**МИНИСТЕРСТВО ОБОРОНЫ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ВОЕННО-МОРСКОЙ ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ**

**им. А. С. Попова**

Филиал Федерального Государственного военного образовательного учреждения

высшего профессионального образования

«Военный учебно-научный центр Военно-Морского Флота «Военно-морская академия

имени Адмирала Флота Советского Союза Н. Г. Кузнецова»

**Кафедра боевого применения АСУ**

**КУРСОВАЯ РАБОТА**

 по дисциплине:

**«ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ»**

на тему: **«Разработка информационно-справочной системы администратора ЛВС ».**

**Выполнил:** курсант 246 класса Редин Н.А.

**Проверил:** капитан-лейтенант Скородумов Д.Н.

**Петродворец**

**2010**

**СОДЕРЖАНИЕ:**

[**ВВЕДЕНИЕ**](#_Toc263609038)……………………………………………………………...…...3

 **1.Функциональная модель**……………………………………….…...........9

**2.ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

 **2.1. Введение**.….. .….. .….. .….. .….. .….. .….. .….. .….. .….. .….. .….. .….. 13

**2.2. Основания для разработки**…..…..…..…..…..…..…..…..…..…..….. 13

**2.3. Назначение разработки**..… ..… ..… ..… ..… ..… ..…..…..…..…..…..13

 **2.4. Требования к программе или программному изделию**…..…14

**2.4.1. Требования к функциональным характеристикам**…..…….14

 **2.4.2.** **Требования к надежности**…..…..…..…..…..…..…..…..…..…..…..14

 **2.4.3. Условия эксплуатации** ..…..… ..…..… . .…..… ..…..… ..…..… ..…15

 **2.4.4. Требования к параметрам технических средств**…..………...15

**2.4.5. Требования к информационной и программной совместимости**.……………………………………………………………….…15

**2.4.6. Требования к маркировке и упаковке**………………………..…15

**2.4.7. Требования к транспортированию и хранению**……………...16

**3. Требования к программной документации**……………………….16

**4. Стадии и этапы разработки**…………………………………………..…16

**4.1 Стадии разработки**………………………………………………………..16

 **4.2 Этапы разработки**………………………………………………………...16

 **4.3 Содержание работ по этапам**…………………………………………..17

 **4.4 Календарный план**……………………………………………………..…18

 **5. Порядок контроля и приемки**………………………………………..…19

 **5.1 Виды испытаний**………………………………………………………..…19

 **5.2 Общие требования к приемке работы**……………………………..19

 **6. Предложения по архитектуре программной системы**……….....19

 **7.Информационно-логическая модель базы данных**…………….20

 **8.Заключение**…………………………………………………………………….21

 **9. Список используемой литературы**………………………………….....22

**Введение**

Технология создания крупных информационно-справочных систем (далее - ИСС) предъявляет особые требования к методикам реализации и программным инструментальным средствам, а именно:

Реализацию крупных проектов принято разбивать на стадии анализа (прежде чем создавать ИСС необходимо понять и описать бизнес-логику предметной области), проектирования (необходимо определить модули и архитектуру будущей системы), непосредственного кодирования, тестирования и сопровождения. Известно, что исправление ошибок, допущенных на предыдущей стадии обходится примерно в десять раз дороже, чем на текущей, откуда следует, что наиболее критичными являются первые стадии проекта. Поэтому крайне важно иметь эффективные средства автоматизации ранних этапов реализации проекта.

Крупный проект невозможно реализовать в одиночку. Коллективная работа существенно отличается от индивидуальной, поэтому при реализации крупных проектов необходимо иметь средства координации и управления коллективом разработчиков.

Жизненный цикл создания сложной ИСС сопоставим с ожидаемым временем ее эксплуатации. Другими словами, в современных условиях компании перестраивают свои бизнес - процессы примерно раз в два года, столько же требуется (если работать в традиционной технологии) для создания ИСС. Может оказаться, что к моменту сдачи ИСС она уже никому не нужна, поскольку компания, ее заказавшая, вынуждена перейти на новую технологию работы. Следовательно, для создания крупной ИСС жизненно необходим инструмент значительно (в несколько раз) уменьшающий время разработки ИСС.

Вследствие значительного жизненного цикла может оказаться, что в процессе создания системы внешние условия изменились. Обычно внесение изменений в проект на поздних этапах создании ИСС - весьма трудоемкий и дорогостоящий процесс. Поэтому для успешной реализации крупного проекта необходимо, чтобы инструментальные средства, на которых он реализуются, были достаточно гибкими к изменяющимся требованиям.

На современном рынке средств разработки ИСС достаточно много систем, в той или иной степени удовлетворяющих перечисленным требованиям. Ниже будет рассмотрена вполне конкретная технология разработки, основывающаяся на решениях фирм Logic Works и Rational Software, которая является одной из лучших на сегодняшний день по критерию стоимость/эффективность.

Рис.1. Общая схема взаимодействия инструментальных средств Logic Works и Rational Software.

Для проведения анализа и реорганизации бизнес-процессов Logic Works предлагает CASE - средство верхнего уровня - BPwin, поддерживающий методологии IDEF0 (функциональная модель), IDEF3 (WorkFlow Diagram) и DFD (DataFlow Diagram). Функциональная модель предназначена для описания существующих бизнес-процессов на предприятии (так называемая модель AS-IS) и идеального положения вещей - того, к чему нужно стремиться (модель TO-BE). Методология IDEF0 предписывает построение иерархической системы диаграмм - единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция - система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. После каждого сеанса декомпозиции проводится сеанс экспертизы, каждая диаграмма проверяется экспертами предметной области, представителями заказчика, людьми, непосредственно участвующими в бизнес - процессе. Такая технология создания модели позволяет построить модель адекватную предметной области на всех уровнях абстрагирования. Если в процессе моделирования нужно осветить специфические стороны технологии предприятия, BPwin позволяет переключиться на любой ветви модели на нотацию IDEF3 или DFD и создать смешанную модель. Нотация DFD включает такие понятия как внешняя ссылка и хранилище данных, что делает ее более удобной (по сравнению с IDEF0) для моделирования документооборота. Методология IDEF3 включает элемент “перекресток”, что позволяет описать логику взаимодействия компонентов системы.

На основе модели BPwin’а можно построить модель данных. Для построения модели данных Logic Works предлагает мощный и удобный инструмент - ERwin. Хотя процесс преобразования модели BPwin в модель данных плохо формализуется и поэтому полностью не автоматизирован, Logic Works предлагает удобный инструмент для облегчения построения модели данных на основе функциональной модели - механизм двунаправленной связи BPwin - ERwin (1, рис.1). ERwin имеет два уровня представления модели - логический и физический. На логическом уровне данные представляются безотносительно конкретной СУБД, поэтому могут быть наглядно представлены даже для неспециалистов. Физический уровень данных - это, по - существу, отображение системного каталога, который зависит от конкретной реализации СУБД. ERwin позволяет проводить процессы прямого и обратного проектирования БД (2, рис.1). Это означает, что по модели данных можно сгенерировать схему БД или автоматически создать модель данных на основе информации системного каталога. Кроме того, ERwin позволяет выравнивать модель и содержимое системного каталога после редактирования того, либо другого. ERwin интегрируется с популярными средствами разработки клиентской части - PowerBuilder, SQLWindows, Visual Basic, Delphi (3, рис.1), что позволяет автоматически генерировать код приложения, который готов к компиляции и выполнению (4, рис.1).

Создание современных информационно-справочных систем, основанных на широком использовании распределенных вычислений, объединении традиционных и новейших информационных технологий, требует тесного взаимодействия всех участников проекта: менеджеров, бизнес и системных аналитиков, администраторов баз данных, разработчиков. Для этого использующиеся на разных этапах и разными специалистами средства моделирования и разработки должны быть объединены общей системой организации совместной работы. Фирма Logic Works разработала систему Model Mart - хранилище моделей, к которому открыт доступ для участников проекта создания информационной системы (5, рис.1). Model Mart удовлетворяет всем требованиям, предъявляемым к средствам разработки крупных информационных систем, а именно:

Совместное моделирование. Каждый участник проекта имеет инструмент поиска и доступа к интересующей его модели в любое время. При совместной работе используются три режима: незащищенный, защищенный и режим просмотра. В режиме просмотра запрещается любое изменение моделей. В защищенном режиме модель, с которой работает один пользователь не может быть изменена другими пользователями. В незащищенном режиме пользователи могут работать с общими моделями в реальном масштабе времени. Возникающие при этом конфликты разрешаются при помощи специального модуля - Intelligent Conflict Resolution (ICR). В дополнение к стандартным средствам организации совместной работы Model Mart позволяет сохранять множество версий, снабженных аннотациями, с последующим сравнением предыдущих и новых версий. При необходимости возможен возврат к предыдущим версиям.

Создание библиотек решений. Model Mart позволяет формировать библиотеки стандартных решений, включающие наиболее удачные фрагменты реализованных проектов, накапливать и использовать типовые модели, объединяя их при необходимости “сборки” больших систем. На основе существующих баз данных с помощью ERwin возможно восстановление моделей (обратное проектирование), которые в процессе анализа пригодности их для новой системы могут объединяться с типовыми моделями из библиотек моделей.

Управление доступом. Для каждого участника проекта определяются права доступа, в соответствии с которыми они получают возможность работать только с определенными моделями. Права доступа могут быть определены как для групп, так и для отдельных участников проекта. Роль специалистов, участвующих в различных проектах может меняться, поэтому в Model Mart можно определять и управлять правами доступа участников проекта к библиотекам, моделям и даже к специфическим областям модели.

Как было указано выше, при разработке крупных проектов критичным становится время реализации проекта. Одним из решений проблемы может стать автоматическая генерация кода приложения (клиентской части) CASE - средствами на основе модели предметной области. Хотя ERwin решает эту задачу, код генерируется на основе модели IDEF1X, то есть фактически на основе реляционной модели данных, которая непосредственно не содержит информацию о бизнес - процессах. Как следствие этого, сгенерированный код не может полностью обеспечить функциональность приложения со сложной бизнес-логикой. Существует альтернативная технология кодогенерации, которая лишена этого недостатка - объектно-ориентированное проектирование, реализованное в Rational Rose (Rational Software). Rational Rose – позволяющее строить объектные модели в различных нотациях (OMT, UML, Буч) и генерировать на основе полученной модели приложения на языках программирования C++, Visual Basic, Power Builder, Java, Ada, Smalltalk и др. Поскольку генерация кода реализована на основе знаний предметной области, а не на основе реляционной структуры данных, полученный код более полно отражает бизнес-логику. Rational Rose поддерживает не только прямую генерацию кода, но и обратное проектирование, то есть создание объектной модели по исходному коду приложения (6, рис.1).

Rational Rose предназначен для генерации клиентской части приложения. Для генерации схемы БД объектную модель следует конвертировать в модель данных IDEF1X. Модуль ERwin Translation Wizard (Logic Works) позволяет перегружать объектную модель Rational Rose в модель данных ERwin (и обратно) и, с помощью ERwin, сгенерировать схему БД (7, рис.1). Таким образом, технологическая цепочка Rational Rose - ERwin Translation Wizard - ERwin позволяет реализовывать крупные проекты в технологии клиент - сервер.

**1.ФУНКЦИОНАЛЬНАЯ МОДЕЛЬ**

В качестве инструмента для описания функциональной модели был взят программный продукт BPWin версии 4.1.

На рисунке 2 показано обеспечение деятельности администратора ЛВС. Для работы системы необходимо наличие данных для анализа, т.е. наличие входного потока данных(информация о аппаратных средствах; информация о пользователях; информация о политике безопасности). Выходным ресурсом системы являются результат запроса. В качестве механизма управления выступает администратор ЛВС, а механизмом исполнения являются программные средства, которые выводят на экран нужную информацию для администратора ЛВС.

Получаем контекстную диаграмму верхнего уровня, представленную на рисунке 2.

Рис.2 Контекстная диаграмма верхнего уровня

Для более детального рассмотрения данной системы надо сначала учесть то, что данная система подразумевает под собой программы-оболочки, служащие для управления массивами и базами данных. В наш век

всеобщей компьютеризации информационно-справочные системы значительно облегчают труд человека. Соответственно надо учесть то, что любой результат автоматической обработки данных должен проверятся и только после этого могут быть получены определенные выводы и рекомендации.

Для быстрой и корректной работы всей системы необходимо наличие блока обработки данных. Он позволит упорядочить входной поток данных.

Кроме всего прочего необходима база данных, которая будет собирать и хранить всю информацию. К такой информации относится как входная отсортированная информация, так и информация прошедшая автоматический и экспертный анализ. Управление базой данных должен осуществлять администратор ЛВС.

Для работы системы также необходим блок выработки решений, который на основании требований руководящих документов будет осуществлять процесс выработки необходимой документации. Обеспечивать работу всей системы будет администратор ЛВС.

Из выше сказанного получаем контекстную диаграмму 2 уровня представленную на рисунке 3.

Рис.3 Контекстная диаграмма 2 уровня

Рассмотрим более детально блок формирования отчетных документов. Значения параметров из блок настройки политики безопасности и блока редактирования списка аппаратных средств поступают в блок формирования отчетных документов, основываясь на полученных данных администратор ЛВС проводит экспертную оценку и вырабатывает конкретное решение для выбора нужной ему информации. После принятия решения оно оформляется в виде документа для дальнейшего использования. Помимо этого вся информация такая как:

- учетные записи пользователей;

- типы учетных записей пользователей;

- пароли пользователей;

- данные о используемой политике безопасности;

- данные о используемых аппаратных средствах;

заносится в базу данных для дальнейшего использования.

Контекстная диаграмма блока настройки политики безопасности имеет след вид:

**Рис. 4**

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**2.1. Введение**

Разработать информационно-справочную систему администратора ЛВС, для этого создать базу данных и клиентское приложение.

Основные задачи - организация наглядной информационной системы, ведение базы, поиск данных и выдача справок администратора ЛВС, возможность введения новых данных, а также удаления и редактирования имеющихся данных для администратора.

Данная информационно-справочная система может применяться для быстрого поиска требующих ресурсов.

**2.2. Основания для разработки**

Документом для разработки является бланк задания на выполнение курсовой работы по дисциплине «Технология разработки программного обеспечения», подписанный начальником кафедры БП АСУ капитаном

1 ранга О. Пантиховским \_\_ марта 2010 г. по теме «Разработка информационно-справочной системы администратора ЛВС»

**2.3. Назначение разработки**

Информационно-справочная системы для администратора ЛВС должна:

-осуществлять просмотр информации об аппаратных средствах;

-включать возможность добавления информации, удаления и редактирования;

-возможность просмотра и печати отчетов, содержащих информацию:

- учетные записи пользователей;

- типы учетных записей пользователей;

- пароли пользователей;

- данные о используемой политике безопасности;

- данные о используемых аппаратных средствах;

**2.4. Требования к программе или программному изделию**

**2.4.1. Требования к функциональным характеристикам**

Разрабатываемая информационно-справочная система должна позволять:

1. Программно анализировать поступающие данные, давать выводы по запросу администратора ЛВС.
2. Производить формирование отчетов относительно полученной ранее информации.
3. Осуществлять подготовку выходных документов.
4. Производить идентификацию и аутентификацию пользователей.
5. Редактировать учетные записи пользователей.

Кроме того, эта система должна давать возможность сохранять все входящие и обработанные данные. Время на обработку информации должно быть минимальным.

**2.4.2.** **Требования к надежности**

Система должна быть надежной как по отношению к аппаратно программным сбоям и ошибкам, так и организационно техническим вопросам, связанным с установкой и эксплуатацией системы. Необходимо обеспечить проверку на корректность автоматически обрабатываемой информации. Программное обеспечение системы должно быть надежным по отношению к угрозе от НСД. Все техническое оборудование должно быть продублировано для обеспечения быстрой замены. Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

**2.4.3. Условия эксплуатации**

При сбоях в работе системы, она должна сохранять информацию, путем создания резервной копии.

**2.4.4. Требования к параметрам технических средств**

В состав технических средств должен входить IВМ-совместимый персональный компьютер (ПЭВМ) включающий в себя:

1. Процессор Pentium-300 MHz, не менее;
2. Оперативную память объемом, 32 Мегабайт, не менее;
3. HDD, 2 Гигабайт, не менее;
4. Операционную систему платформы Windows;

6. Microsoft Access 97-2007;

7. Microsoft Word 97-2007.

**2.4.5. Требования к информационной и программной совместимости**

Системные программные средства, используемые программой, должны быть представлены:

-Лицензионной локализованной версией операционной системы платформы -Windows;

-Microsoft Access 97-2007;

-Microsoft Word 97-2007.

**2.4.6. Требования к маркировке и упаковке**

Все технические компоненты должны быть проверены и опломбированы.

**2.4.7. Требования к транспортированию и хранению**

Специальных требований к системе не предъявляется.

3. Требования к программной документации

Предварительный состав программной документации должен включать в себя:

- техническое задание;

- программу и методики испытаний;

- руководство администратора;

4. Стадии и этапы разработки

**4.1 Стадии разработки**

Разработка должна быть проведена в три стадии:

1. Разработка технического задания.

 2. Рабочее проектирование.

3. Внедрение.

**4.2 Этапы разработки**

На стадии разработки технического задания должны быть выполнены следующие этапы:

1. Разработка технического задания.

2. Согласование технического задания.

3. Утверждение технического задания.

На стадии рабочего проектирования должны быть выполнены следующие этапы:

1. Разработка программы.

2. Разработка программной документации.

3. Испытания программы.

На стадии внедрение должны быть выполнены следующие этапы:

1. Подготовка программы.

2. Передача программы.

**4.3 Содержание работ по этапам**

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1.Постановка задачи.

2.Определение и уточнение требований к техническим средствам.

3.Определение требований к программе.

4.Определение стадий, этапов и сроков разработки программы и документации на неё.

5.Согласование и утверждение технического задания.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе тестирования автоматизированной системы должно осуществляться следующим образом:

1. Необходимо проверить точность следования всем алгоритмам.

2. Проверить правильность регистрации пользователей.

3. Проверить наличие у пользователей присвоенных им прав.

4. Необходимо проверить наличие заполненности теоретического курса.

5. Проверить наличие практических заданий.

6. Проверить реакцию системы при вводе некорректных значений.

7. Необходимо проверить корректность добавления, редактирования, удаления данных каждую из таблиц.

8. Проверить возможности поиска необходимых данных.

9. Проверить возможности сортировки необходимых данных.

10. Проверить возможности фильтрации необходимых данных.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

**4.4 Календарный план**

Таблица – календарный план

|  |  |  |  |
| --- | --- | --- | --- |
| Стадии разработки | Этапы работ | Содержание работ | Время выполнения |
| Техническое задание | Постановка задачи. | Построение математической модели и детальное рассмотрение предметной области. | 20.03.2010-20.04.2010 |
| Разработка технического задания | Определение всех частей программы, сроков разработки и определение ее функциональности. | 20.04.2010-20.05.2010 |
| Утверждение технического задания | Согласование и утверждение технического задания. | 20.05.2010-20.06.2010 |
| Разработка проекта. | Проектирование и разработка программы. | Программирование и отладка. | 20.06.2010-20.07.2010 |
| Создание документации. | Разработка программной документации (пользователю и разработчику) в соответствии с предъявленными требованиями. | 20.07.2010-20.08.2010 |
| Тестирование. | Корректировка программы, выявление недочетов.  | 20.08.2010-20.09.2010 |
| Внедрение | Подготовка и сдача программного продукта заказчику. | Сдача проекта заказчику. Оформление соответствующей документации. | 20.09.2010-20.10.2010 |

**5. Порядок контроля и приемки**

**5.1 Виды испытаний**

Приемно-сдаточные испытания должны проводиться на объекте Заказчика в оговоренные сроки.

Приемно-сдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной Заказчиком программы и методик испытаний.

Ход проведения приемно-сдаточных испытаний Заказчик и Исполнитель документируют в протоколе проведения испытаний.

**5.2 Общие требования к приемке работы**

На основании протокола проведения испытаний Исполнитель совместно с Заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

**6. Предложения по архитектуре программной системы**

С учетом специфики разрабатываемой информационной системы была разработана архитектура ИИС администратора ЛВС данной системы (рис. 5). На ней показаны основные блоки и взаимосвязь между ними.

 **Интерфейс для ввода**

 **данных**

 **Информационно-справочная**

 **система администратора ЛВС**

 **БД выходные**

 **Рис.5 докуметы**

**7.Информационно-логическая модель базы данных**

При разработке информационно – логической модели базы данных был использована программный продукт ERwin версии 4.1

В результате разработки получилась модель показанная на рисунке 6.

Рис. 6

Как видно из модели разрабатываемая база данных должна иметь возможность хранения получаемого потока данных, хранить данные. Как видно из модели – это «жесткая система». Все связи четко идентифицированы.

**8.ЗАКЛЮЧЕНИЕ**

Появление сетевых технологий гораздо облегчает, ускоряет работу персонала, позволяет использовать единые базы данных, а также регулярно и оперативно их пополнять и обрабатывать, все это весьма важно и существенно для работы в милиции, где базы данных содержат огромные объемы информации.

Как уже говорилось выше, компьютерные сети связывают компьютеры, каждый из которых может работать и автономно, это позволяет работать на ПЭВМ как индивидуально, так и в сети, имея доступ к общим базам данных и информации.

В ходе выполнения курсового проекта была сделана попытка разработать программные элементы информационно-справочной системы администратора ЛВС. Для выполнения поставленной задачи было использовано средство разработки функциональных моделей BР WIN. Для непосредственного моделирования использована методология функционального моделирования IDEFO, а также методология IDEF1X..

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Иванова Г.С. Технология программирования. Учебник для ВУЗОВ. 2-е изд., стереотип. - М.: Изд-во МГТУ имени Н.Э. Баумана, 2003. -320с.: ил.
2. Орлов С.А. Технологии разработки программного обеспечения . СПб.: Изд-во Питер, 2004. -526с.: ил.
3. Конспект лекций по дисциплине Технологии разработки программного обеспечения.