

ИСПОЛЬЗОВАНИЕ СБОРОЧНОЙ ТЕХНОЛОГИИ ДЛЯ ПОСТРОЕНИЯ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ СЕТЕВОЙ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

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

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

* Новосибирский государственный технический университет,
630092, Новосибирск, Россия

УДК 681.3.06

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: 1) уменьшить временные затраты на разработку графического интерфейса и упростить его последующую модификацию, поскольку разработчику предоставляется только декларативное описание; 2) 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.

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 interface of this domain. The approach makes it possible to a) 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.

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

Как известно, математическое моделирование является важной составляющей инновационного процесса, обеспечивающей успешную и быструю реализацию инновационного проекта. СуперЭВМ превратились в действенный инструмент математического моделирования как научных, так и прикладных задач больших размеров. Потребность в высокопроизводительных вычислениях постоянно растет, расширяется область применения суперЭВМ, вовлекаются новые пользователи, занимающиеся исследованиями в различных предметных

областях. Для эффективного использования суперЭВМ в инновационной деятельности необходимо создание сетевой информационной инфраструктуры, включающей не только системные и прикладные программные средства, обеспечивающие математическое моделирование на суперЭВМ, но и программные средства, обеспечивающие интерактивное общение разработчиков инновационного проекта, быстрый доступ к очень большим объемам общей и профессиональной информации и т. д. Поэтому актуальной является задача создания сетевой системы, способствующей извлечению из фундаментальных научных исследований и передаче в наукоемкую промышленность “технологичных” идей посредством активной поддержки решения сложных задач научно-исследовательской и опытно-конструкторской работы (НИОКР) созданными в сети Интернет (виртуальными) коллективами.

В работах [1–3] сформулированы требования к сетевой информационно-вычислительной системе (СИВС), описаны ее архитектура, назначение и вариант реализации. Основу системы составляет ядро, включающее библиотеку Динамически конфигурируемая модульная система (ДКМС) [4] и набор модулей, образующий каркас системы (рис. 1). Этот набор реализует минимальный набор функций, необходимых для использования системы: унифицированный доступ к наборам функций операционной системы (ОС) для осуществления 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 ввода, редактирования, поиска и просмотра разнотипных данных в системе через браузер. С помощью технологии сборки над ядром надстраиваются модули расширения. 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

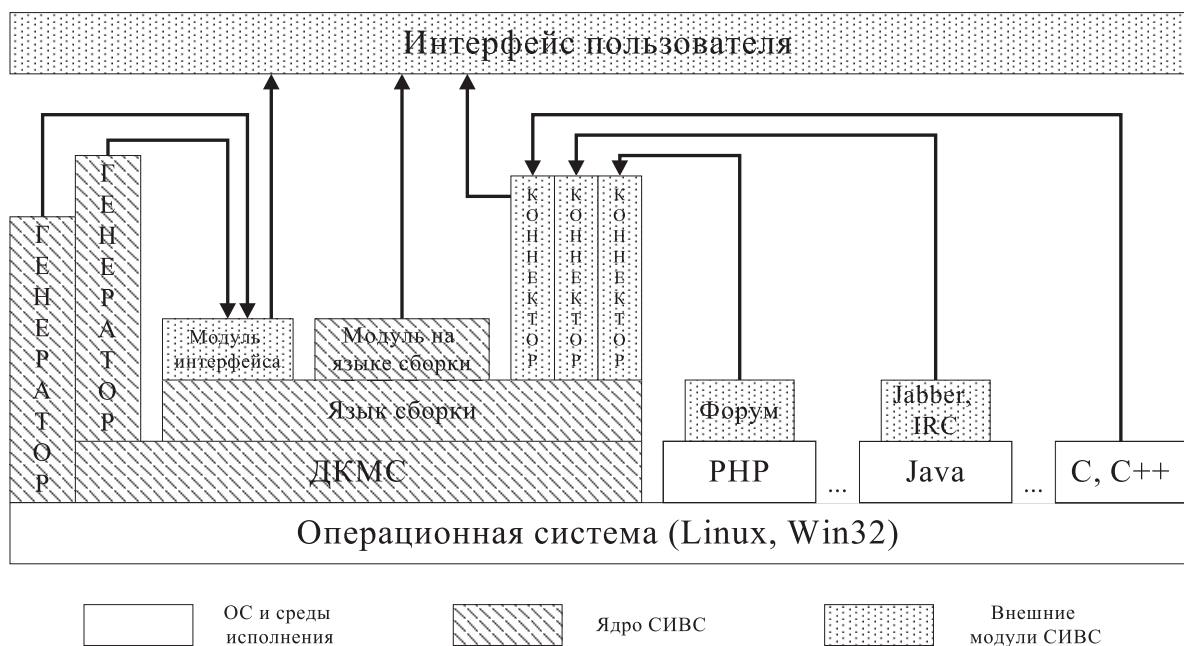


Рис. 1. Модульная архитектура СИВС

построения которой представлены в работах [5–7], 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 прежде всего понимается равноправное участие разработчиков и пользователей 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 (в первую очередь Web- и GUI-интерфейсов), что позволит существенно расширить круг пользователей СИВС, а значит, сделать ее развитие более динамичным и многоплановым.

Выбор пути развития СИВС. Все современные системы строятся как модульные с открытой архитектурой, обеспечивающие переносимость на различные платформы, масштабируемость, интероперабельность (кооперативную работу с другими открытыми системами) и дружелюбность пользовательского интерфейса. Существует два пути построения СИВС как программного продукта.

1. Сложную систему создает (от начала и до конца) подходящий по численному составу и квалификации коллектив разработчиков. То, что система проектируется одним коллективом и при этом используется открытая архитектура, обеспечивает повторную используемость модулей, облегчает тестирование, отладку, сопровождение, логическую целостность системы с точки зрения пользователя и ее модернизацию.

2. Небольшой коллектив создает только ядро системы и делает доступными 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 набора полезных функций эти модули представляют пример того, как подобные модули могут разрабатываться пользователями, которые нуждаются в некоторой функции и желают реализовать ее в рамках подобной системы. Если система оказывается привлекательной для достаточного числа пользователей, набор ее функций начинает пополняться такими пользователями без участия разработчиков. Это, в свою очередь, может привлечь еще большее число пользователей. В случае удачных систем возникает эффект снежного кома. Именно этот путь выбран разработчиками для дальнейшего развития СИВС.

Для того чтобы развитие системы было успешным, должны выполняться следующие условия:

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 (С, С++, Java, Perl, PHP), такие средства, как Protégé.

3. Допускается участие в развитии системы пользователей различной квалификации от программистов до экспертов в конкретной предметной области, которые могут только точно ответить на вопрос, что модуль должен делать.

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.

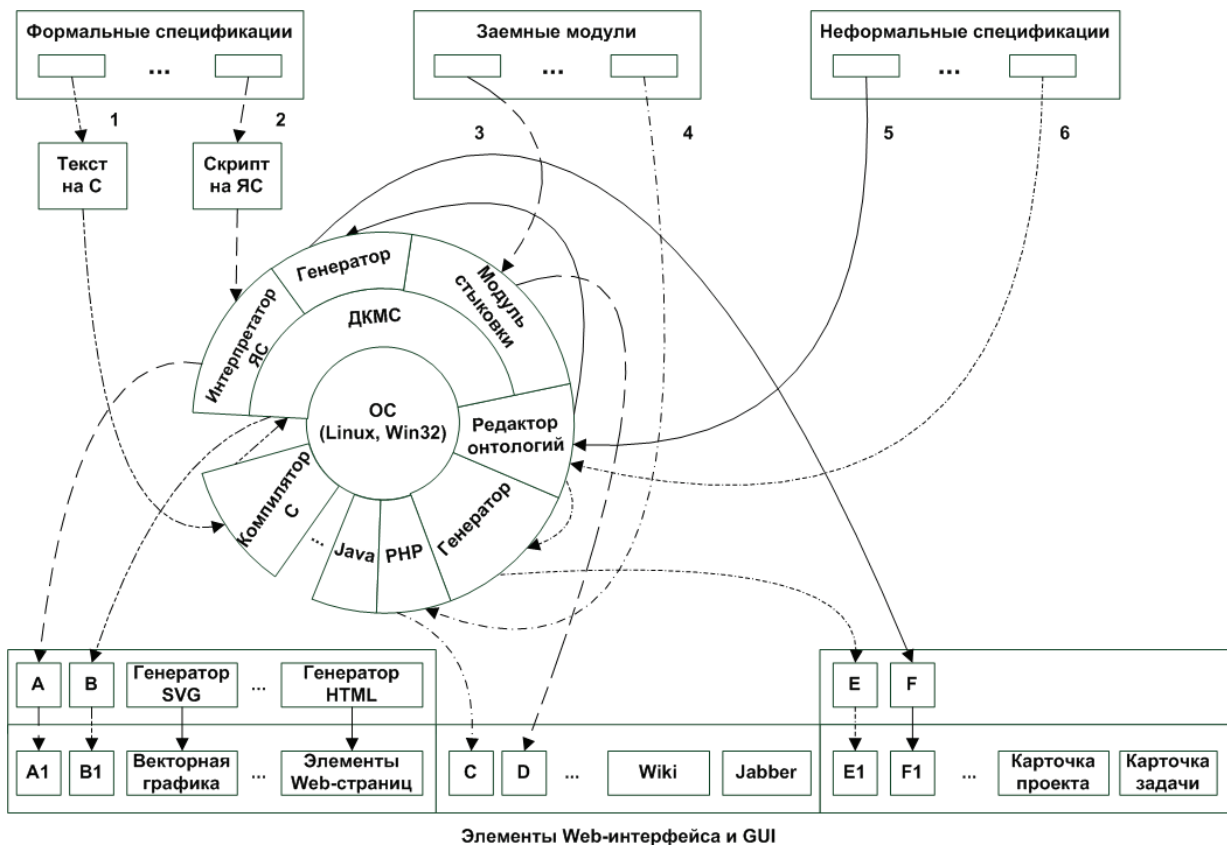


Рис. 2. Схема сборочной технологии

Первый вариант основан на библиотеке ДКМС, которая является надстройкой над операционной системой (Win32, Linux) и состоит из процедур. Все модули, построенные на основе этой библиотеки, содержат только обработчики событий. Все межмодульные взаимодействия происходят на основе событий. Событийный интерфейс обеспечивает возможность как встраивания новых, так и замещения старых модулей без модификации исходных текстов ядра. Первый вариант предполагает два пути построения модулей (пути 1 и 2 на рис. 2). Модули, строящиеся на основе ДКМС, представлены динамически компонуемыми библиотеками, написанными на C/C++ (модуль B на рис. 2). Модули, написанные на языке сборки (ЯС на рис. 2), представлены скриптами (модуль A 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 A может быть использован для решения следующей задачи. Некоторый пользователь, применяющий приложение KOrganizer для хранения контактов, хочет обеспечить их перенос в информационное пространство СИВС. Для этого он строит на языке C модуль поддержки формата данных KOrganizer для хранения контактной информации, который содержит обработчик события, возникающий в тот момент, когда происходит считывание объекта данных в соответствующем формате. При этом событийный интерфейс ДКМС позволяет интегрировать данный модуль в СИВС без какой-либо модификации ее 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 исполнения в ИВС процедур, написанных на Java; реализация генератора текстов фрагмен-

тов кода на VBScript для улучшения интерактивности и дизайна Web-страниц, передаваемых клиенту с MS Internet Explorer; и др.

Второй вариант — присоединение к ИВС заемных модулей (форум, Jabber-сервер, Wiki и др.). Этот вариант также имеет два пути (пути 3, 4 на рис. 2). В обоих случаях 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 программного обеспечения, 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.

Рассмотренные выше варианты построения модулей и сборки из них приложений можно назвать классическими. Эти варианты соответствуют наиболее распространенному пути 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 может работать только с определенным набором объектов. Поэтому при добавлении новых объектов требуется модификация программного кода и перекомпиляция приложения. Существующие объекты можно изменить, используя их описания в виде шаблонов. Однако в любом случае требуется помощь программиста.

Между тем проблема заключается в том, что при разработке СИВС, с одной стороны, невозможно заранее описать все объекты инновационной деятельности во всех ее сферах, с другой — желательно не потерять потенциальных пользователей системы, желающих адаптировать СИВС к своей предметной области, но не являющихся квалифицированными программистами. Поэтому необходим еще один вариант построения модулей, дополняющий представленные выше и предназначенный для использования не только программистами.

Данный (третий) вариант основан на модели ориентированном подходе [8, 9] к построению приложений и ориентирован на то, чтобы описать “что делать”, а не “как делать”. Суть подхода состоит в следующем: строится высокоуровневое описание интерфейса — его модель, по которой затем генерируется программный код с использованием технологии сборки. Такой подход позволяет уменьшить временные затраты на разработку пользовательского интерфейса и упростить его последующую модификацию, поскольку пользователь использует только декларативное описание, 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 использовались 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 (рис. 3) построена на основе содержательной онтологии, представленной в [10]. Базовым объектом в ней является карточка (карточки информационного ресурса (книга, ста-

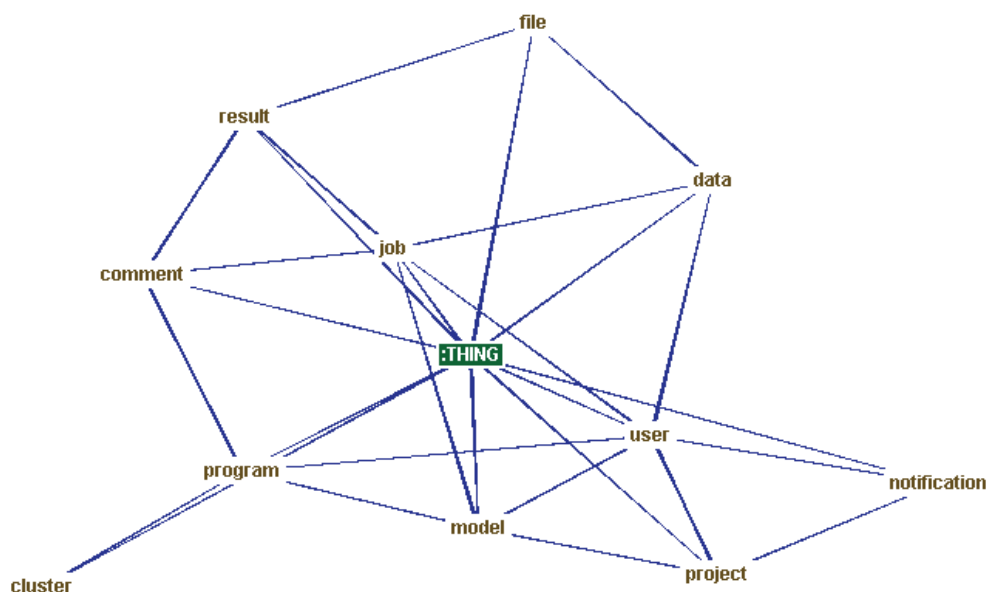


Рис. 3. Фрагмент онтологии инновационного процесса

тя, ссылка и т. д.), исполнителя проекта, руководителя проекта, этапа проекта и т. д.). Связи между данными 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), например текстовая строка, поле ввода текста, ячейка таблицы, кнопка, ссылка, элемент меню, картинка и др. 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 построить фрагмент описания элемента пользовательского интерфейса для работы с объектом данных. Объект может быть объектом любого типа, для которого имеется описание в онтологии прикладной области. Работа с объектом может представлять собой, например, просмотр объекта, редактирование объекта или его удаление. Для объектов каждого типа в одной из онтологий должно присутствовать описание всех видов работы с ним. 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. Результатом работы программного модуля является текст на языке сборки (для GUI) или на языке разметки (для Web-интерфейса). Выполнены две реализации генератора кода, одна из которых построена как внешний модуль на базе библиотеки ядра (путь 5), другая — на базе Perl и HTML (путь 6).

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:

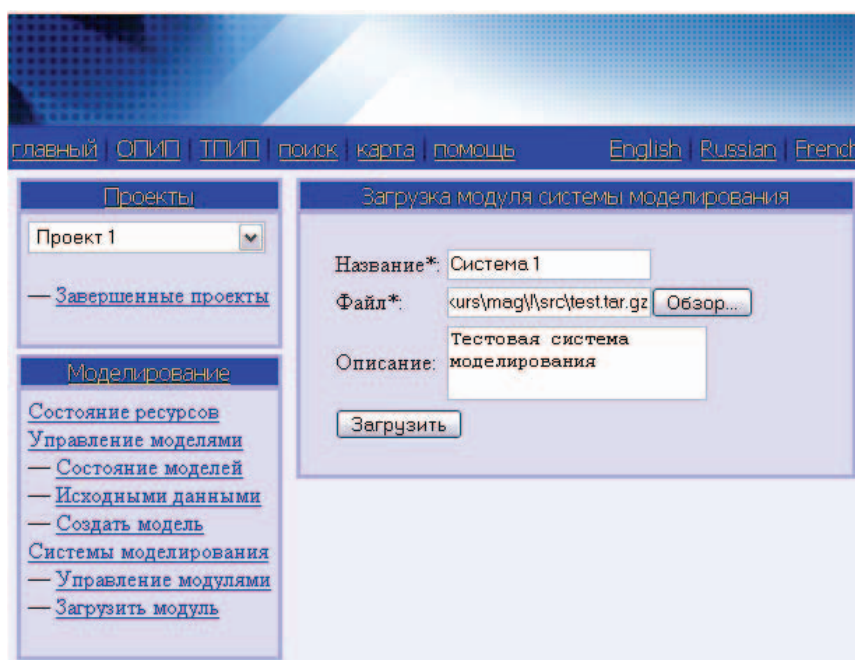


Рис. 4. Фрагмент интерфейса рабочего места пользователя

подокна проектов, загрузки модулей системы моделирования, компиляции модулей, управления модулями и т. д. (рис. 4).

Используя 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. Предложенная сборочная технология построения системы, в первую очередь та ее часть, которая основана на онтолого-ориентированном подходе, обеспечивает возможность пользователю, получившему 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. С. 203–210.
2. ЗАГОРУЛЬКО Ю. А., ПИСКУНОВ С. В., БОРОВИКОВА О. И., ОСТАПКЕВИЧ М. Б. Распределенная Интернет-система формирования и поддержки инновационных проектов // Тр. 8-й Междунар. конф. "Проблемы управления и моделирования в сложных системах", Самара, 24–28 июня 2006 г. Самара: Самар. науч. 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: Тр. Всерос. науч. конф., г. Новороссийск, 24–29 сент. 2007 г. М.: Изд-во МГУ, 2007. С. 352–354.
4. ОСТАПКЕВИЧ М. В. The DCMS library for open architecture application // Bull. Novosibirsk Comput. Center: Comput. Sci. 2004. N 21. P. 99–111.
5. ВЕСЕЛОВ А. В. Применение онтологий для разработки графического пользовательского интерфейса // Тр. конф. молодых ученых. Новосибирск: ИВМиМГ СО РАН, 2008. С. 18–28.
6. ВЕСЕЛОВ А. В., ОСТАПКЕВИЧ М. Б., ПИСКУНОВ С. В. Автоматизированная генерация пользовательских интерфейсов для сетевой информационно-вычислительной системы // Тр. 7-й Междунар. конф. памяти акад. А. П. Ершова “Перспективы систем информатики”. Новосибирск: Сиб. науч. изд-во, 2009. С. 84–90.
7. ВЕСЕЛОВ А. В., КРАТОВ С. В., ОСТАПКЕВИЧ М. Б., ПИСКУНОВ С. В. Разработка пользовательских интерфейсов сетевой информационно-вычислительной системы на основе модели ориентированного подхода // Материалы 5-й Азиат. междунар. школы-семинара “Проблемы оптимизации сложных систем”, Бишкек (Кыргызстан), 12–22 авг. 2009 г. Сер. Информатика. Вып. 9. Новосибирск: ИВМиМГ СО РАН, 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 // Workshop “Making model-based UI design practical: Usable and open methods and tool”, Funchal (Madeira, Portugal), 13 Jan. 2004. P. 245–247.
9. ГРИБОВА В. В., КЛЕЩЕВ А. С. Концепция разработки пользовательского интерфейса на основе онтологий. [Электрон. ресурс]. <http://www.iacr.dvo.ru/is/science.php?type=publications&luid=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. Новосибирск: ИЭОПП СО РАН, 2007. С. 35–54.

*Пискунов Сергей Владиславович — канд. техн. наук, зав. лабораторией
Института вычислительной математики и математической геофизики СО РАН;
e-mail: piskunov@ssd.sccc.ru;*

*Кратов Сергей Викторович — мл. науч. сотр. Института вычислительной математики
и математической геофизики СО РАН; e-mail: kratov@ssccc.ru;*

*Остапкевич Михаил Борисович — мл. науч. сотр. Института вычислительной математики
и математической геофизики СО РАН; e-mail: ostap@ssd.sccc.ru;*

*Веселов Александр Васильевич — асп. Новосибирского государственного
технического университета; e-mail: alexandr_v@ngs.ru*

Дата поступления — 6.09.2010