

исключить человеческий фактор из процесса локализации отклонений в ходе технологического процесса.

Здесь может помочь информационная система, которая в числе прочего, должна выполнять следующие функции:

1. Сбор статистики о результатах прохождения пластины по маршруту. Результаты всегда видны на главном экране сотрудника, ответственного за качество продукции в виде таблиц или диаграмм. Там он может увидеть, когда начался спад процента выхода годных пластин, после чего следует детальный разбор процессов, проводившихся в это время.

2. Мгновенное оповещение о результатах постоперационного контроля всех заинтересованных лиц. Это гарантирует, что любое отклонение результата от ожидаемого не останется незамеченным, а также поможет точнее и быстрее локализовать проблему.

После локализации проблемы, все последующие производственные процессы приостанавливаются до ее решения.

Затем проводится мозговой штурм с использованием диаграммы Исикавы, где снова используется такая особенность информационной системы, как возможность централизованного хранения информации о том, как пластины проходили по маршруту ранее, с какими параметрами проводились различные технологические процессы ранее и к какому проценту выхода годных изделий это приводило.

После мозгового штурма, в случае успешного решения проблемы, производственные процессы запускаются с соответствующими корректировками.

ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «УЧЕБНЫЕ ПОТОКИ ТПУ»

Отц А.С., Кузнецов Д.Ю.

Томский политехнический университет
634050, Россия, г. Томск, пр-т Ленина, 30
E-mail: eld0727@mail.ru

Введение

В современном мире сеть Интернет стала самой распространённой сферой в жизни общества. Предприятия оказались не исключением из этой тенденции, и на сегодняшний день, пожалуй, все средние и крупные компании используют сеть для автоматизации деятельности и документооборота. На смену тяжеловесным и зачастую платформозависимым интернет приложениям пришли веб сервисы. Их преимущество заключается в том, что они могут использоваться на любой операционной системе, на которой есть поддержка графического интерфейса и Интернет браузер, что делает такое приложение абсолютно платформо-независимым. Помимо получения информации из удаленного источника, веб сервисы могут обеспечивать выполнение некоторых бизнес процессов, что позво-

ми.

Заключение

Согласно закону Мура, который действует по сей день, в полупроводниковом производстве в настоящее время наблюдается рост количества технологических операций, по которым проходит пластина, рост их сложности, уменьшение топологических размеров, что, в свою очередь, приводит к ужесточению требований качества для этих операций. Так как на качество изделия влияет все больше и больше факторов, со временем становится сложнее учитывать хотя бы большую часть из них, не говоря уже обо всех. Но если не учитывать причины, оказывающие влияние на процент выхода годных изделий, невозможно обеспечить надлежащий уровень прибыли, необходимый для поддержания производства в активном режиме. Поэтому, такие инструменты оперативного управления, как диаграмма причин и результатов, становятся незаменимы в современном полупроводниковом производстве.

Литература

1. Ishikawa, Kaoru / Introduction to Quality Control (1 ed.). – Tokyo: 3A Corp, p. 98.
2. Стукач О.В. Программный комплекс Statistica в решении задач управления качеством : учебное пособие для вузов / Национальный исследовательский Томский политехнический университет (ТПУ). Томск: Изд-во ТПУ, 2011. – 163 с.: ил. – Библиогр.: с. 158-159.
3. Статистические методы повышения качества / Под ред. Х. Кумэ. – Пер. с англ. – М. : Финансы и статистика, 1990. – 304 с.

ляет существенно сократить время, а как следствие и финансовые затраты, на выполнение этих процессов.

Все веб сервисы строятся по клиент-серверному шаблону разработки. В качестве сервера выступает приложение, расположенное на некоторой удаленной машине и принимающее запросы от клиентов с дальнейшим формированием ответа. Клиентская часть реализуется в виде веб страниц на языке гипертекстовой разметки HTML. Также клиентские веб приложения делятся на два вида:

- со статическим содержанием страницы, требующим постоянного обновления всей страницы;

- с динамическим содержанием страницы, содержание которого обновляется при помощи технологии AJAX.

Модель проектирования

MVC – это схема использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента. При этом модификация одного из компонентов оказывает минимальное воздействие на остальные. Этими компонентами являются:

- модель – слой, содержащий некоторые данные или знания. А также способ взаимодействия с этими данными и модель их изменения;
- представление – слой, который представляет собой графический интерфейс пользователя, либо данные, которые можно интерпретировать для показа;
- контроллер – слой, в котором реализованы все реакции на действия пользователя. Он является связующим между программой и интерфейсом. В этом слое решается, как поступить с полученной информацией, и в какую модель ее передать.

Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области. На рисунок 1 показана стандартная схема приложения построенного по модели MVC.

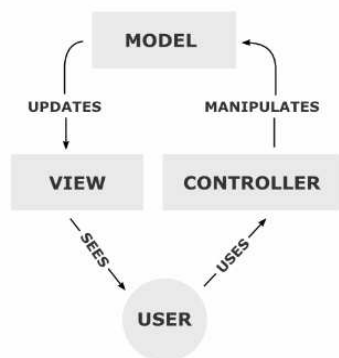


Рис. 1. Стандартная схема MVC приложения

Разработка приложения

Стоит упомянуть, что эта система будет располагаться в среде, где уже имеется великое множество приложений, так что большая часть информации и сущностей у нас уже имеется. Для того чтобы создать схемы баз данных, нужно сформулировать, какие сущности будут присутствовать в нашей системе. Самая очевидная сущность это «поток». Очевидно, что потоки будут группироваться по основным параметрам, а именно: год обучения, тип семестра, учебное подразделение, форма обучения и курс. Также для формирования учебных планов в потоки прописаны четкие требования. Нужно чтобы совпадали:

1. Названия дисциплины.
2. Вид занятия.
3. Количество аудиторных часов в неделю по виду занятия.
4. Обеспечивающей кафедры по дисциплине.
5. Стратегии распределения занятий (все до смены расписания, больше до смены расписания, все после смены расписания, меньше до смены расписания, равномерно по семестру).
6. Номера семестра, в котором изучается дисциплина.
7. Вида отчетности.
8. Количество кредитов.
9. Длительности семестра.
10. Количество семестров изучения.

Первые четыре пункта являются обязательными для формирования потока, следовательно, они должны быть также добавлены в сущность, так как они будут определяющими. Остальные пункты не являются основными, и могут быть получены их других, уже существующих, таблиц и поэтому не обязательны. Исходя из всего выше изложенного, можно создать таблицу сущности поток как показано на рисунок 2.

P_UCHEBNIY_POTOK (POUP)	
#	ID
*	DISCIPLINA_ID
*	VID_ZANYATIYA_ID
*	AUD_CHASOV
*	PODRAZDELENIE_ID
*	PREC_PODR_ID
*	GOD
*	NOMER_CURSA
*	SEMESTER_ID
*	FORMA_OBUCHENIYA_ID

Рис. 2. Таблица сущности поток

В целом же было выделено три сущности: учебный поток, информация об учебных планах потока и дополнительные учебные планы. Дополнительные учебные планы нужны, чтобы для конкретных параметров хранить учебные планы которые будут участвовать в генерации учебных потоков. Вся схема показана на рисунок 3.

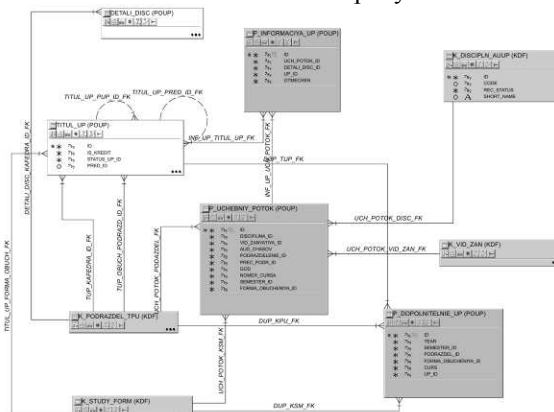


Рис. 3. Схема учебные потоки ТПУ

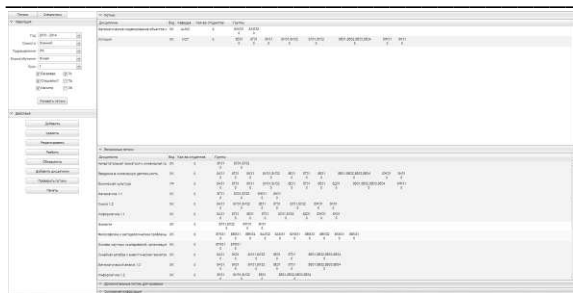


Рис. 4. Пользовательский интерфейс

Для разработки серверной части самого приложения использовались язык программирования Scala, MVC-фреймворк play 2.0, Thrift protocol и Scrooge генератор для преобразования thrift в Scala сервисы. В качестве IDE использовался Eclipse Indigo. Для создания клиентской части применялся Smart Client. Это JavaScript фреймворк для создания RIA (Rich Internet Application, богатое веб приложение), который позволяет писать пользовательский веб интерфейс, не прибегая к верстке на HTML и использую привычный для большинства программистов объектно-ориентированный стиль программирования. Также для удобства написания JavaScript'a использовался Coffee Script, который позволяет писать код в функциональном стиле, что было удобно учитывая, что Scala функциональный язык программирования. Также Smart

Client имеет в своем наборе большое количество тем, в том числе и приспособленную для мобильных версий, что в дальнейшем позволит с легкостью сделать мобильное приложение без больших затрат. И в конечном итоге получился следующий интерфейс:

Заключение

Использование веб приложений для автоматизации бизнес процессов является одним из самых удобных способов. Так как он обеспечивает доступ к сервисам с любого устройства, в котором имеется веб браузер и доступ к интернету, и при этом не нужно задумываться о реализации под конкретную платформу.

Литература

1. Playframework 2.0 documentation [Электронный ресурс]. – Режим доступа: <http://www.playframework.com/>, свободный.
2. Model-View-Controller // Википедия [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/wiki/Model-View-Controller>, свободный.
3. Official page SmartClient [Электронный ресурс]. – Режим доступа: <http://smartclient.com/>, свободный.

АВТОМАТИЗАЦИЯ РАБОТЫ С ЗАЯВКАМИ НА БИРЖЕ

Петрова Г.Г.

Томский политехнический университет
634050, Россия, г. Томск, пр-т Ленина, 30
E-mail: ggp_pgg@mail.ru

Введение

Быстрое и правильное выставление заявок на покупку или продажу ценных бумаг является одним из самых важных моментов в торговле на биржевом рынке. Трейдеры, торгующие вручную, иногда могут ошибаться как в расчетах, так и при выставлении заявок. Автоматизация данного процесса позволит сделать процедуру работы с заявками быстрой, производительной и практически безошибочной.

Можно использовать торгового (биржевого) робота для осуществления торгов на бирже или использовать программный комплекс, заменяющий трейдера, но полностью контролируемый и управляемый им.

Биржевой робот – это полноценные программы, которые торгуют вместо человека. Сегодня существует несколько видов таких программ.

Есть торговые роботы, которых чаще всего называют советниками, они выдают рекомендации – когда нужно покупать, а когда продавать ту или иную ценную бумагу. Основное преимущество такого советника перед человеком – возможность анализировать большой объем информации в короткий срок, что довольно затруднительно для

человека. Анализ производится по тем параметрам, которые были заданы разработчиками советников. Рекомендации выдаются в виде сообщения или какого-либо звукового сигнала.

Второй вид торговых роботов – это полноценные роботы, которые полностью берут на себя и риск-менеджмент (т.е. контролируют возможные риски и убытки) и управление ценными бумагами. Торговые стратегии программы-роботы используют разнообразнейшие.

Эффективность от использования робота для торговли акциями в среднем составляет 30...40% годовых.

В кризисное время, когда рынок ведет себя совершенно непредсказуемо – вне логики и здравого смысла – некоторые торговые роботы показали себя очень эффективно. В основном, это были низкорисковые роботы.

Существуют роботы-скальперы, работающие по другим алгоритмам (они нацелены на получение большой прибыли в достаточно короткое время) – они могут сделать порядка 1000% годовых. Естественно, с ростом доходности возрастает и риск. То есть вы можете запустить сто скальперов,