

A. N. Rukavitsyn

Saint Petersburg Electrotechnical University «LETI»

## THE DEVELOPMENT OF WEB PAGE CLASSIFICATION MODEL BASED ON DATA MINING TECHNIQUES

*Describes the development of web page classification model using data mining techniques. The model allows to make multi-label soft classification of the web pages. For developing of this classification model we used the combination of developing methods with existing methods. The experiments show increasing of classification precision with describing of metrics.*

**Web page classification, data mining, machine learning, text processing**

---

УДК 681.322

В. В. Цехановский, В. Д. Чертовской

Санкт-Петербургский государственный электротехнический  
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

## Программная структура модели процесса оптимального планирования в многоуровневой системе

*Рассмотрена программная реализация математической модели функционирования производства. Модель охватывает бизнес-процесс «Производство» с помощью однородного метода, описывающего процессы планирования и управления средствами динамического линейного программирования. Сформулированы требования к составляющим структуры, по которым выбрана система программных средств.*

**Математическое описание, однородный метод, процесс планирования, требования к структуре, система программных средств**

С появлением фактически нового класса адаптивных систем управления производством с качественно изменяющейся целью в процедуре функционирования было сформировано их математическое системное описание на основе однородного метода [1]–[3]. Под производством понимается совокупность подсистем технико-экономического планирования и оперативного управления основным производством – при подсистемном представлении [1] или бизнес-процесс «Производство» – при процедурном представлении [2], [3].

Ввиду сложности предложенных авторами математических моделей и большого объема данных важное значение приобретает разработка и реализация компьютерных моделей, которые позволяют проводить исследование динамических свойств и проектирование многоуровневых автоматизированных адаптивных систем.

**Постановка задачи.** В настоящее время имеющиеся модели используют [4], как правило, одноуровневые варианты с решением задач «пря-

мого счета». В то же время перспективным с позиций повышения конкурентоспособности предприятия являются многоуровневые адаптивные системы, использующие оптимизационные алгоритмы. Такой класс систем и рассматривается в данной статье.

Сам процесс проектирования процедур функционирования и адаптации может быть представлен схемой переходов

$M(0a) \rightarrow M(0) \rightarrow M(1a) \rightarrow M(1) \rightarrow M(2) \rightarrow M(3)$ .

Дело в том, что простота описания в многоуровневых системах противоречит необходимости учета ее важнейших координат и связей. Следствием являются различные уровни математической абстракции описания системы, определяемые целью системы и не зависящие от уровня иерархии управления: теоретико-множественный  $M(0a)$ ,  $M(0)$ ; абстрактно-алгебраический  $M(1a)$ ; топологический  $M(1)$ ; структурный  $M(2)$ ; параметрический  $M(3)$ .

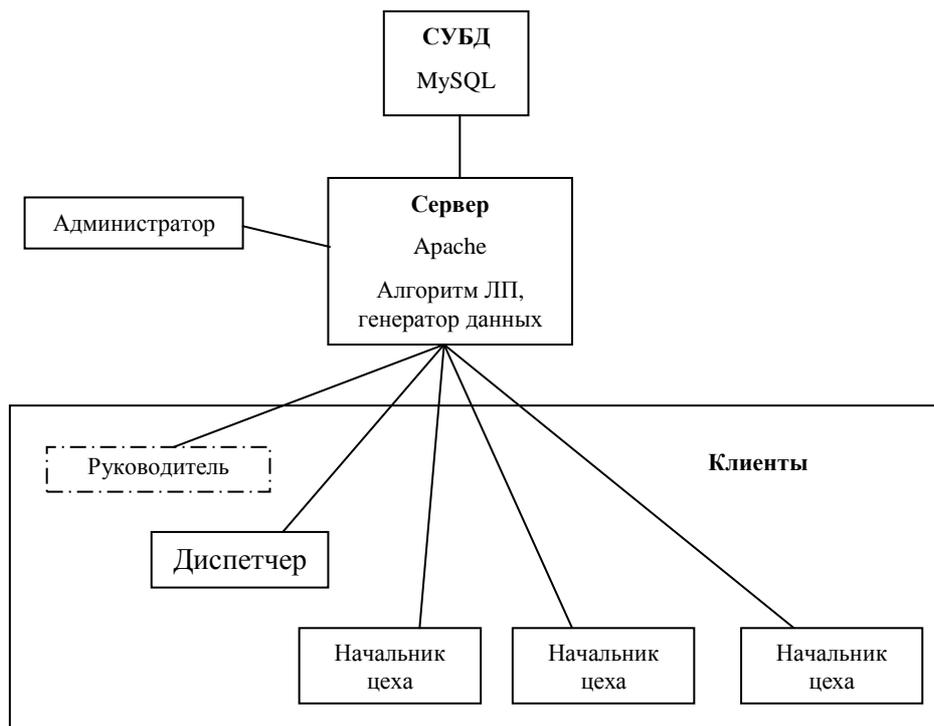


Рис. 1

В силу сложности математического описания и анализа такой структуры с содержательной и методической точек зрения удобнее эту структуру представить в виде компьютерной модели производства (рис. 1).

Выделяют две группы структурных элементов: администратор (в составе сервера), руководитель, диспетчер, начальники цехов (в качестве клиентов).

Режимы планирования: стационарный традиционный (нет новой продукции) и адаптивный (переход на выпуск новой продукции) переход и согласование интересов.

Модель работает следующим образом. Администратор осуществляет генерацию (C++) и ввод разных вариантов числовых данных. Затем руководитель, диспетчер и начальники цехов проводят различные варианты планирования в соответствии с принятыми алгоритмами.

Можно выделить две разновидности модели: для небольшой размерности системы планирования; для высокой размерности, характерной для реальных систем.

Первая разновидность подробно описана в работах [1]–[3], поэтому здесь рассмотрим вторую разновидность.

Основной противоречивой особенностью рассматриваемой компьютерной модели является необходимость хранения большого объема данных при развитой системе алгоритмов. Реализо-

вать изложенное в одном программном продукте не представляется возможным.

Необходимо, следовательно, интегрировать в систему несколько программных продуктов [5]–[8] в режиме «клиент-сервер» с применением «тонкого» клиента. Клиенты при этом разнесены территориально. Для решения такой задачи необходим системный анализ.

Первичное системное рассмотрение связано с исследованием двух случаев: без использования Интернета; с применением Интернета.

Второй случай требует особо тщательного рассмотрения вопросов защиты данных, поэтому здесь ограничимся первым случаем.

В работе [5] предлагаются следующие варианты интеграции:

1. MySQL – Apache– php.
2. MySQL - ADO.NET.
3. Java – ASP.NET.
4. SQL Server – ADO.NET.

Наиболее удобной формой хранения данных для разработчика и пользователя является база данных. В связи с этим следует остановиться на первом и последнем вариантах. В то же время целесообразно оценить возможности системы, состоящей из СУБД, сервера, средств передачи данных, клиентов. Общие требования к программным продуктам в такой интегрированной системе представлены в табл. 1.

Таблица 1

СУБД	Сервер	Язык программирования
Гибкость доступа, простота ввода-вывода. Простое создание системы данных. Хороший интерфейс. Режим «клиент-сервер»	Простота реализации и доступа. Наглядность (интерфейс). Режим «клиент-сервер». Учет связей элементов, связей с другими подсистемами и системами. Связь с СУБД. Распространенность используемого программного пакета. Блочная реализация с применением внутренних интерфейсов, делегатов	Возможность оптимизации процессов планирования и управления. Наличие программы статического линейного программирования (СЛП). Генерация задач СЛП. Модульность и возможность расширения. Реализация разностных уравнений. Учет циклов. Возможность работы с СУБД. Потребность оперативной апробации системы алгоритмов

Целесообразно оценить возможности основных сетевых СУБД: Microsoft SQL Server, Oracle, Firebird, PostgreSQL, InterBase, MySQL.

Все перечисленные СУБД удовлетворяют требованиям табл. 1, поэтому сравнительную оценку следует проводить по остальным требованиям.

Таблица 2

Требования	СУБД					
	SQL Server	Oracle	Firebird	PostgreSQL	Inter-Base	MySQL
Надежность	Да	Да		Да	Да	Да
Производительность	Высокая	Высокая	Высокая	Высокая	Высокая	Самая высокая
Поддержка производителем	Да	Да	Да	Да	Да	Да
Бесплатность	Да	Нет	Нет		Да	Да
Размерность высокая	Единицы гигабайт	Единицы гигабайт	Единицы гигабайт	Единицы гигабайт	Единицы гигабайт	Единицы гигабайт
Широкая распространенность	–	–	Слабая распространенность	Наиболее продвинутая	–	Самая популярная и распространенная
Масштабируемость (работа с большими объемами данных)	Да	Да	Да	Да	Да	Да
Режим «клиент-сервер»	Да	Да	Да	Да	Да	Да
Расширяемость	Да	Да	Да	Да	Да	Да
Трудность обслуживания	–	–	Да	Да	–	–
Гибкость доступа, простота ввода-вывода	–	–	–	Да	Да	Да
Простое создание системы данных	–	–	–	Да	Да	Да
Хороший интерфейс	–	–	–	Да	Да	Да
Наличие инструкций пользователям	–	–	–	–	–	Большое количество методической информации
Безопасность (большое количество функций)	–	–	–	Да	–	Да
Работа с веб-сайтами и веб-приложениями	–	–	–	Да	–	Да
Дата разработки СУБД	Давно	Давно	Давно	Давно	Давно	Недавно

Таблица 3

Требования	Серверы	
	Apache	IIS
Интеграция с Windows	Да	Да
Простота реализации и доступа	Да	Да
Наглядность (интерфейс)	Да	Да
Режим «клиент-сервер»	Да	Да
Учет связей элементов, связей с другими подсистемами и системами	Да	Да
Связь с СУБД	Да	Сложно устанавливаемая
Распространенность используемого программного пакета	Широко распространен	Менее распространен
Блочная реализация с применением внутренних интерфейсов, делегатов	Да	Ограничена
Открытый исходный код	Да	–
Кроссплатформенность (установка на любой ОС и на любом «железе»)	Да	–
Простая расширяемость (огромное число стандартных библиотек)	Да	–
Простота начальной установки и настройки пользователем	Да	–

Эти требования, совместно с ранее упомянутыми, приведены в табл. 2.

В соответствии с табл. 2 следует отдать некоторое предпочтение СУБД MySQL (Denver), к тому же более современной и тщательно отлаженной.

Теперь необходимо оценить серверы, особенно с позиции интеграции с MySQL. Приведем их сравнительные характеристики (табл. 3).

Имеется широкий спектр серверов: [https://ru.wikipedia.org/wiki/Apache\\_HTTP\\_Server](https://ru.wikipedia.org/wiki/Apache_HTTP_Server), Ascet HTTPd, Cherokee HTTP Server, Jetty, Sambar Server, FreeProxy. Требование работы в рамках Windows сразу ограничивает этот спектр первыми двумя видами серверов.

Первые 5 требований удовлетворяются в обоих серверах. Требования 6–8 лучше реализуются в сервере Apache. Остальные требования удовлетворяются в полной мере только в сервере Apache.

Из табл. 3, таким образом, следует, что наиболее подходящим является сервер Apache.

Сетевая связь программных продуктов осуществляется, как правило, протоколом HTTP.

На роль языков программирования алгоритмической части программной системы претендуют языки JavaScript, C#, html, xml, PHP, C++, Java. В наибольшей мере требованиям табл. 1 удовлетворяют PHP, C++, Java.

PHP применяется для Web-приложений, тогда как Java, C++ – для создания не только Web-приложений, но и крупных клиент-серверных приложений, требующих более жесткой структуры. В то же время язык Java обладает большими возможностями в сравнении с C++.

Таким образом, наиболее предпочтительной в рамках поставленных требований является программная система MySQL – Apache – HTTP, Java.

Реализация программной системы с применением Интернета [9], [10] требует дополнительно уделить особое внимание защите данных, что является самостоятельной задачей.

## СПИСОК ЛИТЕРАТУРЫ

1. Чертовской В. Д. Интеллектуализация автоматизированного управления производством. СПб.: Изд-во С.-Петерб. ун-та, 2007.
2. Советов Б. Я., Чертовской В. Д. Автоматизированное адаптивное управление производством. СПб.: Лань, 2002.
3. Советов Б. Я., Цехановский В. В., Чертовской В. Д. Адаптивные автоматизированные системы управления производством. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2013.
4. Габец А. П., Гончаров Д. И. Профессиональная разработка в системе 1С: Предприятие. М.: Питер, 2007.
5. Кузнецов М. Д., Симдянов И. В. MySQL5. СПб.: БХВ-Петербург, 2010.
6. Троелсон Э. Язык программирования C# и платформа Net4.5. М.: Вильямс, 2013.
7. Мэйо Д. Самоучитель Microsoft Visual Studio. СПб.: БХВ-Петербург, 2011.
8. Бей И. Взаимодействие разноразличных программ. М.: Вильямс, 2005.
9. Дюбуа П. MySQL / пер. с англ. М.: Вильямс, 2001.
10. Васвани. MySQL. Использование и администрирование. СПб.: Питер-Юг, 2011.

V. V. Tsehanovsky, V. D. Chertovskoy  
Saint Petersburg Electrotechnical University «LETI»

## PROGRAM MODEL STRUCTURE OF OPTIMAL PLANNING PROCESS IN MULTILEVEL SYSTEMS

*Program realization of mathematical large dimensions model for manufacturing function was considered. The model describes business-process «manufacturing» with help homogenous method. Choice of program means system using suggested requirements was performed.*

**Mathematical description, homogenous method, planning process, requirements to structure, program means system**

---

УДК 681.3.06 (075.8)

А. В. Смирнов, К. А. Борисенко, Е. С. Новикова, А. В. Шоров, И. В. Петухов  
Санкт-Петербургский государственный электротехнический  
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

## Модуль обработки сетевых данных для обнаружения инфраструктурных атак в облачной вычислительной среде OpenStack

*Представлены результаты проектирования модуля сбора и подготовки исходных данных компонента защиты облачной инфраструктуры от DDoS-атак. Разработанный модуль способен обрабатывать как внешний, так и внутренний относительно облачной инфраструктуры сетевой трафик. Это позволяет компоненту защиты обнаруживать и реагировать на DDoS-атаки, источником которых может быть как внешний узел, так и узел, входящий в состав облачной инфраструктуры. Приводятся оценки производительности модуля, показывающие, что он способен обрабатывать сетевой трафик, превышающий по мощности современные DDoS-атаки.*

**Безопасность облачных вычислений, инфраструктурные атаки, архитектура компонента защиты облачной инфраструктуры, сбор и подготовка исходных данных, нагрузочное тестирование, система облачных сервисов OpenStack**

В настоящее время рынок облачных систем активно развивается; облачные технологии предлагают гибкие решения как для бизнеса, так и для частных пользователей, что обеспечивает постоянный рост спроса на них. При использовании облачных сервисов одним из важных моментов является их доступность пользователям. Согласно последним исследованиям, атаки типа «распределенный отказ в обслуживании» (Distributed Denial of Service – DDoS) входят в пятерку наиболее серьезных угроз информационной безопасности облачным сервисам [1]. Мощные DDoS-атаки влияют на работоспособность веб-сайтов государственных учреждений разных стран, ведущих IT-компаний, включая Amazon, Google, Yahoo,

Microsoft и т. д. Большая часть современных систем защиты от DDoS-атак рассматривает облачную систему как единое целое, при этом не учитывается внутреннее устройство облака [2]. Такой подход обеспечивает универсальность системы защиты и ее независимость от реализации конкретной инфраструктуры облачных вычислений. Очевидным недостатком таких систем является невозможность детектирования инфраструктурных атак, источник и цель которых расположены внутри самого облака. Внутренние атаки, источником которых являются непосредственно «облачные» узлы, могут быть обнаружены установкой на каждую виртуальную машину специального программного обеспечения, выполняющегося в фоновом режиме и