

УДК 004.4'2

М. А. ВЕРХОТУРОВ, Н. М. САДРЕТДИНОВА

**ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ БИЗНЕС-ПРИЛОЖЕНИЙ
НА БАЗЕ ИНТРАНЕТ-ТЕХНОЛОГИЙ**

Рассматриваются вопросы разработки модели системы интранет-приложений, архитектуры соответствующих инструментальных средств, а также процесс создания с их помощью бизнес-приложений. *Интранет-технологии; программные инструментальные средства*

В современных условиях, когда информация становится важным производственным фактором, большое значение приобретает автоматизация внутренних бизнес-процессов организации. На сегодняшний день различные виды бизнес-процессов зачастую автоматизированы с помощью разных программных продуктов. Информация труднодоступна и порой оказывается противоречивой, затрудняя принятие решений. Комплексная автоматизация бизнес-процессов может быть проведена на базе ERP-систем (Enterprise Resource Planning, управление ресурсами предприятия). Однако в условиях, когда отсутствует организационная интеграция и не формализованы бизнес-процессы, внедрение ERP не всегда эффективно [2]. Особенно остро проблема организационной интеграции стоит в России.

Более эффективный в данных условиях путь комплексной автоматизации предприятия — интеграция существующего программного обеспечения и разработка дополнительных модулей с поэтапным внедрением. Возникает необходимость создания единого информационного центра предприятия, который предоставлял бы доступ каждому сотруднику предприятия ко всем необходимым данным, документам и приложениям. Такой центр повышает прозрачность и контролируемость бизнес-процессов на предприятии.

Информационный центр предприятия может быть реализован на базе интранет-технологий в силу ряда причин: удобство использования веб-браузера в качестве клиента, быстрая и гибкая адаптация к меняющимся условиям, возможность интеграции с другими приложениями, возможность удаленной работы. Такой центр на базе интранет-технологий может стать промежуточным этапом автоматизации предприятия, подготовляющим «фундамент» для внедрения ERP-системы.

В связи с высокой эффективностью применения интранет-технологий для автоматизации внутренних процессов предприятия представляет интерес создание специализированных инструментальных средств разработки систем интранет-приложений.

**ОСОБЕННОСТИ ЗАДАЧИ СОЗДАНИЯ
ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ**

Исторически можно выделить несколько направлений развития инструментальных средств: системы управления контентом, инструменталь-

ные средства на базе HTML-редакторов, инструментальные среды программирования. Системы управления контентом, например, Metric Optimizer, ориентированы на пользователей среднего уровня квалификации и предоставляют ограниченные возможности модификации приложений и адаптации их под специфику предприятия. Инструментальные средства на базе HTML-редакторов (MS FrontPage, Macromedia Homesite) и среды программирования (Microsoft.Net, Borland JBuilder), напротив, предоставляют множество средств реализации веб-приложений. Такие средства зачастую перенасыщены функциональностью и, как правило, сложны в освоении, дорогостоящи, а главное — не предоставляют упрощенных возможностей управления приложениями для пользователей с небольшой квалификацией.

Наибольший интерес представляют специализированные средства разработки интранет-систем или порталов — IBM WebSphere, Oracle9i Application Server Portal Developer Kit [1]. Такие средства в большей степени удовлетворяют требованиям, предъявляемым к инструментальным средствам разработки интранет-приложений, однако также обладают рядом недостатков: предъявляют высокие требования к аппаратным ресурсам; предоставляют возможности только для высококвалифицированных специалистов, затрудняя быструю адаптацию приложений к меняющимся бизнес-процессам; включают большой объем спецификации, сложной для изучения.

Таким образом, возникает задача построения инструментальных средств разработки интранет-приложений, которые, с одной стороны, позволили бы пользователям достаточно гибко модифицировать и создавать приложения без непосредственной работы с исходным кодом, а с другой — должны предоставлять возможности изменения кода и включать достаточно подробную спецификацию для квалифицированных специалистов.

При декомпозиции процесса разработки инструментальных средств можно выделить следующие задачи: разработка модели системы интранет-приложений, формализация процесса разработки интранет-приложения, автоматизация процесса разработки интранет-приложения и разработка документации и спецификаций.

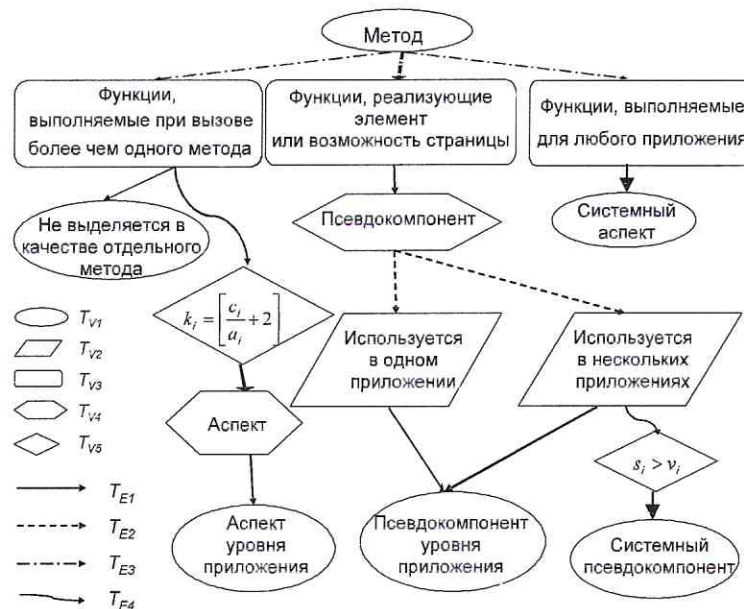


Рис. 1. Семантическая сеть — модель процесса декомпозиции системы интранет-приложений

МОДЕЛЬ СИСТЕМЫ ИНТРАНЕТ-ПРИЛОЖЕНИЙ

Для решения задачи разработки модели системы интранет-приложений выявлены особенности современных подходов к программированию, которые позволяют снизить трудоемкость разработки с учетом специфики области применения. Согласно проведенному анализу, наиболее эффективно использовать элементы компонентного, объектно-ориентированного и аспектно-ориентированного [3] подходов.

В модели интранет-системы выделены системный уровень и уровень приложения. Используются следующие принципы компонентной объектной модели: идея сборки приложения из компонентов и строгое определение интерфейса как решение проблемы совместимости различных версий компонентов, позволяющее приложениям развиваться независимо друг от друга. Учитывая специфику системы интранет-приложений, от ряда принципов можно отказаться. В частности, не имеет значения условие бинарного представления исходных файлов в силу специфики традиционных (скриптовых) языков программирования веб-приложений. Кроме того, можно отказаться от традиционного понятия компонента как хранилища одного или нескольких классов. Так как все приложения интранет-системы находятся на едином сервере, можно хранить компоненты в общем классе (модуле). Компоненты удобно определить как методы, а не классы, потому что любой компонент веб-приложения имеет дело с рядом одинаковых объектов: http-запрос и его параметры, шаблон страницы и др. Поэтому им дано название псевдокомпонентов. Псевдокомпонент — это метод, реализующий определенный элемент или возможность веб-страницы и обладающий неизменным интерфейсом вызова. Модуль, содержащий общие для нескольких прило-

жений (системные) псевдокомпоненты, называется рабочим классом системы интранет-приложений и представляет собой системный уровень модели.

На уровне приложения большое значение имеют принципы аспектно-ориентированного подхода — идеи локализации сквозной функциональности. Учитывая наличие в интранет-системах достаточно большого количества общих функций, применение данного принципа позволяет дополнительно снизить трудоемкость разработки. Сквозные функции каждого конкретного приложения можно объединить в аспектные модули: для каждого приложения — свой аспектный модуль. Показательные примеры такого аспекта на уровне приложения — установка соединения с базой данных приложения или запись действия пользователя в журнал.

Для автоматической расстановки аспектов потребовался бы специальный компилятор. Но так как псевдокомпоненты представляют собой методы единого класса, можно сделать аспектный модуль наследником рабочего класса. Пусть приложение обращается не напрямую к методам псевдокомпонентам рабочего класса, а к промежуточным методам аспектного модуля. Такие переопределенные методы и будут являться точками применения аспектов. В аспектный модуль до и/или после вызова наследуемого метода можно вставить необходимую аспектную функциональность. Дополнительно в аспектный модуль выносятся псевдокомпоненты, уникальные для данного приложения.

За генерацию конкретной страницы отвечают методы модуля «сборки», формирующие страницу из компонентов обоих уровней. Кроме того, необходим скрипт, ответственный за инициализацию приложения и вывод страницы.

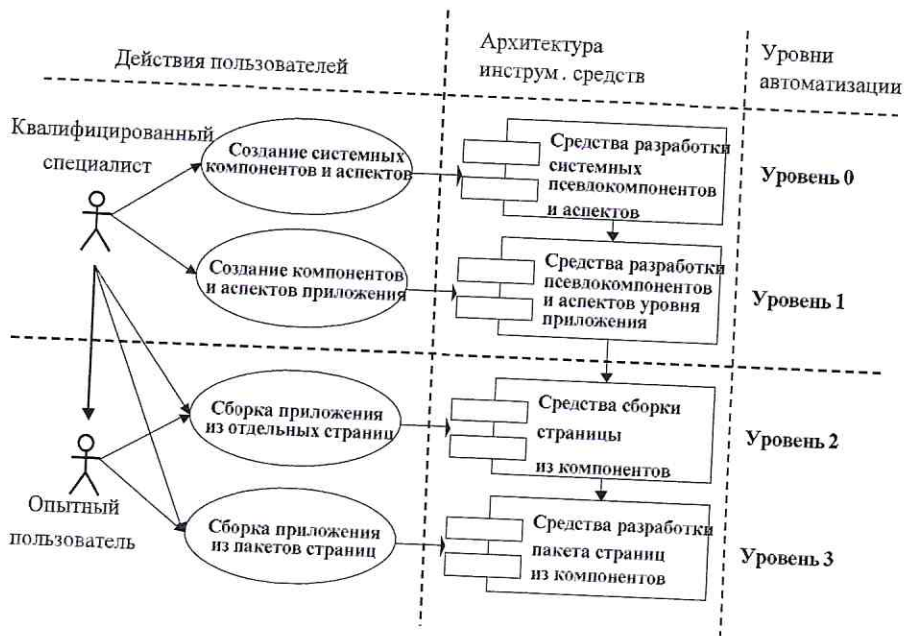


Рис. 2. Архитектура инструментальных средств

Разница в трудоемкости разработки приложения «с нуля» или с использованием ранее разработанных системных псевдокомпонентов представляет собой:

$$\Delta T = T_1 - T_2 = \sum_{i=1}^n (s_i - v_i), \quad (1)$$

где s_i — трудоемкость разработки компонентов приложения, которые могут быть заменены системными псевдокомпонентами (выполняют идентичные функции); v_i — трудоемкость вызова системного псевдокомпонента, n — количество системных псевдокомпонентов. Очевидно, что для максимального снижения трудоемкости согласно предложенной псевдокомпонентной модели необходимы условия:

$$\Delta T \rightarrow \max; \quad n \rightarrow \max; \quad s_i > v_i. \quad (2)$$

Далее рассмотрим сравнительную трудоемкость реализации приложения без выделения или с выделением аспектов:

$$T_1 = k_i a_i, \quad k_i \in Z, \quad (3)$$

$$T_2 = c_i + a_i, \quad (4)$$

где a_i — трудоемкость реализации некоторой сквозной функции (аспекта) i ; c_i — трудоемкость определения точки вставки аспекта i ; функцию i необходимо выполнить при вызове k_i методов, Z — множество целых чисел.

Тогда минимальное k_i , при котором выгодно выделить ту или иную функцию в аспект, будет равно

$$k_{i \min} = \left[\frac{c_i}{a_i} + 2 \right]. \quad (5)$$

Общее снижение трудоемкости за счет применения псевдокомпонентной модели с использованием аспектов выглядит как

$$\Delta T = \sum_{i=1}^n (s_i - v_i) + \sum_{j=1}^m (a_j(k_j - 1) - c_j). \quad (6)$$

Для фактических подсчетов под трудоемкостью в данном случае можно понимать количество строк кода q , умноженное на весовой коэффициент — экспертную оценку сложности кода w .

Отдельную задачу представляет собой задача декомпозиции интранет-приложения согласно предложенной модели и при условии минимизации трудоемкости разработки. Необходимо выделить псевдокомпоненты и аспекты на системном уровне и на уровне приложения. Разработанные принципы декомпозиции отражает семантическая сеть (рис. 1).

РАЗРАБОТКА ИНТРАНЕТ-ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ

Прежде всего, необходима формализация бизнес-процесса, подлежащего автоматизации данным приложением. В результате можно определить разделы приложения и функции разделов. Далее производится инициализация приложения, затем — разработка разделов и, наконец — генерация готового приложения. Согласно модели, на инструментальные средства возложены такие функции, как инициализация приложения, разработка разделов приложения и генерация готового приложения. Очевидно, что наиболее трудоемкий процесс — разработка разделов приложения. Необходимо предоставить возможности автоматизации разработки разделов как высококвалифицированным специалистам, так и пользователям с

относительно невысоким уровнем квалификации. В соответствии с выделенными уровнями декомпозиции можно выделить четыре уровня автоматизации. Уровни автоматизации отражает архитектура инструментальных средств.

В основе архитектуры инструментальных средств лежит предложенная модель системы интранет-приложений. Обозначим работу с рабочим классом как **нулевой уровень** автоматизации. Инструментальные средства предоставляют визуальный интерфейс для работы с аспектным модулем (**первый уровень**) и модулем «сборки» компонентов (**второй уровень**). Кроме того, предусмотрены средства автоматизации высокого уровня (**третий уровень**). Таким образом, инструментальные средства предоставляют четыре уровня автоматизации (см. рис. 2).

Действия на первых двух уровнях автоматизации осуществляются квалифицированными специалистами с помощью специализированных средств этих уровней. Далее специалисты могут передать работы по созданию приложений опытным пользователям, для этого используются средства второго и третьего уровней автоматизации. Кроме того, если приложение состоит только из системных псевдокомпонентов и готовый их набор достаточен для его создания, пользователи могут разработать его самостоятельно.

В модели системы интранет-приложений за ту или иную функциональную возможность или элемент отвечает отдельный метод-псевдокомпонент. Некоторые из таких функций и элементов тесно связаны между собой. В частности, предпроцессинг страницы — методы, которые необходимо вызвать для формирования той или иной страницы, и постпроцессинг, возникающий, когда страница включает в себя форму, предназначенную для ввода данных пользователем. В этом случае требуется обработка введенных данных. Дополнение этих методов формированием HTML-шаблонов позволяет обеспечить полностью автоматическое создание кода для реализации той или иной функциональной возможности.

Пример типичной функциональной возможности: редактирование записи в таблице БД. Например, пользователю нужно предоставить возможность редактировать позицию в прайс-листе. Для этого достаточно выбрать метод «Редактирование записи таблицы» и таблицу, содержащую прайс-лист. Будут созданы: шаблон, содержащий форму и поля ввода, необходимые для редактирования; метод предпроцессинга — заполнение формы ранее сохраненными в базе данными; метод постпроцессинга — сохранение введенной пользователем информации в базе данных и переход к следующей странице.

Для пользователей высокой квалификации на последнем этапе реализации функциональной возможности предоставляется возможность откорректировать созданный исходный код. Разрабатываемые инструментальные средства позволяют создавать взаимосвязанные метод предпроцессинга, метод постпроцессинга и HTML шаблон одновременно. Таким образом, в рассматри-

ваемых инструментальных средствах реализован принцип автоматической сборки взаимосвязанных компонентов.

Итак, специалистам предоставлены средства автоматизации для создания типовых страниц и псевдокомпонентов, пользователям предоставлена возможность самостоятельно создавать и модифицировать приложения без непосредственной работы с исходным кодом. Снижение трудоемкости достигается, в том числе, и за счет применения предложенной модели интранет-системы, направленной на минимизацию количества кода и позволяющей избежать его дублирования.

РЕАЛИЗАЦИЯ МОДЕЛИ СИСТЕМЫ ИНТРАНЕТ-ПРИЛОЖЕНИЙ И ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ. ОЦЕНКА ТРУДОЕМКОСТИ РАЗРАБОТКИ

Рассмотренная модель системы интранет-приложений реализована в виде базового комплекса программных модулей с использованием Perl + mod_perl + СУБД MySQL. Разработана пилотная версия инструментальных средств «Интранет-мастер», основанная на предложенных подходах.

Рассмотренные подходы были применены при создании системы интранет-приложений «NGT Портал», внедренной в крупном проектом институте ЗАО «УфаНИИНефть», где она успешно функционирует и развивается в течение двух лет [4]. «NGT Портал» основывается на предложенной модели системы интранет-приложений, ряд разделов разработан с помощью рассматриваемых инструментальных средств. Он включает ленту новостей на рабочем столе каждого сотрудника, справочник информации о сотрудниках, систему навигации по зданию, систему заявок сервисным службам предприятия, систему анонимных опросов, систему совместной работы над документами, электронный архив проектной документации и ряд других приложений.

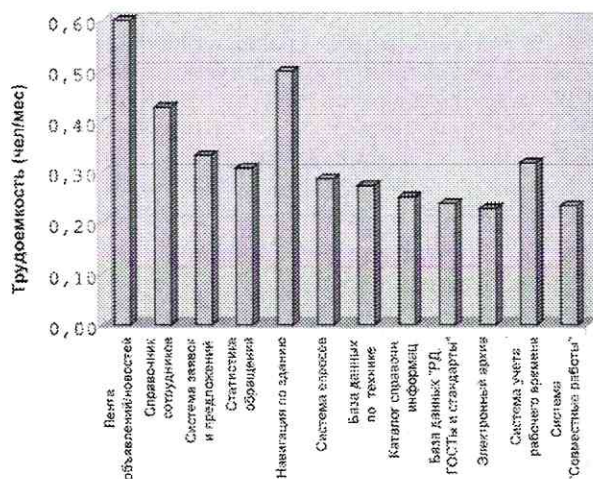


Рис. 3. Динамика относительной трудоемкости разработки приложений по мере развития системы «NGT Портал»

Таблица

Фактическая трудоемкость реализации интранет-приложений, чел/дней

Приложение	I вариант реализации	MS SharePoint Portal	II вариант реализации			
			уровни автоматизации 0–1		уровни автоматизации 2–3	
			рапес обученные специалисты	включая обучение	рапес обученные специалисты	включая обучение
Каталог СИФ	11,3	—	4,2	14,2	1,7	4,7
Система заявок и предложений	14,2	5	6,5	16,5	3,6	6,6

Для оценки трудоемкости была использована модель СОСОМО, учитывающая возможности повторного использования компонентов [5]. Фактическая трудоемкость реализации, включая варианты создания приложения с помощью средств MS SharePoint, различных уровней автоматизации разработанных инструментальных средств и время на обучение сотрудников работе с ними, представлена в таблице.

Практически, по мере развития системы «NGT Портал», рабочего модуля (создание новых псевдокомпонентов) и инструментальных средств трудоемкость разработки новых интранет-приложений снижается.

На рис. 3 показана динамика относительной трудоемкости разработки приложений по мере развития «NGT Портала».

Значительная часть кода повторяющихся элементов может быть создана автоматическим образом. Повторное использование компонентов для различных приложений возможно благодаря использованию принципов компонентной модели. Основное внимание разработчиков может быть сконцентрировано на создании уникальных псевдокомпонентов для реализации специфических функций новых приложений.

Исследования показали, что применение предложенной модели интранет-системы позволяет снизить трудоемкость разработки простых приложений, использующих в основном стандартные псевдокомпоненты, почти на 70%. Применение инструментальных средств позволяет снизить трудоемкость практически любых, в том числе достаточно сложных и масштабных, приложений почти на 40%. Чем дольше развивается система и чем больше в ней приложений, тем ниже трудоемкость разработки новых разделов.

ЗАКЛЮЧЕНИЕ

В современных условиях, когда информация стала важным производственным фактором, большое значение приобретает создание единого информационного центра, способствующего интеграции предприятия. Такой центр эффективно разрабатывать на базе интранет-технологий путем создания системы интранет-приложений для автоматизации бизнес-процессов предприятия. В связи с тем, что бизнес-приложения требуют быстрой разработки и адаптации одновременно с развитием бизнес-процессов, возникает необходимость в эффективных инструментальных средствах разработки интранет-приложений.

С учетом особенностей внутренних бизнес-процессов и согласно требованиям к интранет-приложениям, разработана модель интранет-системы. Созданы инструментальные средства разработки интранет-приложений, предоставляющие четыре уровня автоматизации. Первые два уровня предоставляют средства автоматизации для специалистов. Средства более высокого уровня позволяют высококвалифицированным специалистам значительно сократить время на выполнение рутинных операций и программирование стандартных элементов, уделяя максимум внимания специфическим особенностям каждого конкретного приложения. Пользователям предоставлена возможность создавать и модифицировать интранет-приложения самостоятельно, без непосредственной работы с исходным кодом.

Особенное значение имеет создание единого информационного центра для научно-исследовательских организаций и обществ. Созданная на базе предложенных подходов в проектом институте ЗАО «УфНИПИнефть» более двух лет успешно работает и развивается интранет-система «NGT Портал». Исследования трудоемкости разработки и быстродействия данной системы доказывают эффективность предложенных подходов.

СПИСОК ЛИТЕРАТУРЫ

1. **Интернет-порталы:** содержание и технологии: сб. науч. ст. / Ред.: А. Н. Тихонов [и др.]; ГНИИ ИТТ «Информика». М.: Просвещение, 2003. Вып. 1. 720 с.
2. **Нельсон, Р.** Интеграция ERP: Организационная интеграция / Рэнни Нельсон // URL: http://consulting.ru/econs_wp_4651.
3. **Павлов, В.** Аспектно-ориентированное программирование / В. Павлов // Технология клиент-сервер. 2003. № 4. С. 35–54.
4. **Садретдинова, Н. М.** NGT Портал и его роль в совершенствовании работы проектного института / Н. М. Садретдинова // Интернет и нефтегаз – 2004: докл. и статьи всерос. конф. Трехгорный: ООО «Интернет-лаборатория ORA», 2004. С. 31–43.
5. **Соммервилл.** Инженерия программного обеспечения. М.: Издат. дом «Вильямс», 2002. 624 с.

ОБ АВТОРАХ



Верхотуров Михаил Александрович, проф., каф. выч. матем. и киберн. Дипл. инж.-системотех. по АСУ (УАИ, 1983). Д-р техн. наук по АСУ (УГАТУ, 2000). Иссл. в обл. оптимизации размещения двух- и трехмерн. геометрич. объектов.



Садретдинова Нелли Маратовна, зав. сектором в «ЮНГ-НТЦ Уфа». Дипл. мат.-экон. (УГАТУ, 2001). Канд. техн. наук по мат. и прогр. обесп. выч. машин, комплексов и компьют. сетей (УГАТУ, 2004). Иссл. в обл. интернет/интраст-технологий.

УДК 004.4'2

Г. А. МАКЕЕВ

СВОЙСТВО УСТОЙЧИВОСТИ ДЛЯ СИСТЕМ РАСПРЕДЕЛЕННОЙ СОВМЕСТНОЙ ФИЛЬТРАЦИИ ИНФОРМАЦИИ

Рассматривается теоретический подход к анализу системы совместной фильтрации. Для этого формализуются критерии управляемости и устойчивости по пользователям и сообщениям. Предлагаются различные виды функций транзитивного замыкания и агрегации сообщений, производится их формальный анализ, позволяющий выбрать наилучшую из них по соответствию формализованным критериям. *Системы совместной фильтрации; устойчивость; формализация.*

Интеграция новых информационных технологий с глобальными компьютерными сетями, с одной стороны, увеличивает объем данных, доступных и подлежащих обработке, а с другой стороны, предоставляет возможность организации распределенных вычислений, в том числе и для распределенной обработки данных.

К этой области, в частности, относятся получающие все более широкое распространение и применение системы рекомендаций (Recommender Systems) [6] и системы совместной фильтрации информации (Collaborative Filtering Systems) [8].

Системы совместной фильтрации требуют от пользователя сначала предоставить некоторый набор своих рекомендаций, выявляя тем самым его предпочтения. Основной целью системы является определение на основе рекомендаций тех пользователей, предпочтения которых схожи с предпочтениями данного пользователя, и формирование рекомендаций для него на основе рекомендаций найденных пользователей.

Пользователи фиксируют свои предпочтения, оценивая фильтруемые элементы (сообщения, книги, звукозаписи, и т.д.). Традиционная система совместной фильтрации, например, такая как GroupLens [7], ищет пользователей со схожими предпочтениями среди всех своих пользователей. Далее система предлагает пользователю

в качестве результата совместной фильтрации те элементы, которые были высоко оценены «похожими» пользователями. Таким образом, из всего множества фильтруемых элементов пользователь получает только некоторый набор, потенциально более ценный для него.

По мере распространения систем совместной фильтрации информации, по мере роста числа пользователей, вовлеченных в них, и по мере увеличения объема обрабатываемой ими информации все больше начнут проявляться качественные аспекты функционирования, в частности, ограничения таких систем, и все более важными будут вопросы их организации для наиболее эффективного функционирования.

Одним из таких вопросов является устойчивость системы совместной фильтрации к разрушающим воздействиям со стороны «плохих» пользователей. Предлагаемый в данной работе формальный аппарат доказательства тех или иных аспектов устойчивости систем совместной фильтрации позволит строить системы совместной с качеством более высоким уровнем защищенности от злонамеренных действий других пользователей.

Вопросы безопасности в существующих системах совместной фильтрации. Общее описание известных подходов к системам рекомендаций было приведено автором в работе [1].

Исследования проводились в рамках Федеральной целевой программы «Интеграция науки и высшего образования РФ на 2002–2006 гг.» по проекту «Фундаментальные исследования и новые технологии проектирования сложных технических систем» и частично поддержаны грантом РФФИ 03-07-90242 «Интернет-комплекс поддержки выполнения проектов фундаментальных исследований сложных систем с применением интеллектуальных технологий на базе экспертных систем» (2003–2005 гг.).