Разработчики ИВС реализовали несколько вариантов модуля стыковки и включили их в ядро. Модуль стыковки делает видимыми интерфейсные функции заемного модуля в программной среде ДКМС. Структура модуля стыковки типична, т.е. для присоединения нового заемного модуля пользователь может взять имеющийся модуль стыковки в качестве шаблона и доработать его. Четвертый путь основан на использовании существующих решений в области свободного ПО, интегрируемых в систему без каких-либо модификаций. Подобные решения в дальнейшем используются в системе в качестве различных сервисов, например, для осуществления синхронных и асинхронных коммуникаций между пользователями системы (Jabber, IRC, форумы). Для пользователей они представляются отдельными элементами Web-интерфейса и GUI.

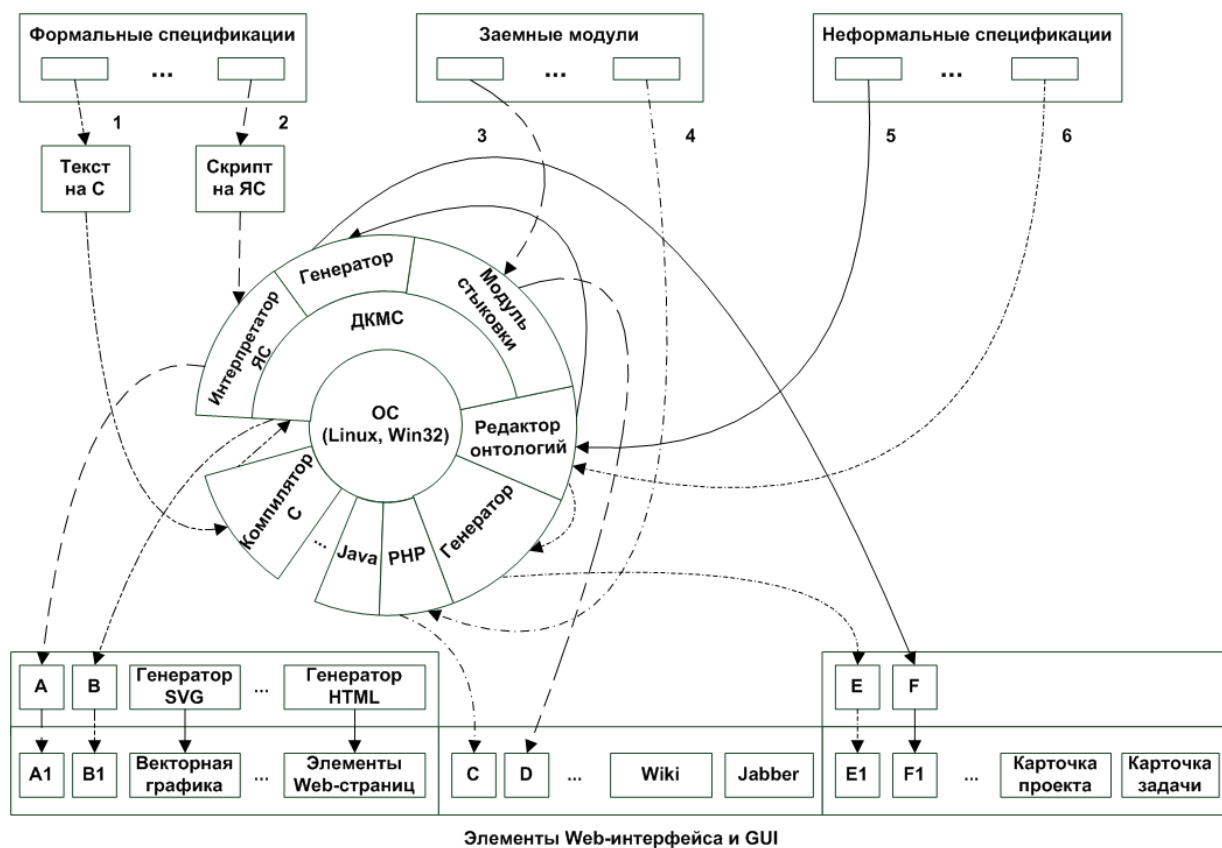


Рисунок 2. Схема сборочной технологии

Рассмотренные выше варианты построения модулей и сборки из них приложений можно назвать «классическими». Они укладываются в наиболее распространенный путь разработки приложений, который включает создание спецификации, проектирование всех используемых структур данных и кодирование на одном из языков программирования. После завершения этапа разработки приложение готово к использованию, но оно «умеет работать» только с predetermined набором объектов. Поэтому при добавлении новых объектов требуется модификация программного кода и перекомпиляция приложения. Существующие объекты можно изменить, используя их описания в виде шаблонов. Но в любом случае требуется помощь программиста.

Между тем, проблема заключается в том, что при разработке СИВС, с одной стороны, просто невозможно заранее описать все объекты инновационной деятельности во всех её сферах, с другой - желательно не потерять потенциальных пользователей системы, желающих адаптировать СИВС к своей предметной области, но не являющихся квалифицированными программистами. Поэтому представляется очевидным, что необходим еще один вариант построения модулей, дополняющий представленные выше, и такой, которым может пользоваться не только программист.

Этот (третий) вариант базируется на моделиориентированном подходе [8, 9] к построению приложений, он ориентирован на то, чтобы описать «что делать», а не «как делать». Суть подхода состоит в следующем: строится высокоуровневое описание интерфейса - его модель, а затем по ней генерируется программный код с использованием технологии сборки. Такой подход позволяет снизить временные затраты на разработку пользовательского интерфейса и упрощает его последующую модификацию, потому что пользователь имеет дело только с декларативным описанием и при изменении этого описания не требуется перепрограммирования пользователем модулей приложения. Этот вариант построения модулей использовался для построения модулей пользовательского интерфейса СИВС по объединению двух онтологий: онтологии инновационного процесса и графического представления интерфейса. Для описания онтологий использовался язык RDF/RDFS и редактор онтологий Protégé. Для обработки запросов к онтологиям применялась свободно распространяемая библиотека Rasqal RDF Query Library, реализующая язык запросов RDQL. Онтология инновационного процесса (см. рис. 3) построена на основе содержательной онтологии, представленной в [10], и базовым объектом в ней является карточка. Это может быть карточки информационного ресурса (книга, статья, ссылка и т.д.), исполнителя проекта, руководителя проекта, этапа проекта и т.д. Связи между этими объектами фактически определяют инновационный процесс. Пользователи могут присоединять к этой онтологии, например, к некоторым выделенным вершинам свои подонтологии, отражающие специфику их предметной области, средств моделирования, представления результатов моделирования и т.д. и также представленные карточками. Вершинами второй онтологии являются примитивы (неделимые элементы интерфейса). Примерами примитивов являются текстовая строка, поле ввода текста, ячейка таблицы, кнопка, ссылка, элемент меню, картинка и другие. Наличие двух отдельных онтологий позволяет при генерации кода комбинировать одну и ту же онтологию предметной области с различными онтологиями графического представления, например, для различных платформ или языков. Генератор кода решает следующую задачу – по онтологии прикладной области и онтологии пользовательского интерфейса он строит фрагмент описания элемента пользовательского интерфейса для работы с объектом данных. Объект может иметь любой тип, для которого есть описание в онтологии прикладной области. Работа с объектом может быть, например, просмотром объекта, редактированием объекта или его удалением. Для каждого типа объектов в одной из онтологий должно присутствовать описание всех реализуемых с ним видов работы. В пользовательском интерфейсе они реализуются визуальными режимами – по одному на каждый вид работы с объектом. Генерацию кода осуществляет программный модуль

(генератор), получающий на вход высокоуровневое описание в виде онтологии Результат работы программного модуля - текст на языке сборки (для GUI) или на языке разметки (для Web интерфейса). Выполнены две реализации генератора кода. Одна построена как внешний модуль на базе библиотеки ядра (пятый путь), другая - на базе Perl и HTML (шестой путь).

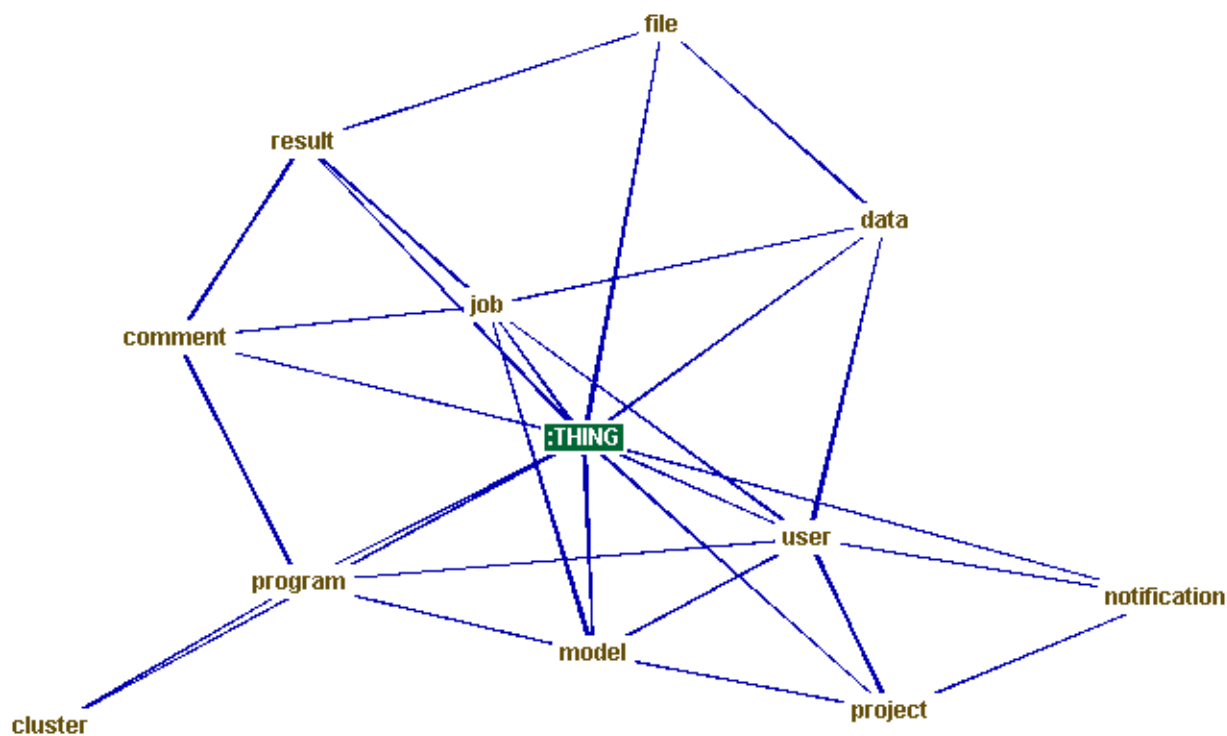


Рисунок 3. Фрагмент онтологии инновационного процесса

С практической точки зрения использование третьего варианта построения модулей означает, что пользователь получает возможность сформировать свое рабочее место, включающее рабочий стол, панели вызова операций, подокна просмотра объектов и их каталогов в различных режимах (списковом, древовидном, календарном), подокна, позволяющие организовать взаимодействие с ресурсами Центра по высокопроизводительным вычислениям: подокна проектов, загрузки модулей системы моделирования, компиляции модулей, управления модулями и т.д. (см. рис. 4)

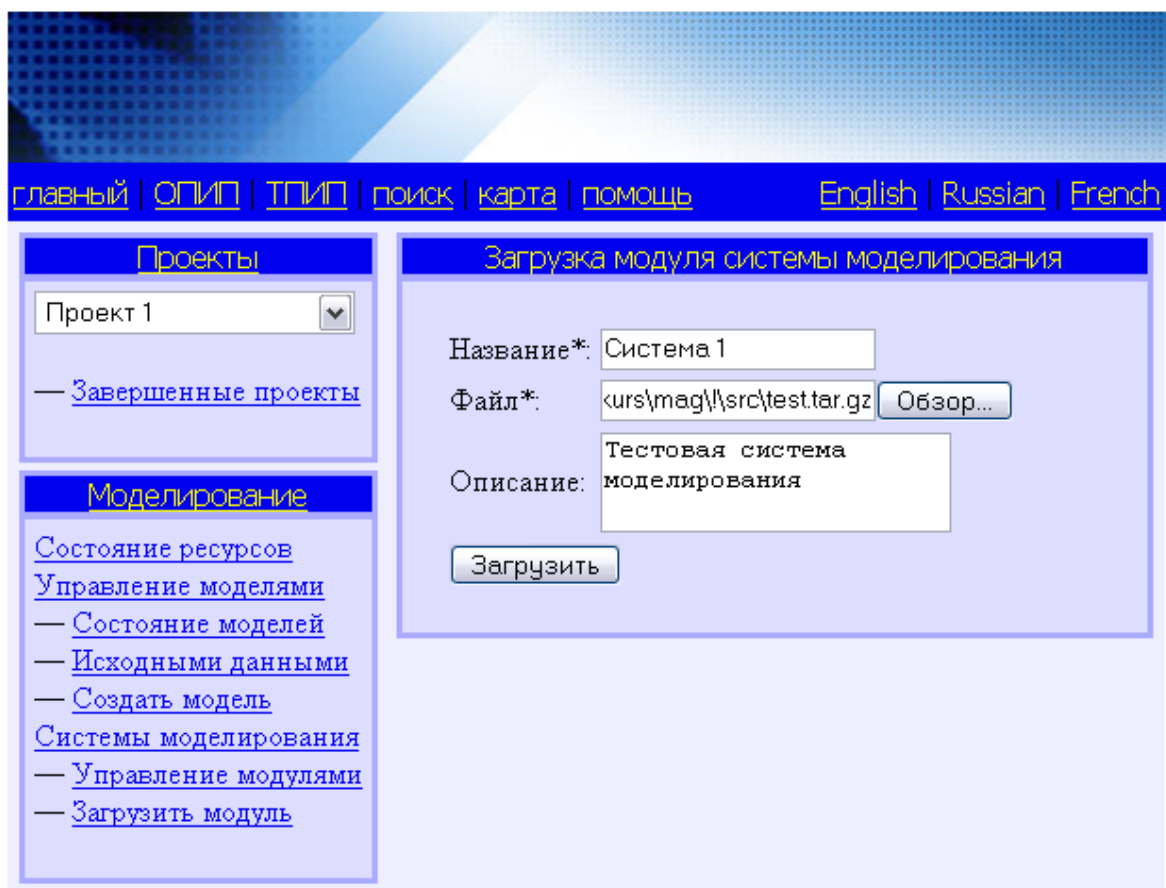


Рисунок 4. Фрагмент интерфейса рабочего места пользователя

Опираясь на предложенную технологию модульного расширения СИВС, разработчики и пользователи системы получают широкий спектр возможностей (от оперирования в терминах платформ и архитектур вычислительных систем и операционных систем до оперирования в терминах проблемных областей) по построению и модификации пользовательских интерфейсов, адаптации СИВС к новым аппаратно-программным платформам.

Заключение

В заключение отметим, что предложенная сборочная технология построения системы и в первую очередь, та ее часть, которая основана на онтолого-ориентированном подходе, обеспечивает пользователю возможность, получив систему в свое распоряжение, установить ее на своей программно-аппаратной платформе, адаптировать к своей предметной области, информационному окружению и т.д. Применение онтолого-ориентированного подхода позволяет пользователям описывать собственные и настраивать существующие приложения с помощью онтологий и использовать ресурсы

СуперЭВМ для выполнения имитационного моделирования отдельных этапов инновационного проекта. Вместе с тем пользователь имеет возможность помещать разработанные им модули в банк модулей на сервере разработчиков, пополнять информационное пространство системы на сервере разработчиков своими информационными материалами и т.п.

Список литературы

1. Алексеев А.С., Остапкевич М.Б., Пискунов С.В. О проекте информационно-вычислительной системы обеспечения инновационной деятельности // Оптимизация. Управление. Интеллект. – 2005 – №2(10) – С. 203-210.
2. Загорюлько Ю.А., Пискунов С.В., Боровикова О.И., Остапкевич М.Б. Распределенная Интернет-система формирования и поддержки инновационных проектов // Труды VIII-й международной конференции "Проблемы управления и моделирования в сложных системах" - Самара: Самарский Научный Центр РАН, 2006. - С. 427-432.
3. Алексеев А.С., Кратов С.В., Остапкевич М.Б., Пискунов С.В. Сибирская сетевая система поддержки инновационной деятельности //«Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ». Труды Всероссийской научной конференции (24-29 сентября 2007 г., г. Новороссийск). – М.: Изд-во МГУ, 2007. – С. 352-354.
4. Ostapkevich M.B. The DCMS library for open architecture application // Bull.Nov.Comp. Center, Comp. Science, 2004, № 21. - P. 99 - 111.
5. Веселов А.В. Применение онтологий для разработки графического пользовательского интерфейса. // Труды конференции молодых ученых. – Новосибирск: ИВМиМГ СО РАН, 2008. – С. 18-28.
6. А.В. Веселов, М.Б. Остапкевич, С.В. Пискунов. Автоматизированная генерация пользовательских интерфейсов для сетевой информационно-вычислительной системы // Тр. 7-ой междунар. Конф. памяти акад. А.П. Ершова «Перспективы систем информатики». Раб. семинар «Наукоемкое программное обеспечение». Новосибирск, ООО «Сибирское Научное Издательство», 2009. - С. 84–90.
7. А.В. Веселов, С.В. Кратов, М.Б. Остапкевич, С.В. Пискунов. Разработка пользовательских интерфейсов сетевой информационно-вычислительной системы на основе модели ориентированного подхода. // Серия: Информатика.

- Выпуск 9. Материалы пятой азиатской международной школы-семинара "Проблемы оптимизации сложных систем". Новосибирск. 2009.– С. 34 - 39.
8. Molina J.P., López M.G., López P.G. Model-based design and new user interfaces: current practices and opportunities. - CEUR Workshop Proceedings. – 2004.
9. Грибова В.В., Клещев А.С. Концепция разработки пользовательского интерфейса на основе онтологий - <http://www.iacp.dvo.ru/is/science.php?type=publications&uid=11>.
10. Зверев В.С., Унтура Г.А. Онтология инновационной деятельности // Инновационный потенциал научного центра: методологические и методические проблемы анализа и оценки / отв. ред. В.И. Суслов ; науч. ред. Н.А. Кравченко, Г.А. Унтура. - Новосибирск : ИЭОПИ СО РАН, 2007. - Гл. 1, § 3. - С. 35-54.

Сведения об авторах:

1. **ФИО:** Пискунов Сергей Владиславович

Ученая степень: к.т.н.

Должность и место работы: зав. лабораторией параллельных алгоритмов и структур, ИВМиМГ СО РАН

Контактный e-mail: piskunov@ssd.sccc.ru

Почтовый адрес организации: 630090 г. Новосибирск, пр. ак. Лаврентьева, 6

2. **ФИО:** Кратов Сергей Викторович

Ученая степень: б/с

Должность и место работы: м.н.с. лаборатории прикладных систем, ИВМиМГ СО РАН

Контактный e-mail: kratov@sscc.ru

Почтовый адрес организации: 630090 г. Новосибирск, пр. ак. Лаврентьева, 6

3. **ФИО:** Остапкевич Михаил Борисович

Ученая степень: б/с

Должность и место работы: м.н.с. лаборатории синтеза параллельных программ, ИВМиМГ СО РАН

Контактный e-mail: ostap@ssd.sccc.ru

Почтовый адрес организации: 630090 г. Новосибирск, пр. ак. Лаврентьева, 6р

4. ФИО: Веселов Александр Васильевич

Ученая степень: б/с

Должность и место работы: аспирант, НГТУ

Контактный e-mail: alexandr_v@ngs.ru

Почтовый адрес организации: 630092 г. Новосибирск, пр-т К. Маркса, 20