

сит не только способность расширяемости и удобство в использовании среды, но и такие параметры как *надежность* и *производительность*.

Пользуясь общими принципами построения систем управления, можно выделить следующие модульные составляющие такой системы:

- программные модули, представляющие версии вычислительного процесса;
- модуль сравнения двух результатов;
- файл проекта;
- исполнительное устройство (ИУ);
- датчики;
- среда мультиверсионного исполнения.

Рассмотрев содержание структуры, перейдем к более сложному вопросу: определение типов связей между модулями системы. Попробуем определить, в каких модулях может произойти непредвиденный отказ, последствия которого нужно предотвратить. Файл проекта, загружается в начале работы, и сохраняется при завершении — отказу тут произойти негде. Отказ ИУ и датчиков в любом случае приведет к отказу всей системы. Модуль сравнения двух результатов содержит довольно простой алгоритм и просто хорошим тестированием этого модуля можно свести вероятность отказа в нем к нулю. Следовательно, связь этих модулей со средой исполнения можно реализовать через статические или динамические библиотеки. Таким образом, единственное «слабое звено» нашей системы – это версии, так как в них производятся сложные вычислительные операции. Здесь возможны несколько вариантов взаимодействия со средой (представлены, по порядку возрастания потребности в памяти).

- Статическая компоновка: из исходных кодов, либо с использованием статических библиотек
- Компоновка с использованием динамических библиотек
- Компоненты выполнены в отдельных исполняемых модулях

Для запуска проекта необходимо запустить все эти модули. Делать это вручную и нецелесообразно, так как это ставит в зависимость производительность среды от оператора. Поэтому следует создать специальные программы-агенты, следящие за запуском всех модулей и перезапускающие их при необходимости. При использовании такой модели для исполнения всех моделей на одном компьютере, агентов можно встроить в саму среду исполнения. А при сетевом распределении модулей встроить агентов в среду очень сложно, так как возникнут трудности с отслеживанием состояния процесса на удаленной машине и его удаленного перезапуска.

Данную модель можно разделить на две, по виду взаимодействия между исполняемыми модулями:

А) через общие участки памяти (например, файлы)

В) через сетевой протокол

С учетом разнообразия моделей можно сделать вывод о нереальности создания действительно универсальной среды мультиверсионного исполнения программных модулей, в частности, из-за противоре-

чивости требований платформенной независимости и максимальной надежности.

Подводя итог, хотелось бы резюмировать ряд требований к исходному коду для того, чтобы приблизить конкретную реализацию среды исполнения к универсальной. А именно:

- использовать открытый исходный код
- все обращения к функциям ОС вынести в отдельный модуль, и сделать его функционирование прозрачным для СМВИ
- использовать язык С или С++

АНАЛИЗ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ЭЛЕКТРОННЫМ ДОКУМЕНТООБОРОТОМ СЕРВИСНЫХ ЦЕНТРОВ

Морозов С.А., Манжула В.Г.

Южно-Российский государственный университет экономики и сервиса, Шахты

На сегодняшний день процесс автоматизации документооборота предприятий является одной из актуальных проблем в сфере информационных технологий. Практически любое предприятие имеет особенности, которые отличают его от других аналогичных предприятий. Эти особенности не позволяют создать универсальную комплексную систему, которая подошла бы всем предприятиям.

Массовое появление на российском рынке офисной техники зарубежных производителей повлекло за собой увеличение спроса на ее гарантийное и послегарантийное обслуживание. Сложившееся положение вещей повлекло рынок офисной техники в серьезную конкурентную борьбу – борьбу за потребителя. Очень важную роль в этом процессе играет отбор сервисных центров для создания сервисной сети. Поэтому сервисные компании проходят жесточайший контроль, и одним из критериев оценки является наличие у сервиса автоматизированной системы управления.

Актуальная проблема, связанная с процессом автоматизации сервисных центров – выбор программных средств позволяющих в кратчайшие сроки и с минимальной потерей финансовых средств достигнуть цели автоматизации производства.

Какие программные продукты существуют в мировом информационном сообществе? Какие из них можно применить для автоматизации сервисных центров российского рынка сервисных услуг?

Все продукты для автоматизации сервисных центров, представленные на рынке можно условно разделить на 4 группы:

- 1) зарубежные системы уровня CRM-систем;
- 2) разработки российских фирм (исключая 1С);
- 3) разработки на базе 1С;
- 4) специализированные разработки.

Для оценки программных средств возьмем такие показатели, как функциональность, стоимость поставки, уровень сервиса и стоимость владения.

Системы зарубежного производства имеют средства адаптации под нужды конкретного заказчика (встроенные языки программирования бизнес-логики, генераторы отчетов, генераторы структур информационных объектов). Несмотря на высокую стоимость приобретения таких систем, стоимость их внедрения и эксплуатации гораздо выше. Поэтому заказчики очень часто останавливаются на половине пути, поняв всю бесперспективность выбранного решения. И только очень немногие доходят до логического конца, затратив огромные усилия на доводку системы, обучение персонала и реинжиниринг своей компании.

Поэтому, зарубежные системы, считаем слабо применимыми в российских условиях по целому ряду причин: высокая стоимость, сложности связанные с внедрением и сопровождением, проблемы русификации и учета российской специфики.

Несмотря на то, что продукты второй группы гораздо дешевле зарубежных, отсутствие поддержки и постоянного развития продукта, а также во многом моральное устаревание, не позволяет использовать их в качестве основы эффективного управления.

Третья группа, программы, разработанные на платформе 1С, наиболее привлекает. Платформа 1С обладает достаточной гибкостью для реализации практически любых пожеланий пользователя. На базе 1С написаны продукты для автоматизации большинства отраслей промышленности, сельского хозяйства, торговли и сферы услуг. С качеством услуги этих программ так же нет замечаний. Практически в каждом городе существует фирма, занимающаяся поддержкой программ 1С, а так же сама фирма 1С регулярно выпускает обновления своих продуктов. Стоимость приобретения и сопровождения систем на базе 1С ненамного больше стоимости аналогичных российских систем и гораздо меньше (в 10-ки раз) стоимости зарубежных продуктов.

Специализированные разработки, относящиеся к четвертой группе, имеют так же свои плюсы и минусы. Из отрицательных сторон можно выделить следующие замечания. Написанный программный продукт не имеет связи с другими платформами. Так же существует зависимость от разработчика программного продукта. Отсутствия обновлений, все доработки программного продукта производятся за отдельную плату. Из положительных сторон это, прежде всего низкая стоимость программного продукта и четкая заточенность программы под определенную организацию.

Таким образом, для авторизации сервисного центра лучше всего подойдет программы, разработанные на платформе 1С. У этих программ существует гибкость, что очень важно при автоматизации. Так же программы, относящиеся к этой группе, имеют высокое качество сервиса. Немаловажным плюсом является и регулярный выпуск обновлений своих продуктов. Ну и конечно, по ценовой характеристике она входит в диапазон, наиболее подходящий для российского рынка.

Так же можно сделать вывод, что специализированные разработки, так же можно использовать для автоматизации документооборота сервисного центра. Для этого Вам необходимо включить в рабочий со-

став сотрудника, который, напишет Вам данный программный продукт, и будет поддерживать его в рабочем состоянии.

СПИСОК ЛИТЕРАТУРЫ

1. Баранов В.В. Автоматизация управления предприятий. Инфра – М, 2000.
2. Петров Ю.А. Комплексная автоматизация управления предприятием. Информационные технологии – теория и практика. М.: 2001.
3. Шуремов Е.Л. Информационные технологии управления взаимоотношениями с клиентами. 1С Пабблишинг, 2001.

КОМПОНЕНТНАЯ ПРОГРАММНАЯ АРХИТЕКТУРА МУЛЬТИВЕРСИОННЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Поздняков Д.А.

Аннотация

В статье предложены методы компонентного программирования для построения мультиверсионного программного обеспечения отказоустойчивых систем управления и обработки информации.

Основное содержание

На сегодняшний день разработаны различные методы проектирования отказоустойчивого программного обеспечения. Среди них одним из наиболее перспективных является метод мультиверсионного проектирования. Он состоит в том, что в систему включаются несколько программных модулей, дублирующих друг друга по своему целевому назначению.

Мультиверсионное программирование предполагает, что большинство модулей, реализующих ту же самую функциональность, производят сбой в различных точках, так что сбои могут быть обнаружены и исправлены. Независимость сбоев различных модулей может быть достигнуто с помощью ввода разнообразия в процесс разработки. Но эта независимость сбоев находится на уровне исходных кодов – это является ключевой проблемой. На стадии выполнения мультиверсионных модулей независимость сбоев теряется из-за того, что остались не учтенными возможные взаимодействия модулей в рамках всей программной системы. Модули работают в едином адресном пространстве памяти, разделяют одни и те же ресурсы операционной системы, и из-за этого возникают дополнительные зависимости между модулями программного обеспечения. Вследствие этого, сбой одного модуля может привести к сбою других или даже к отказу всей системы в целом. Для решения этой проблемы необходимо сохранить введенное разнообразие при разработке модулей и перенести его на стадию выполнения. Для этого были сформулированы и обоснованы требования, которым должны соответствовать разрабатываемые мультиверсионные модули.

1. Динамическое подключение модулей, цель которого – реализовать возможность замены модулей во время работы программной системы.
2. Требование инкапсуляции следует из требования динамического подключения модулей. Модули