

**С. А. Цыганкова,
В. Д. Сквирский**

**КИБЕРНЕТИЧЕСКИЕ ОСНОВЫ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**



**Министерство образования и науки, молодежи
и спорта Украины
Государственное учреждение
«Луганский национальный университет
имени Тараса Шевченко»**

**С. А. Цыганкова,
В. Д. Сквирский**

КИБЕРНЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

*Учебно-методическое пособие
для магистрантов специальности
8.04030201 – «Информатика»*



**Луганск
ГУ «ЛНУ имени Тараса Шевченко»
2012**

УДК [681. 51: 004] (075. 8)

ББК 32. 81я73

Ц 94

Рецензенты:

Меняйленко А. С. – доктор технических наук, профессор, проректор по научно-педагогической работе Луганского национального университета имени Тараса Шевченко.

Адаменко Е. В. – доктор педагогических наук, профессор, декан факультета допрофессиональной подготовки Луганского национального университета имени Тараса Шевченко.

Руденко М. А. – кандидат технических наук, доцент кафедры экономической кибернетики Луганского национального аграрного университета.

Цыганкова С. А.

Ц94 Кибернетические основы информационных технологий : учеб.-метод. пособие для магистрантов специальности 8.04030201 – «Информатика» / С. А. Цыганкова, В. Д. Сквирский ; Гос. учрежд. «Луган. нац. ун-т имени Тараса Шевченко». – Луганск : ГУ «ЛНУ имени Тараса Шевченко», 2012.– 187 с.

Учебно-методическое пособие ориентировано на использование магистрантами специальности «Информатика». Назначение пособия – приобрести системное представление об источниках процессов функционирования и управления современных информационных технологий (ИТ) и оценить степень их связи с методами кибернетики на примере использования конкретных ИТ в проектировании информационных систем и создании электронного учебника.

УДК [681.51: 004] (075. 8)

ББК 32.81я73

*Рекомендовано к изданию учебно-методическим советом
Луганского национального университета имени Тараса Шевченко
(протокол № 10 от 04 мая 2012 года)*

@ Цыганкова С. А., Сквирский В. Д., 2012
@ ГУ «ЛНУ имени Тараса Шевченко», 2012

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ.....	4
ЧАСТЬ 1. КИБЕРНЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ.....	6
Введение.....	6
1. Истоки кибернетики и ее понятийный аппарат.....	6
2. Основные черты кибернетики.....	9
3. Методы кибернетики.....	10
4. Структура кибернетики.....	11
5. Влияние кибернетики на развитие ИТ.....	12
Контрольные вопросы.....	18
ЧАСТЬ 2. ПРИМЕНЕНИЕ КИБЕРНЕТИЧЕСКИХ АНАЛОГИЙ В ПРОЕКТИРОВАНИИ ИНФОРМАЦИОННЫХ СИСТЕМ (ИС).....	20
РАЗДЕЛ 1. ОСНОВНЫЕ ЭТАПЫ ЖИЗНЕННОГО ЦИКЛА ИС И ЕГО СТАНДАРТИЗАЦИЯ.....	20
Введение.....	20
1.1. Жизненный цикл ИС.....	23
1.2. Этапы разработки ПО.....	24
1.3. Реализация жизненного цикла в современных технологиях проектирования ИС.....	33
1.4. Обзор стандартов проектирования ИС.....	41
Контрольные вопросы.....	46
РАЗДЕЛ 2. МЕТОДИКА ПРОЕКТИРОВАНИЯ ПО НА ОСНОВЕ СЕТЕВОЙ МОДЕЛИ И MS PROJECT.....	47
2.1. Понятие о проектах и управлении ими.....	47
2.2. Методы управления проектами и основные этапы.....	48
2.3. Основные возможности MS PROJECT.....	60
Лабораторная работа №1. Знакомство с MS Project.....	65
Лабораторная работа № 2. Общая характеристика проекта ИС.....	73
Лабораторная работа № 3 Определение параметров проекта.....	77
Лабораторная работа № 4. Расчет стоимости проекта.....	104
Контрольные вопросы.....	109
ЧАСТЬ 3. ПРИМЕНЕНИЕ КИБЕРНЕТИЧЕСКИХ АНАЛОГИЙ В ФОРМИРОВАНИИ ПЕДАГОГИЧЕСКИХ СИСТЕМ.....	111
РАЗДЕЛ 1. ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ОБУЧЕНИЯ.....	111
1.1. Предпосылки применения кибернетики в построении обучающих систем.....	111
1.2. Этапы становления кибернетической педагогики.....	112
1.3. Компьютерная кибернетическая технология обучения.....	114
РАЗДЕЛ 2. ИСПОЛЬЗОВАНИЕ КИБЕРНЕТИЧЕСКИХ МОДЕЛЕЙ В СИСТЕМЕ ОБУЧЕНИЯ ПРИ СОЗДАНИИ ЭЛЕКТРОННОГО УЧЕБНИКА.....	134
2.1. Исследование дидактических особенностей учебного курса.....	134
2.2. Принципы разработки электронного учебника на основе семантической сети.....	135
2.3. Принципы структурирования учебного материала для электронного учебника.....	139
2.4. Построение семантической модели курса.....	143
2.6. Программная реализация оболочки для создания семантической модели курса.....	150
2.7. Описание навигационной системы курса.....	154
2.8. Технология создания навигационной системы курса.....	161
Лабораторная работа №1 Создание семантической модели учебного курса.....	167
Лабораторная работа № 2 Создание навигатора курса.....	169
ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....	170
ПРОЕКТНЫЕ ЗАДАНИЯ.....	171
ЛИТЕРАТУРА.....	174
ПРИЛОЖЕНИЕ.....	176

ПРЕДИСЛОВИЕ

Увеличивающееся значение кибернетического подхода в создании и сопровождении информационных технологий обусловило потребность в изучении дисциплины «Кибернетические основы информационных технологий» магистрантами по специальности «Информатика». В 1948 г. Винер в своей книге **«Кибернетика или управление и связь в животном и машине»** отметил тесную связь кибернетики и информационных технологий (ИТ), определив кибернетику как науку об управлении в живом и неживом мире, которая основывается на математике и вычислительных машинах.

В результате совместных усилий кибернетиков, программистов, математиков появились универсальные математические методы автоматизированной обработки информации любой специфики с помощью ЭВМ, общие математические модели управления различными по природе явлениями, методы отыскания оптимального процесса управления и многое другое, что составило содержание науки кибернетики. На первых этапах развития систем электронной обработки данных часто ЭВМ именовались кибернетическими машинами. В дальнейшем от кибернетики отпочковалась самостоятельная отрасль науки, которая получила название «Информатика». Однако практически все приемы обработки информации, разработанные в информатике, применяются в системах управления. Таким образом, эти две науки дополняют друг друга и способствуют взаимному развитию.

Цель курса – приобрести системное представление об источниках процессов функционирования и управления современных информационных технологий и оценить степень их связи с методами кибернетики.

КВАЛИФИКАЦИОННЫЕ ТРЕБОВАНИЯ

После изучения курса студенты должны:

1. Владеть основами знаний о методах и чертах кибернетики;

2. Иметь представление о принципах и методах теории систем и системного анализа и уметь использовать системный подход при создании информационных технологий в различных областях общественной жизни;

3. Иметь представление о методах теории информации и о информационных процессах в управлении как в технических, так и общественных системах и уметь использовать методы семиотики и математической лингвистики при создании информационно-поисковых систем, лингвистических интерфейсов и прочее;

4. Иметь представление о методах моделирования информационных процессов в кибернетике и уметь их использовать в проектировании информационных систем для различных областей общественной жизни.

5. Иметь представление о методах кибернетической педагогики и уметь их использовать при создании дистанционных систем обучения.

Базой для приобретения необходимых знаний и навыков по этой дисциплине является овладение учебным материалом предыдущих курсов: компьютерная техника, программирование, компьютерные сети, базы данных, автоматизированные системы управления, проектирование информационных систем, численные методы, педагогика и прочие дисциплины, которые включены в план подготовки специалистов по информатике.

ЧАСТЬ 1. КИБЕРНЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Введение

Начало НТР можно отнести к 1945 г. Если первая техническая революция касалась энергетической стороны явлений, то есть замены мускульной силы человека машиной, то НТР затронула информационные аспекты процессов.

Достижения НТР

- Внедрение автоматизированной обработки информационных данных привело к замене деловой информации в виде печатного слова ее представлением на машинных носителях.
- Соединение телекоммуникаций и ВТ позволило создавать системы нового типа — биосистемы «человек — машина», с помощью которых удается осуществить «перекачку» человеческого интеллекта в ЭВМ, которая с течением времени приводит к созданию искусственного интеллекта.

Роль кибернетики в НТР

У истоков этих достижений была наука – **кибернетика**. Автоматизированные методы обработки информации, разработанные в кибернетике, имеют отношение к специалистам любой профессии, так как все они имеют дело с переработкой деловой, технической, научной и любой другой информации. Эта особенность придает кибернетике универсальный характер и возводит ее в категорию фундаментальных наук.

1. Истоки кибернетики и ее понятийный аппарат

Понятие «**кибернетика**», название которой происходит от соединения двух греческих слов: «**кибер**» (в переводе «**над**») и «**наутис**» (моряк), т. е. «**кибернаутис**» — старший над моряками, главный моряк, кормчий. Греческий философ Платон впервые использовал термин «кибернетика» в смысле искусства

управления обществом. В XVIII в. французский ученый Ампер, составляя классификацию наук, также назвал кибернетикой науку об управлении обществом. Но в то время эта наука не имела математической и технической основы, и этот термин был на 200 лет забыт. В 1948 г. Винер в своей книге **«Кибернетика или управление и связь в животном и машине»** возродил этот термин в более широком, современном смысле и наметил по существу программу развития кибернетики.

Определения кибернетики

- *Кибернетика - это наука об управлении сложными динамическими системами (А.И. Берг, 1961 год).*
- *Кибернетика занимается изучением систем любой природы, способных воспринимать, хранить и перерабатывать информацию и использовать ее для управления и регулирования (А.Н. Колмогоров, 1959 год).*
- *«Кибернетика - это наука об общих законах получения, хранения, передачи и преобразования информации в сложных управляющих системах» (Крайзмер Л.П. Кибернетика. М., Экономика, 1977.-279 с.).*
- *Кибернетика изучает организацию систем в пространстве и времени, то есть то, каким образом связаны подсистемы в систему и как влияет изменение состояния одних подсистем на состояние других подсистем. Основной упор делается ...на организацию во времени, которая в случае, когда она целенаправленна, называется управлением".(Турчин В.Ф. Феномен науки: Кибернетический подход к эволюции. Изд. 2-е - М.: ЭТС, 2000. - 368 с.)*

Кибернетика носит синтетический характер, она многое взяла от разных наук

Исторически она создавалась как бы по двум направлениям:

- усилиями естественных наук– физики, биологии, химии–, которые стремились создавать модели явлений и процессов, используя методы математики;
- усилиями математики, которая понимая все значение задач управления, стала разрабатывать методы, которые позволили бы

наиболее полно и адекватно описывать различные процессы с информационной точки зрения, находить оптимальные режимы управления.

Наиболее весомыми теориями, объединяемыми кибернетикой, можно назвать следующие:

- § **Теория передачи сигналов** - изучает процессы формирования, накопления, сбора, измерения, переработки и преобразования (прохождения через цепи), хранения, передачи и приёма информации
- § **Теория информации** - определяет понятие информации, её свойства и устанавливает предельные соотношения для систем передачи данных. Использует, главным образом, математический аппарат теории вероятностей и математической статистики. Основные разделы теории информации — кодирование источника (сжимающее кодирование) и каналное (помехоустойчивое) кодирование. Теория информации тесно связана с криптографией и другими смежными дисциплинами.
- § **Теория систем** - концепция исследования объектов, представляющих собой системы, с целью изучения различных видов и типов систем, основных принципов и закономерностей поведения систем, функционирования и развития систем.
- § **Теория управления** - наука о принципах и методах управления различными системами, процессами и объектами.
- § **Теория автоматов** - в дискретной математике, разделе информатики, эта теория изучает абстрактные машины в виде математических моделей, и проблемы, которые они могут решать.
- § **Теория принятия решений** - область исследования, вовлекающая понятия и методы математики, статистики, экономики, менеджмента и психологии для изучения закономерности выбора людьми путей решения разного рода задач, а также исследует способы поиска наиболее выгодных из возможных решений.

- § **Синергетика** - направление научных исследований, задачей которого является изучение природных явлений и процессов на основе принципов самоорганизации и возникновения, поддержания, устойчивости и распада систем и подсистем самой различной природы.
- § **Теория алгоритмов** - наука, изучающая общие свойства и закономерности алгоритмов и разнообразные формальные модели их представления.
- § **Исследование операций** - дисциплина, занимающаяся разработкой и применением методов нахождения оптимальных решений на основе математического моделирования, статистического моделирования и различных эвристических подходов в различных областях человеческой деятельности.
- § **Распознавание образов** - раздел кибернетики, развивающий теоретические основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков.
- § **Теория оптимального управления** – изучает задачи оптимального (наилучшего в смысле выбранного критерия) управления и методы решения этих задач.

2. Основные черты кибернетики

Основными особенностями кибернетики являются:

- Информационный подход к процессам управления: там, где имеет место переработка информации с целью управления, присутствует кибернетика.
- Моделирование — сведение процессов управления к кибернетическим моделям. Одна и та же модель может применяться для разработки оптимального управления парикмахерскими, счетчиками космических частиц, системой противовоздушной обороны, процессами наследственности и пр.
- Системный подход к исследованию процессов управления. Системность предполагает единое комплексное рассмотрение и проектирование системы управления в целом.

- Вероятностный статистический подход к процессам управления. Эта концепция взята из статистической физики.

3. Методы кибернетики

Научных методов, используемых кибернетикой, достаточно много, основными из них, которые применяются для исследования большинства кибернетических моделей являются:

- Структурные методы (структурные схемы), использующие геометрическое представление системы, т. е. методы или процедуры в виде графа: в системах массового обслуживания, теории игр, теории детерминированных и вероятностных автоматов, при сетевом планировании, матричных расчетах и лингвистических методах анализа и синтеза .
- Метод дифференциальных уравнений, который применительно к АСР был трансформирован в частотный структурный метод.
- Топологический метод с использованием алгебры, логики, теории графов и комбинаторного исчисления (дискретная математика).
- Принцип обратной связи, который в дальнейшем перерос в методы самонастройки, самоорганизации и самоусовершенствования.
- Методы оптимизации в классическом вариационном исчислении, в основе которого лежит метод Эйлера — Лагранжа, и его дальнейших развитиях в плане динамического программирования и принципа максимума.
- Метод моделирования процессов управления на ЭВМ.

Способы исследования в кибернетике

Основным приемом исследования в кибернетике является: описание распространения информационных потоков в системе и изучение способов управления ими.

При этом выделяются следующие объекты исследования:

- **Источник информации.** В рамках человеко-машинной системы в качестве источника информации выступает человек или датчики.

- **Среда, по которой информация распространяется.** Для кибернетики - это, как правило, техническая или человеческая среда.
- **Приемник информации.** В качестве такого приемника может выступать как человек, так и техническое устройство.
- **Сток информации.** Иногда при рассмотрении информационных потоков сталкиваются с ситуацией, когда информация "входит, но не выходит" - это естественно интерпретировать как сток информации, как своего рода "информационную черную дыру" в изучаемой системе.
- **Искажение информации.** В рамках технической кибернетики их связывают, обычно, с шумами.
- **Восстановление информации** (или уменьшение искажений).

4. Структура кибернетики

Структура кибернетики может быть разбита на четыре раздела:

- общие сведения о кибернетике,
- математические основы кибернетики (табл.1.1),
- основы кибернетических моделей (табл.1.2),
- специальные и прикладные разделы кибернетики (табл.1.3).

Табл. 1.1 Математические основы кибернетики

Группа методов	Методы и теории
Вероятностные методы	Теория вероятностей
	Математическая статистика
	Теория информации и кодирования
	Теория марковских процессов
Методы оптимизации	Численные методы
Методы дискретной математики	Теория множеств и общая алгебра
	Математическая логика
	Теория алгоритмов
	Теория автоматов
	Общая теория графов
	Комбинаторное исчисление
	Математическая лингвистика

Табл.1.2 Основы кибернетических моделей

Группа моделей	Теории и методы
Управление вероятностными моделями	Теория массового обслуживания
	Теория игр и статистических решений
	Распознавание образов
Управление моделями АСР	Анализ и синтез непрерывных во времени АСР
	Анализ и синтез дискретных во времени АСР
	Самонастраивающиеся системы
Управление дискретными моделями	Управление с помощью графов
	Управление с помощью автоматов
	Лингвистическое управление

Табл.1.3. Специальные и прикладные разделы кибернетики

№ п/п	Специальные и прикладные разделы
1	Теория больших систем
2	Теория искусственного интеллекта
3	Проектирование автоматизированных систем управления (АСУ) различных назначений
4	Системы автоматизированного проектирования (САПР)
5	Основы информационного обеспечения систем управления
6	Основы алгоритмического обеспечения систем управления
7	Основы технического обеспечения систем управления
8	Основы программного обеспечения систем управления

5. Влияние кибернетики на развитие ИТ

1. Информационный подход:

- **Проектирование информационных систем.** В них создается информационное обеспечение, основными составляющими которого являются информационная модель предметной области и банк данных.
- **Измерение количества информации.** Теория Шеннона, семантическая и прагматическая теория информация, семиотика используется в измерении информационных потоков в системах

передачи данных как в технических устройствах, так и живых организмах.

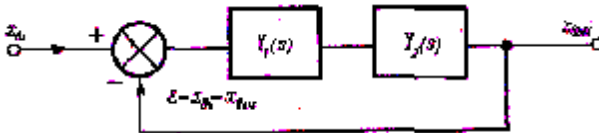
- **Создание информационно-поисковых систем (ИПС).** Теория дескрипторов, тезауруса, теория кодирования информации, семиотика, лингвистическая модель

2. Моделирование

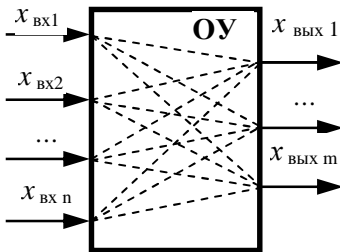
А). В проектировании технических устройств ЭВМ и коммуникативных систем:

Автоматические системы регулирования (АСР)

Система регулирования представляется в виде структурной схемы, состоящей из отдельных звеньев $Y_1(s)$, $Y_2(s)$ (здесь $Y_2(s)$ - объект управления). Как правило, во всех системах имеется звено сравнения задающего (входного) сигнала $x_{вх}$ с сигналом обратной связи (в данном случае - $x_{вых}$). Принцип обратной связи, позволяет получить сигнал управления $u = x_{вх} - x_{вых}$, обеспечивающий более высокую эффективность работы системы.



В теории управления применяется кибернетический подход в представлении многосвязной модели объекта как "черного ящика" с установлением зависимостей выходных сигналов от входных.



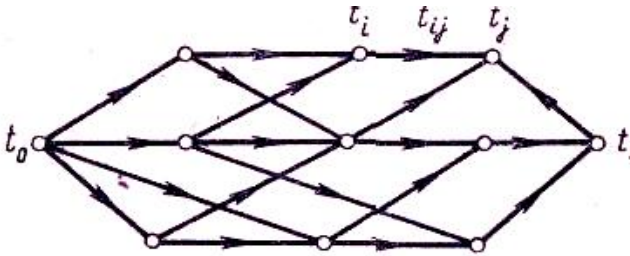
$$X_{\text{вых}}(s) = Y(s) X_{\text{вх}}(s),$$

где $X_{\text{вх}}(s) = \{x_{\text{вх}1}, \dots, x_{\text{вх}n}\}$,
 $X_{\text{вых}}(s) = \{x_{\text{вых}1}, \dots, x_{\text{вых}m}\}$,

$$Y(s) = \begin{vmatrix} Y_{11}(s) & \dots & Y_{1n}(s) \\ \dots & \dots & \dots \\ Y_{1m}(s) & \dots & Y_{mn}(s) \end{vmatrix}.$$

Сетевая модель

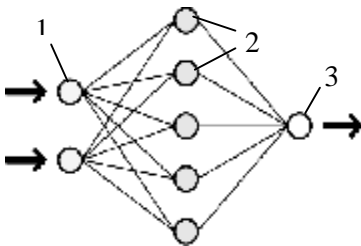
Информационная модель в виде ориентированного графа, называемого транспортной сетью или сетевым графиком (схема). Каждая отдельная работа (монтаж электронной схемы и т. д.) изображается в виде дуги графа, причем вершина, из которой она исходит, соответствует началу работы, а вершина, в которую она входит, — концу работы



Искусственная нейронная сеть

Искусственные нейронные сети (ИНС) — математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Впоследствии, после разработки алгоритмов обучения, получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

ИНС представляют собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов).



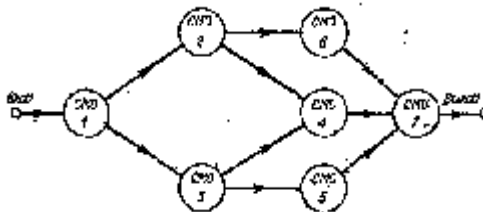
На схеме простейшей нейросети: 1 - входные нейроны, 2 - скрытые нейроны, 3 - выходной нейрон.

Модель в виде сети массового обслуживания (схема)

Система массового обслуживания (СМО) — система, которая производит обслуживание поступающих в неё требований. Обслуживание требований в СМО производится обслуживающими приборами. Классическая СМО содержит от одного до бесконечного числа приборов. В зависимости от наличия возможности ожидания поступающими требованиями начала обслуживания СМО подразделяются на

- системы с потерями, в которых требования, не нашедшие в момент поступления ни одного свободного прибора, теряются;
- системы с ожиданием, в которых имеется накопитель бесконечной ёмкости для буферизации поступивших требований, при этом ожидающие требования образуют очередь;
- системы с накопителем конечной ёмкости (ожиданием и ограничениями), в которых длина очереди не может превышать ёмкости накопителя; при этом требование, поступающее в переполненную СМО (отсутствуют свободные места для ожидания), теряется.

Выбор требования из очереди на обслуживание производится с помощью так называемой дисциплины обслуживания. Их примерами являются FCFS/FIFO (пришедший первым обслуживается первым), LCFS/LIFO (пришедший последним обслуживается первым), random (англ.) (случайный выбор). В системах с ожиданием накопитель в общем случае может иметь сложную структуру.



В показанной модели для каждой работы выделяется узел сети, представляющий собой отдельную систему массового обслуживания со своим (как правило, случайным) законом обслуживания.

Модель для задачи оптимального управления

Оптимальное управление — это задача проектирования системы, обеспечивающей для заданного объекта управления или процесса закон управления или управляющую последовательность воздействий, которые направлены на достижение максимум или минимум заданной совокупности критериев качества системы.

Для решения задачи оптимального управления строится математическая модель управляемого объекта или процесса, описывающая его поведение с течением времени под влиянием управляющих воздействий и собственного текущего состояния. Математическая модель для задачи оптимального управления включает в себя: формулировку цели управления, выраженную через критерий качества управления; определение дифференциальных или разностных уравнений, описывающих возможные способы движения объекта управления; определение ограничений на используемые ресурсы в виде уравнений или неравенств.

Модель оптимального управления:

- Уравнения состояния: $\dot{x}(t) = a[x(t), u(t), t]$.
- Граничные условия $x(t_0) = x_0^*$, $x(t_1) = x_1^*$
- Минимизируемый функционал: $h = \int_{t_0}^{t_1} F[x(t), \dot{x}(t), t] dt$.

здесь $x(t)$ — вектор состояния $u(t)$ — управление, t_0 , t_1 — начальный и конечный моменты времени.

Задача оптимального управления заключается в нахождении функций состояния $x(t)$ и управления $u(t)$ для времени $t_0 \leq t \leq t_1$, которые минимизируют функционал.

Б). При создании ПО, системных и прикладных интерфейсов:

Лингвистическая модель (семиотическая модель)

Лингвистическая модель, обладающая большими логическими возможностями, наилучшим образом описывает поведение систем, в которых существенную роль играют люди, так как представляется возможным (через семантику) учесть элементы их разумного, творческого поведения. Термин «лингвистический» означает использование методов, развиваемых в математической лингвистике, которая исследует алгебраическими методами естественные языки. Любая модель системы (сетевая, массового обслуживания и т. д.) становится семиотической, когда ее записывают в виде программы в кодах машины или на каком-нибудь алгоритмическом языке для моделирования на ЭВМ. Большую роль играет лингвистическая модель в формировании современных технологий автоматизации процесса обучения, т.е. педагогических систем.

3. Системный подход

При создании в комплексе всех подсистем ИС.

Системность предполагает единое комплексное рассмотрение и проектирование системы управления в целом и представляет достаточно сложное переплетение организационных, информационных, математических, программных и технических средств и приемов. Первый этап разработки АИС состоит в системном обследовании объекта, который предполагается автоматизировать, в результате чего уточняется его функциональная структура, объем массивов информации и документов, циркулирующих в нем. Системность достигается единым информационным, математическим и техническим обеспечением в отдельных функциональных подсистемах.

Следует заметить, что не все системы могут исследоваться методами кибернетики. У **кибернетических систем** предполагается или наблюдается ряд новых свойств, которые не присущи или не обязательно присущи системам других классов:

- множественность поведения;
- управляемость;
- наличие управляющего устройства;
- способность взаимодействовать с окружающей средой как непосредственно, так и через посредство управляющего устройства;
- наличие -каналов информации как в системе, так и между системой и средой;
- наличие обратных связей в каналах информации;
- целенаправленное поведение системы;
- вероятностный характер поведения системы;
- свойство равновесия; свойство самоорганизации

Контрольные вопросы

1. Сформируйте понятие кибернетики, историю формирования этой науки и ее родоначальника.
2. Каковы предпосылки и объективная необходимость появления кибернетики?
3. Какие научные открытия послужили основой кибернетики?
4. Какое влияние оказывает кибернетика на развитие информационных технологий?
5. Обозначьте основные черты кибернетики
6. Выделите основные структурные компоненты и методы кибернетики
7. Сформируйте основные понятия теории систем, какое понятие Вам более импонирует?
8. По каким признакам принято классифицировать системы, можете ли Вы отнести эти признаки к информационным системам, приведите примеры.
9. Опишите основные закономерности (свойства) систем
10. Определите сущность системного подхода
11. Определите основные этапы системного анализа, используются ли эти подходы при создании ИТ
12. Назовите основные свойства кибернетических систем, найдите аналогии с информационными системами

13. Вы знаете, какая роль информации в кибернетической системе?
14. Приведите классификацию форм представления информации.
15. Графическая информация и ее основные понятия.
16. Что такое «символьная информация»?
17. Что такое структурирование информации, какие методы структурирования информации Вы знаете?
18. Классификация информации и ее основные методы
19. Кодирование информации и основные методы кодирования
20. Какая наука изучает знаки и знаковые системы и приведите основные понятия этой науки?
21. Сущность статистического подхода к определению количества информации.
22. Сущность семантического и прагматического подходов к определению количества информации.
23. Принцип «максимума информации».
24. Дайте понятие ИПС и опишите ее понятийный аппарат.
25. Опишите технологию поиска.
26. Классификация ИПС и типы документальных ИПС.
27. Сформулируйте понятие «тезаурус», по каким типам связей он строится по стандарту?
28. Релевантность: понятие и показатели оценки релевантности поиска.
29. Роль ИПС в Интернет, по какому принципу они строятся.
30. Направления развития ИПС.

ЧАСТЬ 2. ПРИМЕНЕНИЕ КИБЕРНЕТИЧЕСКИХ АНАЛОГИЙ В ПРОЕКТИРОВАНИИ ИНФОРМАЦИОННЫХ СИСТЕМ (ИС).

РАЗДЕЛ 1. ОСНОВНЫЕ ЭТАПЫ ЖИЗНЕННОГО ЦИКЛА ИС И ЕГО СТАНДАРТИЗАЦИЯ

Введение

Понятие **информационная система** в современном глобальном информационном пространстве стало не только объектом теоретических рассуждений, но и объектом реального практического воплощения (проектирования).

Информационная система (ИС) — система, которая организует память и манипулирование информацией о проблемной (предметной) сфере.

Проблемная (предметная) сфера — все сущности, которые являются объектом интереса, которые были, есть или могут быть.

Любая система (биологическая, техническая, информационная) обладает ограниченным ресурсом развития и проходит стадии развития и угасания, называемые жизненным циклом развития.

Жизненный цикл ИС (ЖЦ ИС) — непрерывный процесс, который начинается с момента принятия решения о необходимости создания системы и заканчивается в момент ее полного изъятия из эксплуатации.

Проектирование ИС — процесс, направленный на совершенствование информации системы объекта управления (ОУ), которое предусматривает создание и внедрение комплексного решения задач проблемной сферы с применением современных ЭВМ и технических средств управления объектом.

Можно отметить следующие современные тенденции, серьезно изменившие подход к проектированию информационных систем :

1. Существенное ускорение изменений внешней среды, в которой работают организации;
2. Все увеличивающаяся глобализация бизнеса;
3. Появление технологий Internet/Intranet, которые позволили существенно упростить разработку и эксплуатацию информационных систем и сделали возможным создание более сложных ИС, в т.ч. распределенных по всему миру.

Первые две тенденции привели к тому, что для адекватного анализа информации организации требуется создавать все более сложные распределенные информационные системы с огромными объемами информации. Такие ИС получили наименование корпоративных.

Корпоративная ИС (КИС) — сложная система, организованная на принципах централизованных коммуникаций и координации; это средство информационной поддержки корпоративного управления; совокупность всех принятых в конкретной компании технологий обработки информации независимо от формы их реализации: электронной, бумажной или любой другой. Включает бизнес-архитектуру предприятия, его персонал, информационные технологии и действующую часть киберкорпорации.

Киберкорпорация (виртуальная корпорация) — объединение предприятий, информационная система которого основывается на принципах нового системного проектирования, которое обеспечивает постоянную гибкую реконструкцию бизнеса

В этих условиях системный подход к проектированию информационных систем стал доминирующим и появилось понятие **новое системное проектирование**.

Новое системное проектирование — совокупность методов и организационных подходов, что интегрирует три источника построения современных информационных систем: методы бизнес-реинжиниринга, автоматизированные информационные технологии, методы учета «человеческого фактора».

Методы проектирования ИС — разные способы их создания, которые поддерживаются соответствующими средствами проектирования.

Бизнес-реинжиниринг — комплекс мероприятий для поэтапного внедрения информационной системы управления предприятием.

Автоматизированная информационная технология — целостная технологическая система, которая обеспечивает целеустремленное создание, передачу, сохранение и отображение информационного продукта с наименьшими расходами и в соответствии с закономерностями той социальной среды, где эта технология внедрена.

Постоянное увеличение объемов и сложности программных систем (ПС), а также рост требований к их качеству, привели к активному развитию технологий, автоматизирующих процессы жизненного цикла программных средств, т.е. к системам автоматизированного проектирования ИС. Потребность в таких технологиях основывается на постоянном усложнении проектов проектирования, разработки и сопровождения программных средств, за счет увеличения числа участников проекта, ужесточении требований к качеству и срокам выхода продукта на рынок.

Программные проекты с увеличением числа участников превращаются в трудно управляемые. Становится невозможным поддерживать высокое качество выпускаемых систем без специальных методов управления жизненным циклом программных средств (ЖЦ ПС).

Автоматизированное проектирование — проектирование с помощью технических и программных средств и участием человека, результатом этой деятельности является планирование и реализация проекта.

Проект — однократная деятельность, направленная на эффективное использование различных ресурсов с целью достижения конкретного результата или разработки нового изделия.

Значение проекта в современном искусстве управления постоянно повышается. Для эффективной деятельности большого числа участников проекта необходима организация коллективов (проектных групп), основанная на четкой регламентации деятельности как коллектива в целом, так и отдельных участников проекта с ясно определенным распределением ответственности. Появилось понятие «управление проектом» и необходимость стандартизировать процесс управления и его автоматизировать. Таким образом, стали особо актуальными проблемы стандартизации всех этапов

жизненного цикла ИС и автоматизации управления процессом проектирования ИС.

Стандарты ЖЦ ИС – набор регламентированных законом или другими правовыми нормами процедур проведения в определенной последовательности мероприятий создания и сопровождения ИС и подготовки необходимой документации.

Управление проектом — это процесс планирования, организации и управления задачами и ресурсами с целью достижения определенной цель, обычно при наличии ограничений по времени, ресурсам и затратам

Большую помощь в решении задач управления проектами создания ИС оказывают программы управления проектами, такие как MS PROJECT и другие.

Microsoft Project предназначен для управления средними проектами, позволяющими планировать и управлять выполнением около 10000 задач. Эта система ориентирована на использование непосредственно руководителями проекта. Microsoft Project позволяет представить информацию о работах проектов в формах PERT, GANTT, календаря, позволяющих отобразить работы плана на одном-двух уровнях иерархии).

В основу создания информационных технологий управления проектами, например, MS PROJECT, положены такие основополагающие методы кибернетики, как моделирование (сетевая модель и т.п), методы оптимизации, информационный подход к процессам управления (теория обратной связи), теория систем и системный анализ и другие. За счет использования этих инструментальных средств повысилась степень автоматизации рутинных операций, что позволило снизить влияние “человеческого фактора” на конечный результат и повысить эффективность проектов.

1.1. Жизненный цикл ИС

Жизненный цикл системы — непрерывный процесс, который начинается с момента принятия решения о

необходимости создания системы и заканчивается в момент ее полного изъятия из эксплуатации.

Понятие жизненного цикла ИС - это исходное понятие для исследования проблем и задач, связанных с информационными системами управления, а также принятия соответствующих решений. Модель жизненного цикла системы является концептуальной структурой, включающей процессы, действия и задачи, которые должны быть выполнены во время построения ИС.

В том, что касается основной и ключевой составляющей жизненного цикла ИС, то ею является жизненный цикл ее программного обеспечения (ПО). Хотя ПО - это базовый компонент для эффективного построения и функционирования ИС, рассмотрение только его жизненного цикла является недостаточным, а также требует дополнительного анализа и реализации более широкого круга процессов. Это связано, прежде всего, с тем, что на сегодняшний момент построение ИС рассматривается не просто как технический процесс внедрения определенного ПО, а более сложный и комплексный процесс, затрагивающий напрямую бизнес-процессы предприятия. А если учитывать глобальность современных бизнес-процессов, то можно представить насколько масштабным может стать процесс построения ИС.

1.2. Этапы разработки ПО

Классический жизненный цикл

Одной из старейших последовательностей шагов разработки программного обеспечения (ПО) является классический жизненный цикл (Автор Уинстон Ройс, 1970).

Чаще классический жизненный цикл называют **КАСКАДНОЙ** или **ВОДОПАДНОЙ** моделью

Каскадная модель — последовательность этапов, причем переход на следующий иерархически нижний этап происходит только после полного завершения работ на текущем этапе и возврата к пройденным этапам не предусматривается.

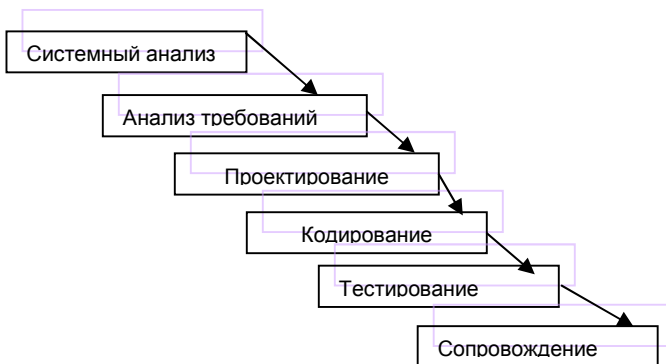


Рис. 1.1.Каскадная модель разработки ПО.

Приведем краткое описание основных этапов. Разработка начинается на системном уровне и проходит через:

- анализ,
- проектирование,
- кодирование (реализация),
- тестирование,
- сопровождение

При этом моделируются действия стандартного инженерного цикла.

Системный анализ определяет роль каждого элемента в компьютерной системе, взаимодействие элементов друг с другом.

Анализ начинается с определения требований и назначения подмножества этих требований программному элементу.

На этом этапе начинается решение задачи планирования проекта ПО.

В ходе планирования проекта определяются:

- объем проектных работ,
- риск проектных работ,
- необходимые трудозатраты,
- формируются рабочие задачи,
- формируется план-график работ.

Анализ требований, относящийся к программному элементу, т.е. к ПО, уточняет и детализирует:

- функции ПО,
- характеристики ПО,
- интерфейс ПО.

Все определения документируются в спецификации анализа.

Проектирование создает представления:

- архитектуры ПО,
- модульной структуры ПО,
- алгоритмической структуры ПО,
- структуры данных,
- входного и выходного интерфейса (входных и выходных форм данных).

Кодирование (реализация) состоит в переводе результатов проектирования в текст на языке программирования.

Тестирование – это выполнение программы для выявления дефектов в функциях, логике и форме реализации программного продукта.

Сопровождение – это внесение изменений в эксплуатируемое ПО. Цели изменений:

- исправление ошибок,
- адаптация к изменениям внешней для ПО среды,
- усовершенствование ПО по требованию заказчика.

Сопровождение ПО состоит в повторном применении каждого из предшествующих шагов (этапов) жизненного цикла, т.е. системного анализа, анализа требований, проектирования и т. д., к существующей программе, но не разработке новой программы.

Каждая стадия (этап) завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков.

Достоинствами классического жизненного цикла являются:

- получение плана и временного графика по всем этапам проекта,
- упорядочение хода разработки.

К недостаткам классического жизненного цикла относятся:

- частое отклонение реальных проектов от стандартной последовательности шагов,

- основанность цикла на точной формулировке исходных требований к ПО, тогда как реально в начале проекта требования заказчика определены лишь частично,
- доступность результатов проекта заказчику лишь в конце работы.

Макетирование (прототипирование)

На начальной стадии проекта полностью и точно сформулировать все требования к будущей модели невозможно, поскольку пользователи, как правило, не в состоянии изложить все свои требования и не могут предвидеть, как они изменятся в ходе разработки, и, кроме того, за время разработки могут произойти изменения во внешней среде, которые могут повлиять на требования к системе. Поэтому процесс создания ПО носит скорее итерационный характер, когда результаты очередной стадии разработки могут вызвать необходимость возврата к предыдущим разработкам.

Поэтому ПО создается не сразу, как в случае каскадного подхода, а постепенно с использованием макетирования (прототипирования), когда создается модель требуемого программного продукта.

Прототип — действующий программный компонент, реализующий отдельные функции.

Модель может принимать одну из трех форм:

- бумажный макет или макет на основе ПК (изображает или рисует человека – машинный диалог),
- работающий макет (выполняет некоторую часть требуемых функций),
- существует программа, характеристики которой затем должны быть улучшены.

Макетирование основывается на многократном повторении итераций, в которых участвуют заказчик и разработчик.

Последовательность действий при макетировании представлена на рисунке.

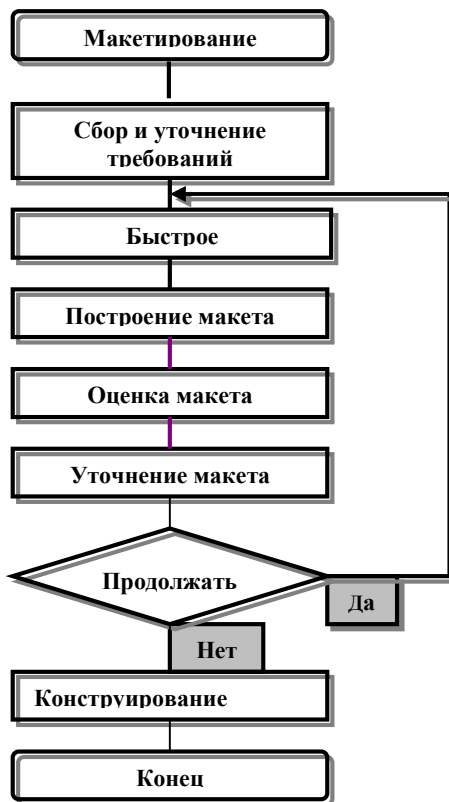


Рис.1.2. Последовательность действий при макетировании.

Достоинством макетирования является обеспечение определения полных требований к ПО.

К недостаткам макетирования относятся:

- возможность принятия заказчиком макета за продукт,
- возможность принятия разработчиком макета за продукт

Заказчик, получивший предварительную версию (макет) и удостоверившийся в ее работоспособности, может перестать видеть недостатки и нерешенные вопросы ПО и перестать соглашаться на дальнейшее усовершенствование, требуя скорейшего преобразования макета в рабочий продукт.

Поскольку часто заказчик не может определиться в своих требованиях по разрабатываемому продукту, а проектировщик сомневается в полноте и целесообразности требований заказчика, то прототипирование (макетирование) начинается со сбора и уточнения требований к создаваемому ПО.

Совместными усилиями разработчик и заказчик определяют все цели ПО, устанавливая, какие требования известны, а какие предстоит доопределить. Следующим шагом является быстрое проектирование, внимание в котором сосредотачивается на тех характеристиках ПО, которые должны быть видимы пользователю. Макет (прототип), построенный на этапе быстрого проектирования, оценивается заказчиком и используется для уточнения требований к ПО. Итерации повторяются до тех пор, пока макет не выявит все требования заказчика и не даст возможности разработчику понять, что должно быть сделано.

В тоже время для экономии времени разработки макета, а также возможности показать работающий вариант, разработчик может использовать неэффективные средства. Забывая о причинах, побудивших использовать эти средства, разработчик может интегрировать неэффективный вариант в систему.

Стратегии разработки ПО

Стратегии разработки ПО можно подразделить на три группы:

1. **Линейная последовательность этапов разработки** – однократный проход (водопадная стратегия)
2. **Инкрементная стратегия**, когда сначала определяются все требования (пользовательские и системные), а затем оставшаяся часть разработки выполняется в виде последовательности версий, первая из которых реализует часть запланированных возможностей, а все последующие версии реализуют дополнительные возможности до тех пор, пока не будет получена полная система.
3. **Эволюционная стратегия.**

При этой стратегии начальный этап не содержит полного объема требования, они уточняются в ходе разработки новых последовательных версий.

Инкрементная стратегия

Инкрементная модель является классическим примером инкрементной стратегии разработки ПО, объединяя элементы последовательной водопадной модели с итерационной философией макетирования. Она представляет собой несколько поставок (инкрементов), представляющих собой последовательность анализа, проектирования, кодирования и тестирования.

Разработка первого инкремента позволяет получить базовый продукт, реализующий базовые требования, при этом многие вспомогательные требования остаются нереализованными. План следующих инкрементов предусматривает последовательную модификацию базового продукта, обеспечивающих дополнительные характеристики и функциональность.

По своей природе инкрементный процесс итеративен, но в отличие от макетирования инкрементная модель обеспечивает в конце инкрементной итерации работающий продукт.

Эволюционная стратегия разработки ПО

Эволюционную стратегию рассмотрим на примерах спиральной модели и компонентно-ориентированной модели проектирования.

Спиральная модель

Спиральная модель (автор Боэм Б, 1988 г.) опирается на лучшие свойства классического жизненного цикла и макетирования, к которым добавляется новый элемент – анализ риска, отсутствующий в этих шагах разработки.

Спиральная модель определяет в режиме итерации (от центра к периферии) планирование (определение целей, вариантов, ограничений), анализ риска (анализ вариантов и распознавание/выбор риска), конструирование (разработка

продукта следующего уровня), оценивание (оценка заказчиком текущих результатов разработки).

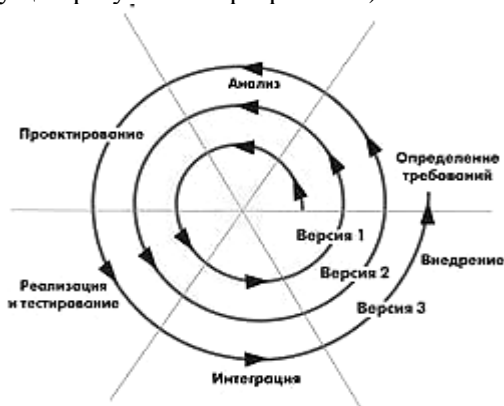


Рис. 1.3. Спиральная модель разработки ПО

С каждой итерацией по спирали (продвижением от центра к периферии) строятся все более полные версии ПО. В первом витке спирали определяются:

- 1) начальные цели, варианты и ограничения;
- 2) распознавание и анализ риска;
- 3) необходимость использования макетирования;
- 4) оценка заказчиком конструктивной работы и внесение предложения по модификации;
- 5) следующая фаза планирования и анализа риска, базируемая на предложениях заказчика.

В каждом цикле по спирали результаты анализа риска формируются в виде «продолжать, не продолжать». Если риск слишком велик, проект может быть остановлен. В большинстве случаев движение по спирали продолжается, с каждым шагом продвигая разработчиков к более общей модели системы. В каждом цикле по спирали требуется конструирование, которое может быть реализовано классическим жизненным циклом или макетированием.

К достоинствам спиральной модели относится:

- 1) наиболее реальное (в виде эволюции) отображение разработки программного обеспечения,
- 2) возможность явно учитывать риск на каждом витке эволюционной разработки,

3) включение шага системного подхода в итерационную структуру разработки,

4) использование моделирования для уменьшения риска и совершенствования программного изделия.

Недостатками спиральной модели являются:

- 1) повышенные требования к заказчику,
- 2) трудности контроля и управления временем разработки.

Компонентно-ориентированная модель.

Компонентно-ориентированная модель является развитием спиральной модели и основывается на эволюционной стратегии разработки ПО. В этой модели конкретизируется содержание конструирования – оно отображает тот факт, что в современных условиях новая разработка должна основываться на повторном использовании существующих программных компонентов.

К достоинствам компонентно-ориентированной модели относятся:

- 1) уменьшение времени разработки ПО;
- 2) снижение стоимости программной разработки;
- 3) повышение производительности разработки.

Тяжеловесные и облегченные процессы

Традиционно для упорядочения и ускорения программных разработок использовались строго упорядочивающие так называемые **тяжеловесные процессы**. В этих процессах прогнозируется весь объем предстоящих работ, поэтому они называются **прогнозирующимися процессами**. Порядок, который должен выполнять при этом человек-разработчик, чрезвычайно строг.

В последние годы появилась группа новых **облегченных процессов** разработки ПО. Их также называют **подвижными процессами**. Эти процессы привлекательны отсутствием бюрократизма, характерного для тяжеловесных (прогнозирующих) процессов.

Облегченные процессы разработки ПО воплощают разумный компромисс между строгой дисциплиной и отсутствием ее.

Подвижные процессы требуют меньшего объема документации и ориентированы на человека. Подвижные процессы учитывают особенности современного заказчика, а именно, частые изменения его требований к ПО. Подвижные процессы адаптируют изменения требований (адаптивная природа).

1.3. Реализация жизненного цикла в современных технологиях проектирования ИС

Общая характеристика подходов к технологии проектирования ИС

Этапы проектирования, эксплуатации и сопровождения программ различаются целями, задачами, методами и средствами. Развитие программирования за более чем 50 лет привело к разработке методов и средств, позволяющих создавать программы для различных областей применения и типов ЭВМ, то есть была создана **современная технология программирования**.

Важность разработки современной технологии программирования следует из того, что, во-первых, процесс программирования приобрел массовый характер, а во-вторых продолжает оставаться трудоемким в связи с тем, что возрастает сложность решаемых на ЭВМ задач.

Технология программирования — совокупность приемов и процедур проведения процесса программирования, обеспечивающего в заданных условиях получение программного продукта с заданными свойствами. Технология программирования определяет некоторую профессиональную культуру работы специалистов, обеспечивающую заданный уровень производительности труда.

Требования к современной технологии программирования следующие:

1. Она должна учитывать человеческий фактор в программировании свести к минимуму.

2. Она должна быть ориентирована на работу целого коллектива программистов, а не отдельных личностей.

3. Она должна быть безбумажной.

4. Средства автоматизации технологии должны охватывать все этапы разработки и эксплуатации программного обеспечения.

5. Она должна быть простой в освоении.

Персонализация информационно-вычислительных средств особенно сильно повлияла на номенклатуру, организацию и потребительские свойства программного обеспечения, а соответственно и на технологию программирования.

Программы для решения комплексов задач создаются и используются при участии специалистов различной квалификации, которые объединяются в проектную группу на время разработки проекта ИС.

Проектная группа — группа сотрудников, ответственных за выполнение соответствующих частей проекта.

В проектную группу обычно входят:

1. Системные программисты — разрабатывают и сопровождают системное программное обеспечение, необходимое для работы компьютера и создания среды выполнения программ.

2. Прикладные программисты и программисты-аналитики – осуществляют проектирование комплексов взаимосвязанных программ.

3. Тестировщики — составляют тест-планы, тестовые сценарии для отладки программных модулей и программных комплексов;

4. Постановщики задач — участвуют на начальной стадии работ по созданию машинных алгоритмов и структуры информационной базы.

5. Администратор БД — осуществляет организационную поддержку БД, используемой при работе программ.

6. Системный администратор — участвует в согласовании

требований эксплуатации программ в условиях сетевой технологии обработки данных .

7. Конечный пользователь — потребитель программы .

Разработка программ осуществляется на основе **промышленной технологии** выполнения работ с применением современных инструментальных средств программирования.

Основные характеристики программ:

- логика алгоритмов обработки информации;
- качество проработки, полнота и системность реализованных функций;
- внутренняя архитектура программного комплекса, эффективность работы программы;
- качество документации;
- удобство освоения и работы с программой конечного пользователя,
- цена программы.

Спецификой программных продуктов является их эксплуатация на правовой основе: заключаются лицензионные соглашения между разработчиком и пользователями, соблюдаются авторские права разработчиков.

Важнейшими классификационными признаками технологий проектирования ПО являются:

- степень автоматизации выполнения проектных работ;
- методология процесса разработки ПО.

По степени *автоматизации* процесса проектирования ПО можно выделить технологии неавтоматизированного проектирования и методы автоматизированного проектирования (CASE-Computer Aided System Engineering).

Неавтоматизированное проектирование используется при разработке небольших по трудоемкости и структурной сложности программных продуктов, не требующих участия большого числа разработчиков.

Автоматизированное проектирование — проектирование с помощью технических и программных средств и участием человека, результатом этой деятельности является планирование и реализация проекта.

Автоматизированное проектирование возникло в связи с необходимостью сокращения затрат и сроков выполнения

проектных работ, создания и использования типовых элементов и их комплексов при разработке алгоритмов и программ, координации работ большого коллектива разработчиков, использования средств вычислительной техники на различных этапах создания и сопровождения программных продуктов. Автоматизация проектирования может охватывать все или отдельные этапы жизненного цикла программного продукта, в изолированном виде или в комплексе.

О методологии процесса разработки ПО поговорим ниже.

Унифицированная методология процесса разработки ПО

Развитие методологии процесса разработки ПО привела к методологии унификации проектирования на основе создания унифицированных процессов создания ПО и специального языка моделирования.

Унифицированный процесс проектирования ПО и язык моделирования

Унифицированный процесс (УП) создания ПО — сумма различных видов деятельности, необходимых для преобразования требований пользователей в программную систему

УП – это обобщенный каркас процесса, который может быть специализирован для :

- 1) широкого круга программных систем,
- 2) различных областей применения,
- 3) размеров проекта (рис. 1.4).

УП компонентно-ориентирован, т.е. создаваемая программная система строится на основе программных компонентов, связанных хорошо определенными интерфейсами. Для разработки чертежей программной системы используется унифицированный язык программирования (UML). UML является неотъемлемой частью УП. УП и UML разрабатывались совместно.

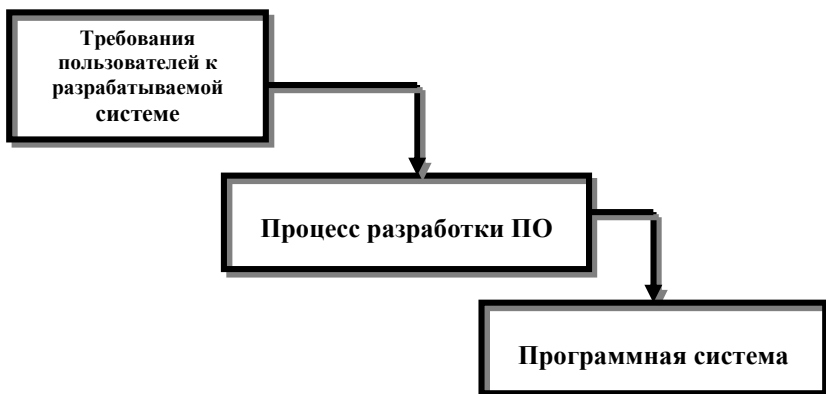


Рис.1.4. Унифицированный процесс (УП) разработки ПО.

Специфические аспекты УП:

- 1) управляемый вариантами использования;
- 2) архитектурно-ориентированный;
- 3) итеративный и инкрементный.

Управляемый вариантами использования - означает, что процесс разработки проходит серию рабочих процессов, порожденных вариантами использования.

Варианты использования — взаимодействия между системой и пользователем для получения значимого результата, которые обеспечивают функциональные требования.

Всю полноту функциональности системы описывает модель вариантов использования, состоящая из совокупности всех вариантов использования, заменяя традиционное описание функций системы.

УП ориентирован на архитектуру – означает, что архитектура должна предоставлять возможность реализации любых понадобившихся вариантов использования, архитектура и варианты использования разрабатываются параллельно.

Архитектура проекта — представление всего проекта с выделением важных характеристик.

Архитектура вырастает из требований к результату пользователей и других заинтересованных лиц, которые

отражаются в вариантах использования. Однако на требования также влияют такие факторы, как:

- 1) выбор платформы для работы программы (т.е. компьютерной архитектуры, ОС, СУБД, сетевых протоколов);
- 2) нефункциональные требования (например, производительность, надежность);
- 3) и т.д.

Связь архитектуры с вариантами использования в ПО представляет собой единство функций, соответствующих вариантам использования, и формы – архитектуры.

УИ является итеративным и инкрементным – означает, что для экономии времени разработки проекта и правильного распределения работы между исполнителями предлагается разделить проект на отдельные мини-проекты (циклы). Каждый мини-проект является итерацией, результатом которой будет приращение (инкремент).

Целью каждого цикла является новый выпуск системы, представляющий собой продукт, готовый к поставке. **Продукт** состоит из тела (исходный код), руководства и дополнительных компонент поставки. Окончательный продукт включает в себя:

- 1) требования,
- 2) варианты использования,
- 3) нефункциональные требования,
- 4) варианты тестирования.

Полное представление готового продукта состоит из:

модели вариантов использования или модели прецедентов, содержащей все варианты использования и их связи с пользователями;

модели анализа, уточняющей детали вариантов использования и создающей первичное распределение поведения системы по набору объектов;

модели проектирования, определяющей статическую структуру системы, такую как подсистемы, классы, интерфейсы, и варианты использования, реализованные в виде коопераций между подсистемами, классами и интерфейсами;

модели реализации, включающей в себя компоненты (представленные исходным кодом) и распределения классов по компонентам;

модели развертывания, представляющей физические компьютеры (узлы сети) и распределение компонентов по этим узлам;

модели тестирования, описывающей тесты для проверки вариантов использования;

представления архитектуры.

Система также может включать в себя модель предметной области или бизнес-модель, описывающую контекст системы.

Все эти модели связаны. Вместе они полностью описывают систему (рис. 1.5.).

Описание схемы:

Анализ. Построение модели анализа необходимо для уточнения и упорядочивания функциональных требований, выявленных с помощью модели прецедентов (вариантов использования).

Проектирование. Модели проектирования представляет собой физическую реализацию прецедентов из модели прецедентов, а также из содержания модели анализа. Модель проектирования служит абстракцией модели реализации.

Модель развертывания. В технологическом процессе проектирования модель развертывания определяет физическую организацию системы в терминах вычислительных узлов.

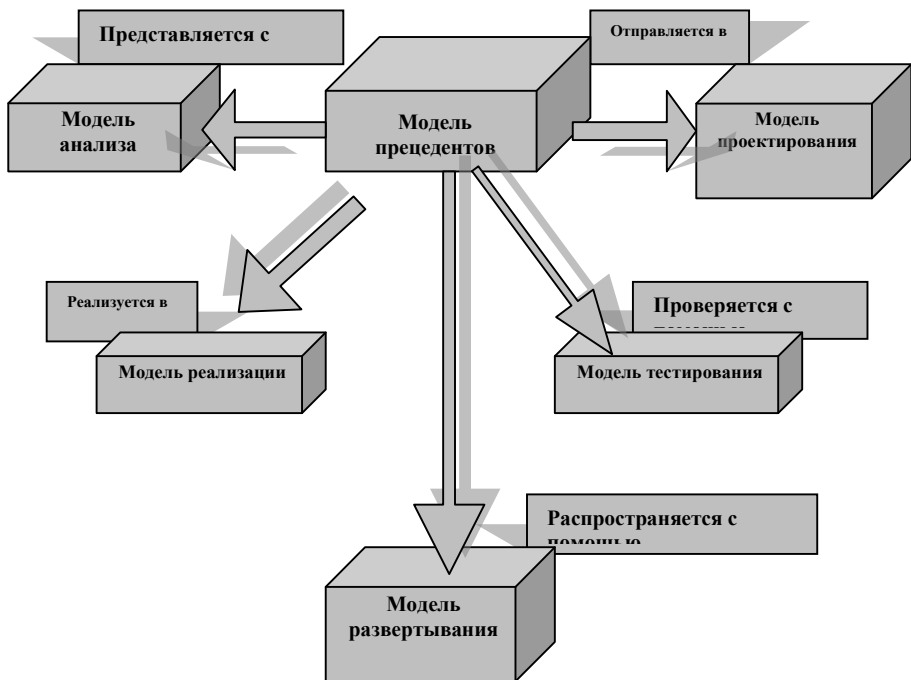


Рис. 1.5. Схема взаимосвязи унифицированных процессов (УП)

Реализация. Целью этого технологического процесса является построение модели реализации, которая описывает, как элементы модели проектирования формируют такие компоненты как файлы исходного кода и библиотеки динамической компоновки (DLL и Enterprise Java Beans).

Тестирование. Основные виды деятельности этого технологического процесса направлены на построение модели тестирования, которая описывает, как системные тесты и тесты интеграции будут применяться к рабочим компонентам модели реализации.

1.4. Обзор стандартов проектирования ИС.

Стандарты ЖЦ ИС – набор регламентированных законом или другими правовыми нормами процедур проведения в определенной последовательности мероприятий создания и сопровождения ИС и подготовки необходимой документации. Когда понятие "информационная система" фигурирует в контексте проектирования для определенной организации (предприятия), чаще всего под ним понимают совокупность бизнес-процессов на предприятии, а также аппаратных и программных средств и обслуживающий персонал, которые могут выполнять определенные функции. Проектирование таких ИС может занимать примерно 3-6 месяцев и включает решение не только проектных вопросов. На рисунке 1.6 представлена схема этапов и задач, которые входят в процесс создания информационной системы для организации. Как видно из схемы:

- ИС включает не только ПО, но и аппаратное и организационное обеспечение, а также человеческие ресурсы;
- ИС затрагивает бизнес-процессы предприятия, непосредственно интегрируясь в них: процессы, реализуемые в рамках ИС, - это не отдельные процессы, а процессы, поддерживающие и оптимизирующие (путем автоматизации) бизнес-процессы;
- проблемы, связанные с построением ИС, зачастую выходят за рамки технических, и приобретают управленческо-организационный характер;
- каждое предприятие имеет специфические условия, определенные ограничения и возможности;

При проектировании такого рода систем встает ряд вопросов, особенно если система создается разными отделами предприятия и привлеченными к работам сторонними компаниями, - а ведь именно так чаще всего и бывает. В таком случае разрабатываются основополагающие нормативные документы, и проведение проекта внедрения системы во всем

согласуется с ними. Обычно для разработки нормативных документов используются широко распространенные стандарты, которые, по сути, не должны иметь многих толкований и описывают процедуры подготовки необходимой документации и проведения определенных мероприятий.

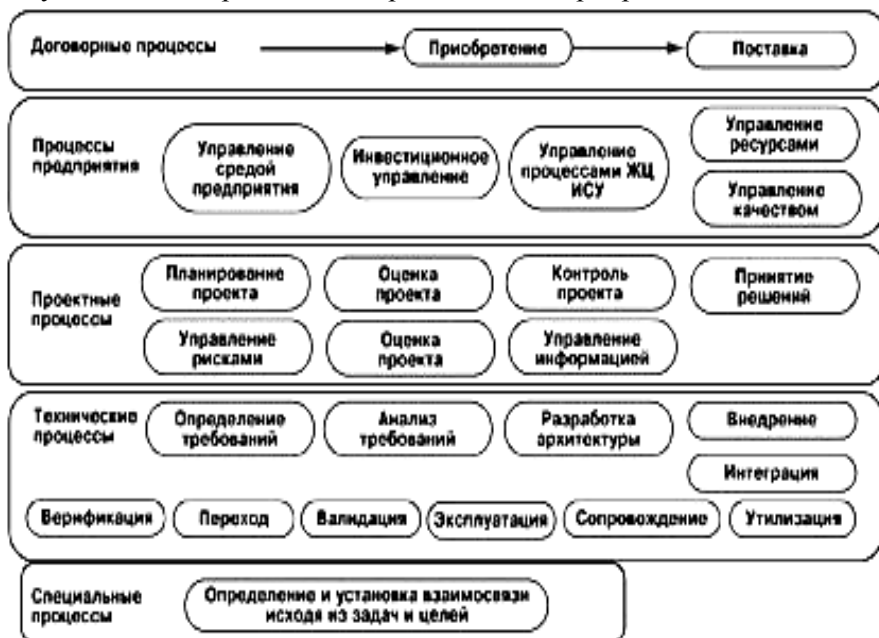


Рис.1.6. Совокупность всех процессов, обеспечивающих ЖЦ ИС
 Поэтому современные ИС разрабатываются на основе стандартов, что позволяет обеспечить, во-первых, их высокую эффективность и, во-вторых, возможность их взаимодействия между собой. Вообще говоря, все стандарты на информационные системы (как и на любые системы вообще) можно разбить на следующие два основных класса:

- **Функциональные стандарты**, определяющие порядок функционирования системы в интересах достижения цели, поставленной перед нею ее создателями.
- **Стандарты жизненного цикла**, определяющие то, как создается, развертывается, применяется и ликвидируется система.

Модели, определяемые стандартами этих двух классов, конечно же взаимосвязаны, однако решают совершенно разные задачи и характеризуются принципиально различными подходами к их построению.

Стоит заметить, что существовавшие еще недавно и популярные в докомпьютерные времена стандарты с приходом информационных систем во все отрасли были признаны недостаточно гибкими и несовершенными. Дело в том, что ранее система управления не включала многих компонентов, которые на данный момент могут считаться наиболее важными в масштабах всей системы.

Системы стали намного сложнее, и ограниченность существующих стандартов стала сказываться на качестве не только проектирования информационных систем управления, но и их эксплуатации, а также прочих этапов их жизненного цикла. Это приводит к мысли о том, что новый уровень использования информационных технологий для управления предприятием требует новых методик и более широкого подхода к понятию "информационная система управления" и жизненного цикла ИСУ на предприятии.

Следует также учитывать, что информационные технологии и их внедрение в во все общественные процессы общества приобрели масштабный мировой характер и уже не вписываются в рамки государственных стандартов. Поэтому регламентация проектной деятельности при создании ПО основывается на стандартах и методологиях как государственных, так и международных.

Среди государственных в настоящее время наиболее популярны стандарты ГОСТ 34-й и 19-й серий, определяющие требования к разрабатываемой документации (они достались нам от СССР), в настоящее время еще используются:

- ГОСТ 34.601-90. Информационная технология. Автоматизированные системы. Стадии создания.
- ГОСТ 34.602-89. Информационная технология. Техническое задание на создание автоматизированной системы.

В настоящее время осуществляется их анализ и адаптация с международными стандартами. В Украине регламентирующими

стандартами в области ЖЦ ПО являются международные стандарты серии ISO/IEC 12207 "Software Life Cycle Processes", а также государственный стандарт ДСТУ 3918-1999 "Процессы жизненного цикла программного обеспечения". В рамках стандарта ISO/IEC 12207 рассматриваются следующие понятия: правила разработки, разработка системы, разработка программных средств, отношения покупатель-продавец, доводка системы. В настоящее время осуществляется стадия FDIS (Final Draft International Standard) в рамках введения в действие первого международного стандарта ISO/IEC серии 15288 (рабочее название: "Управление жизненным циклом. Процессы жизненного цикла системы"), рассматривающего процессы жизненного цикла информационной системы, включающей программные и аппаратные средства, а также организацию взаимодействия людей и бизнес-процессов.

Ниже представлено сравнение процессов ЖЦ ИС, регламентированных стандартом ISO/IEC 12207 и ДСТУ 3918-1999 и стандартом ISO/IEC серии 15288. Как видим, вместе два стандарта ISO/IEC 12207 и 15288 должны представить всестороннюю интегрированную структуру для управления полным жизненным циклом компьютеризированных систем управления, каковыми, несомненно, являются современные ИС. Причем оба стандарта применимы в отношении к предприятиям малого, среднего и большого бизнеса - ограничений нет.

Совместное использование двух стандартов - ISO/IEC 12207 и 15288 - дает целостное представление о ПО и разработке систем, помогает создать структуру процессов, которая может быть кастомизирована для нужд каждого конкретного предприятия. И 12207, и 15288 можно будет использовать как для регламентации жизненного цикла ИСУ и ПО самого предприятия, так и по части планирования совместной деятельности с другими компаниями. Однако некоторые вопросы более точно отражены в старых ГОСТах, например, расчеты трудоемкости работ, некоторая техническая документация. Нам представляется целесообразным не забывать и о них.

Стандарт ISO/IEC 12207 и ДСТУ 3918-1999	Стандарт ISO/IEC 12207 и ДСТУ 3918-1999
<p>Основные процессы:</p> <ol style="list-style-type: none"> 1. Приобретение (заказ). 2. Поставка. 3. Разработка. 4. Эксплуатация. 5. Сопровождение. <p>Вспомогательные процессы, обеспечивающие выполнение основных процессов:</p> <ol style="list-style-type: none"> 1. Документирование. 2. Управление конфигурацией. 3. Обеспечение качества. 4. Верификация. 5. Валидация (аттестация). 6. Оценка (совместный просмотр). 7. Аудит. 8. Решение проблем. <p>Организационные процессы:</p> <ol style="list-style-type: none"> 1. Управление. 2. Создание и сопровождение инфраструктуры. 3. Усовершенствование. 4. Обучение. 	<p>Договорные процессы:</p> <ul style="list-style-type: none"> - приобретение (внутренние или у внешнего поставщика решения); - поставка (внутренние или у внешнего поставщика решения). <p>Процессы предприятия:</p> <ul style="list-style-type: none"> - управление окружающей средой предприятия; - инвестиционное управление; - управление жизненным циклом ИСУ; - управление ресурсами; - управление качеством. <p>Проектные процессы:</p> <ul style="list-style-type: none"> - планирование проекта; - оценка проекта; - контроль проекта; - управление рисками; - управление конфигурацией; - управление информационными потоками; - принятие решений. <p>Технические процессы:</p> <ul style="list-style-type: none"> - определение требований; - анализ требований; - разработка архитектуры; - интеграция; - верификация; - переход; - валидация; - эксплуатация; - сопровождение; - утилизация. <p>Специальные процессы:</p> <ul style="list-style-type: none"> - определение и установка взаимосвязей исходя из задач и целей.

Контрольные вопросы

1. Понятие жизненного цикла ИС и его ключевой компонент.
2. Как называется классический ЖЦ , его основные этапы, достоинства и недостатки.
3. Основное отличие макетирования от каскадной модели ЖЦ, какие обстоятельства вынуждают использовать его?
4. Назовите основные стратегии разработки ПО и дайте им краткую характеристику.
5. Как стратегии разработки ПО взаимосвязаны моделями ЖЦ.
6. Эволюционная стратегия ПО и какая модель ЖЦ лежит в ее основе?
7. Спиральная модель ЖЦ, ее достоинства и недостатки.
8. Что вносит нового в стратегию проектирования ПО компонентно-ориентированная модель разработки ПО?
9. Опишите основные требования к современной технологии проектирования ИС.
10. Какие специалисты включаются в проектную группу проектирования ИС?
11. Какими характеристиками оценивается созданное ПО?
12. Сущность унифицированного процесса проектирования ПО и языка моделирования.
13. Объясните специфические аспекты унифицированного процесса проектирования ПО.
14. Какая модель лежит в основе унифицированного процесса проектирования ПО.
15. Какие основные стандарты проектирования ПО используются в настоящее время?

РАЗДЕЛ 2. МЕТОДИКА ПРОЕКТИРОВАНИЯ ПО НА ОСНОВЕ СЕТЕВОЙ МОДЕЛИ И MS PROJECT

2.1. Понятие о проектах и управлении ими

Еще недавно понятие «проект» в нашей стране относили только к комплекту конструкторских документов, описывающих сооружение или изделие. В последнее десятилетие это понятие приобрело новый смысл, отражающий изменение представлений о современных методах управления. Анализируя работу любой организации, практически всегда можно выделить два вида деятельности:

Процессы – текущие, периодически повторяющиеся операции, такие как работа конвейера или оформление отчетов.

Проекты – последовательность действий (задач), направленная на достижение уникальных результатов, например внедрение новой технологии, написание статьи, проектирование информационной системы (ИС), постройка здания, проведение предвыборной компании, подготовка номера журнала и т.п..

Управление проектами — это процесс планирования, организации и управления задачами и ресурсами с целью достижения определенной цели, обычно при наличии ограничений по времени, ресурсам или затратам.

В структуре управления проектами можно выделить следующие основные компоненты:

- разработка модели проекта;
- формирование и управление рабочей группой;
- управление процессом выполнения проекта;
- управление стоимостью выполнения проекта;
- управление качеством проекта;
- управление информацией проекта;
- управление рисками проекта;
- управление поставками проекта.

Разработка модели проекта предполагает определение целей проекта, разработку бизнес-плана, выполнение операций контроля за его выполнением и корректировку хода выполнения проекта.

Формирование рабочей группы проекта предполагает выполнение операций организационного планирования, разработки функциональных обязанностей членов проектной группы и их ответственности в ходе выполнения проекта.

Управление процессом выполнения проекта базируется на необходимости выполнения календарного плана проекта и дает возможность оперативной корректировки хода выполнения проекта.

Управление стоимостью выполнения проекта происходит на основе предварительно разработанного финансового плана проекта и предполагает постоянный мониторинг финансовых показателей хода выполнения проекта (например, на основе общего баланса проекта).

Управление качеством проекта содержит мероприятия планирования и контроля качества.

Управление информацией проекта предполагает формирование системы распределения информации проекта, определение структуры и содержания технических средств для обработки информации проекта, разработку содержания аналитических отчетов о ходе выполнения проекта, а также управление процессами администрирования информационной системы проекта.

Управление рисками проекта включает определение и планирование рисков, мониторинг рисков, качественный и количественный анализ рисков, планирование ответов на риски.

Управление поставками проекта содержит операции по планированию поставок и заказов, выбору поставщиков, выполнению процессов администрирования контрактов, а также завершению поставок.

2.2. Методы управления проектами и основные этапы

Следует отметить, что существует бесконечное множество путей (способов) перевода системы в желаемое состояние, каждый из которых характеризуется продолжительностью, величиной затрат и качеством результатов. Задачей людей,

осуществляющих руководство проектом, является обеспечение прохождения процесса изменений по оптимальному пути. Для этого существуют определенные принципы и методы, объединенные **методологией управления проектами** (Project Management). Современные методы управления проектами уходят корнями в 50-е годы прошлого столетия. Практически одновременно две проектные группы — Компании Du Pont и Remington Rand — представили метод управления сложными комплексами работ. Он получил название «Метод критического пути» (Critical Path Method — CPM) и стал наиболее распространенным. Суть метода заключается в том, что, зная логику выполнения и длительность работ, рассчитывают наиболее раннюю дату завершения проекта и определяют критический путь, т. е. выявляют работы, которые имеют нулевой резерв времени.

Применение метода **CPM** позволяет представить комплекс взаимосвязанных действий в виде графической нотации сетевой модели - сетевого графика. Сетевой график представляет собой ориентированный граф без контуров, имеющий одну исходную и одну завершающую вершины, в котором вершины поставлены в соответствии некоторым событиям, а дуги - работам. Основным параметром сетевого графика является длительность критического пути - самого продолжительного из путей от

исходного события к завершающему. Важность критического пути определяется тем, что в случае задержки выполнения действий, составляющих критический путь, задерживается срок выполнения всего проекта. Следовательно, чтобы это не произошло, выполнению данных действий следует уделить более пристальное внимание.

Современные представления о методах управления проектами собраны в Своде знаний об управлении проектами (Project Management Body of Knowledge — PMBOK). Структура и содержание PMBOK в разных странах может отличаться, так как многие национальные ассоциации управления проектами имеют неодинаковые точки зрения на содержание этого документа.

Метод CPM положен в основу современных информационных технологий, реализующих управление проектом.

Все этапы планирования являются итеративными процессами и повторяются несколько раз до тех пор, пока не будет получен необходимый результат. Этапы планирования имеют следующее содержание.

Планирование целей (Scope Planing) — разработка документа, в котором определены цели проекта. Отправной точкой служат описание продукта, обоснование проекта, общие ограничения, информация об уже выполненных аналогичных проектах. Анализируются альтернативные пути реализации проекта, определяются критерии успешности. Документ, созданный на этом этапе, является основой для всех проектных решений и единого понимания целей проекта всеми его участниками.

Декомпозиция целей (Scope Definition) — последовательное деление основных результатов проекта на более мелкие элементы, вплоть до пакетов работ, хорошо поддающихся управлению. В итоге получается иерархическая структура (дерево) работ проекта (Work Breakdown Structure — **WBS**).

Определение операций (Activity Definition) — определение перечня элементарных операций (activity), которые должны быть выполнены для достижения результатов, описанных в WBS.

Планирование ресурсов (Resource Planing) — определение того, какие именно ресурсы (люди, оборудование, материалы) и в каком количестве потребуются для выполнения запланированных работ. Учитываются ограничения, связанные с политикой компании по кадровым вопросам, уровнем запасов, использованием оборудования и т. д., а также (обязательно) оценочные данные о стоимости использования ресурсов.

Определение взаимосвязи операций (Activity Sequencing) — определение последовательности проведения работ в проекте с учетом технологических, организационных и других ограничений. Одни работы могут выполняться параллельно, другие же, напротив, могут начаться не раньше, чем завершатся предшествующие. Результатом этого этапа является сетевая диаграмма (Project Network Diagram), которая показывает логическую взаимосвязь между работами в проекте.

Оценка длительности операций (Activity Duration Estimating) — определение количества рабочего времени, которое необходимо для выполнения каждой элементарной операции. Расчет времени

производится на основании экспертных оценок и моделирования (метод Монте-Карло). Учитываются ресурсные и другие ограничения.

Оценка стоимости (Cost Estimating) — определение стоимости ресурсов, необходимых для выполнения проекта. Рассматриваются различные ценовые альтернативы, чтобы стоимости проекта не вышла за рамки ограничений. В результате разрабатывается план управления стоимостью проекта.

Составление расписания (Schedule Development) — определение дат начала и окончания для всех работ проекта. Оцениваются реалистичность расписания (project schedule), загрузка ресурсов и их влияние на срок выполнения проекта.

Разработка бюджета (Cost Budgeting) — определение базисной линии стоимости проекта, называемой S-кривой из-за ее сходства с латинской буквой S. Базисная линия показывает распределение во времени (нарастающим итогом) расходов на проект и служит для сравнения текущих результатов с плановыми.

Разработка плана проекта (Project Plan Development) — создание итогового структурированного документа на основании данных, полученных на предыдущих этапах планирования. Результатом является **план проекта**, который служит руководством для исполнения и управления проектом.

Составленный план приходится неоднократно корректировать в процессе выполнения проекта. Процесс корректировки заключается в повторном выполнении subprocessов планирования на основании дополнительной информации о ходе выполнения проекта, например о выполненных объемах и произведенных затратах для автоматизации подсчета оставшихся объемов, длительностей и затрат и др.

Корректировка необходима, если нарушены сроки выполнения критических операций либо изменились цели и состав работ проекта. Если задержка выполнения отдельных операций не превышает резервов времени, то корректировка необязательна, поскольку сроки выполнения проекта и отдельных его этапов могут быть соблюдены.

Составляющие проектного плана.

Проектный план содержит три основных элемента: задачи (task), ресурсы (resource) и назначения (assignment).

Задачи

Задачей называется работа, осуществляемая в рамках проекта для достижения определенного результата. Поскольку обычно проект содержит много задач, то для удобства отслеживания плана их объединяют в группы, или фазы. Совокупность фаз проекта называется его **жизненным циклом**.

Задачи, в результате исполнения которых достигаются промежуточные или конечные цели, называются **завершающими задачами**. В MS PROJECT они называются **вехами** (milestone).

Длительность задачи – это период времени, который необходим для того, чтобы ее выполнить. Длительность может не соответствовать трудозатратам занимающегося задачей сотрудника. Длительность (Duration) соответствует времени, через которое будет получен результат задачи, а трудозатраты (Work) – времени, затраченному сотрудниками на получение результата.

Задачи в плане проекта взаимосвязаны, например, часто одна задача не может начаться, пока не закончена другая. На плане проекта зависимости (Dependencies) обозначаются с помощью связей (Link), и оба этих термина – зависимость и связь – используются с одним и тем же смыслом, обозначая логику, определяющую последовательность работ в плане проекта.

Ресурсы

Под **ресурсами** в MS PROJECT понимаются сотрудники и оборудование, необходимые для выполнения проектных работ. Каждый сотрудник, участвующий в проекте, получает определенную **роль** в соответствии со своей квалификацией, требованиями проекта и регламентами, действующими в организации. При составлении списка ресурсов часто используется ролевое планирование.

Важное свойство ресурсов – **стоимость** (Cost (затраты)) их использования в проекте. В MS PROJECT есть два типа стоимости ресурсов: повременная ставка и стоимость за

использование. **Повременная ставка** (rate) выражается в стоимости использования ресурса в единицу времени. В таком случае стоимость участия ресурса в проекте составит время, которое он работает в проекте, умноженное на почасовую ставку. Величина **затрат на использование** (cost per use) обозначает стоимость использования оборудования или сотрудника в задаче, которая не зависит от того, сколько времени задействован в задаче сотрудник или материальный ресурс. Общие затраты на использование ресурса определяются путем умножения стоимости использования на число задач, в которых он задействован. При определении общих затрат на использование ресурса в проекте MS PROJECT определяет повременные затраты и затраты на использование и суммирует их.

Назначение

Назначение – это связь определенной задачи и ресурсов, необходимых для ее выполнения. При этом на одну задачу могут быть назначено несколько ресурсов, причем как материальных, так и нематериальных. Назначения объединяют в плане ресурсы и задачи, делая план целостным. Благодаря назначениям решается целый ряд задач планирования.

Во-первых, определяются ответственные за исполнение задач.

Во-вторых, когда определены задачи, за которые отвечает ресурс, можно рассчитать общий объем времени, затрачиваемый на проект, т.е. его стоимость.

В-третьих, определив стоимость участия всех ресурсов в проекте, можно подсчитать его общую стоимость.

Наконец, назначая ресурсы на задачи, можно сокращать срок выполнения работ, выделяя на них больше ресурсов и тем самым сокращая общую длительность проекта.

Треугольник проекта.

Как мы уже знаем, большинство проектов имеют определенную дату окончания, бюджет и объем работ. Это трио времени, денег и объема работ часто называют **проектным**

треугольником, потому что при внесении изменений в один из этих элементов меняются оба других для проекта в равной степени важны все три элемента, как правило, только один из них в зависимости от приоритетов имеет наибольшее влияние на другие (рис. 1.2).

Например, если вы решите изменить план проекта, укоротив расписание, то возрастет стоимость проекта (если вы решите привлечь дополнительных работников) или уменьшится объем работ. Если же изменить план проекта с целью уменьшения его бюджета, то может возрасти длительность проекта и уменьшится объем работ. Наконец, если вы увеличите объем работ, то проект будет длиться дольше и стоить дороже.

То, как изменения в плане влияют на другие стороны треугольника, зависит от обстоятельств и специфики проекта. В некоторых случаях сокращение времени увеличивает стоимость, а в других — уменьшает.

При создании плана вы можете столкнуться с тем, что план не удовлетворяет ожиданиям, например, проект заканчивается слишком поздно или его стоимость превышает допустимые пределы. В таком случае план нужно оптимизировать, чтобы привести его в соответствие ожиданиям.

Когда вы начинаете оптимизировать план, постоянно помните обо всех элементах треугольника и о том, что когда вы изменяете одну из сторон, это затрагивает две другие — позитивно или негативно, в зависимости от вашего проекта. И проверяйте два других элемента треугольника, чтобы быть уверенным, что изменения не делают план невыполнимым. Например, если вы изменили свой план с целью уменьшить расходы, проверьте, что дата окончания проекта все еще находится в допустимых пределах.

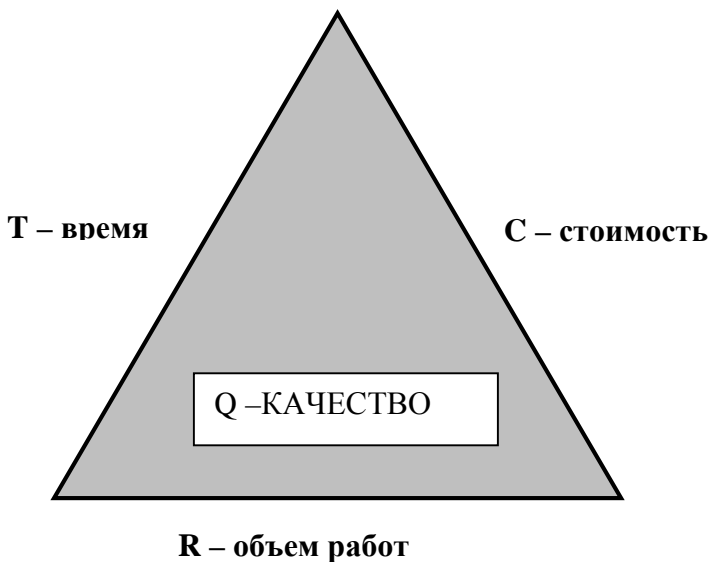


Рис. 1.2. Проектный треугольник – графическое представление специфики управления проектом

Качество, четвертый элемент проектного треугольника, находится в его центре, и изменения, вносимые в любую из сторон треугольника, практически всегда влияют на качество. Качество не является стороной треугольника — он результат ваших действий со временем, стоимостью и объемом работ.

Например, если вы нашли лишнее время в расписании, то можете увеличить объем работ, добавив задачи и увеличив длительность проекта. С этими дополнительными задачами и временем вы сможете добиться более высокого качества в проекте и произведенном продукте или услуге.

Если же вы хотите снизить расходы, чтобы уложиться в бюджет, возможно, вам понадобится уменьшить объем работ, убрав некоторые из задач или уменьшив их длительность. С уменьшенным объемом работ у проекта будет меньше шансов выйти на требуемый уровень качества, поэтому снижение расходов может привести к ухудшению качества проекта.

Формирование проектной группы

Для организации работы над проектом создают группы сотрудников (уровни пользователей), ответственных за формирование соответствующей части проекта:

1. Высшее руководство (Executive Level), т. е. специалисты, отвечающие за постановку целей, задач, критериев проекта и оценку выполнения этих показателей.
2. Менеджеры (Strategic Level) — ответственные за разработку детальных планов достижения целей, поставленных высшим руководством, распределение работ по конкретным исполнителям, планирование использования ресурсов, контроль за выполнением планов и подготовкой укрупненных отчетов для высшего руководства.
3. Специалисты на местах (Desktop Level) — ответственные за выполнение определенных работ в соответствии с графиком предоставления отчетов о состоянии выполняемых работ, их качестве, доступности, загрузке ресурсов и т. д.

Проектная группа формируется для работы над проектом. Иногда ее определяют как «команда проекта». Состав рабочей группы, как количественный, так и качественный, определяется тем объемом работ, который необходимо выполнить для достижения целей проекта.

Для построения эффективной команды проекта и управления ею необходимо обратить внимание на следующие основные процессы: организационное планирование, набор штата и разработку команды.

Организационное планирование предусматривает разделение функциональных ролей в группе, подготовку должностных инструкций, определение ответственности и отчетности исполнителей проекта.

Процессы разработки группы проекта включают работу по повышению индивидуальной и групповой компетентности.

Система коммуникаций в процессе проектирования

Управление коммуникациями в проекте включает процессы, необходимые для гарантии своевременного и соответствующего создания, сбора, преобразования, распространения и хранения проектной информации. Оно обеспечивает важные связи между

людьми и информацией, которая необходима для достижения успеха. Каждый участвующий в проекте должен быть подготовлен к использованию технологий информационных коммуникаций и должен понимать, как обмен информацией, в котором он участвует как индивидуальное лицо, влияет на проект в целом.

Управление информационными коммуникациями в проекте состоит из следующих основных составляющих: планирование информационных ресурсов, распределение информации, подготовка отчетных документов.

В модели информационной коммуникации должны быть определены множество формируемых отчетов проекта и их структура. Обычно они создаются на регулярной основе: ежемесячные, еженедельные, ежедневные.

Управление рисками проекта

Управление рисками проекта — систематический процесс идентификации, анализа рисков. Подразумевает и планирования ответов на них. Управление рисками проекта включает увеличение вероятности и важности положительных событий и уменьшение вероятности и важности неблагоприятных событий при достижении целей проекта.

Управление рисками предполагает выполнение следующих основных работ: планирование, идентификация, качественный; и количественный анализ рисков, планирование ответов на' риски, мониторинг и контроль рисков.

Действия в плане управления рисками по уровню, типу и различиям должны быть соизмеримы с рисками и важностью проекта для организации. План управления рисками должен включать:

- описание технологий, инструментов и источников данных, которые могут быть использованы для управления рисками в данном проекте;
- распределение ролей и ответственности между участниками группы проекта для каждого типа действий в плане управления рисками;
- определение бюджета управления рисками;
- хронометраж процесса управления рисками, который определяет, как часто эти процессы выполняются в течение жизненного цикла проекта.

Риски, воздействующие на проект, можно разделить на категории, отражающие общие источники рисков для прикладной области (отрасли). Различают: технические, управленческие, организационные, внешние риски и др.

Эффективность управления проектом

Для анализа эффективности выполнения работ в проекте прежде всего необходимо выполнить ресурсное планирование, оценку стоимости проекта, бюджетирование, процессы контроля стоимости проекта.

Выполнение операций контроля за ходом выполнения проекта предусматривает постоянную корректировку, поэтому важной составляющей контроллинга является применение современных информационных систем управления проектами.

Фактически во всех программных продуктах используется определенный набор данных при анализе эффективности выполнения работ проекта. Среди них: бюджетная стоимость намеченных работ (Budgeted Cost of Work Scheduled — BCWS), бюджетная стоимость выполненных работ (Budgeted Cost of Work Performed — BCWP), фактическая стоимость выполненных работ (Actual of Work Performed — ACWP), отклонение от графика (Schedule Variance — SV), отклонения от стоимости (Cost Variance — CV), бюджет при завершении (Budgeted at Completion — BAC), оценка при завершении (Estimate at Completion — EAC), отклонения при завершении (Variance at Completion — VAC).

Характеристика систем управления проектами

Успешное управление проектом предполагает постоянную коррекцию содержания работ проекта в зависимости от условий среды, в которой выполняется проект. Необходимость коррекции проектных заданий несет за собой не только изменение графика их выполнения, но и необходимость подготовки новой документации, передача ее исполнителям и т.п. Здесь важным фактором становится время реакции проекта на возникновение незапланированных ситуаций. Для уменьшения этого времени используют методы компьютерной технологии, базирующиеся на специализированном

программном обеспечении, которое определяется как пакет управления проектами.

В настоящее время на рынке программных продуктов имеются различные программные средства, реализующие сетевые методы планирования, - от мощных профессиональных систем до систем, позволяющих эффективно распорядиться рабочим временем, финансовыми средствами и т.п. Данные программные средства можно разделить по следующим категориям:

- профессиональные системы планирования;
- системы планирования среднего класса;
- системы быстрого планирования;
- органайзеры (планировщики).

Особенностью профессиональных систем является: большое количество планируемых задач (до нескольких десятков тысяч), способность поддерживать несколько уровней детализации описания проектов, использование сложных методов оптимизации расписания задач проекта и распределения нескольких различных видов ресурсов. Примерами профессиональных систем могут служить Artemis Project фирмы Metier, Primavera Project Planner фирмы Primavera Systems, Open Plan фирмы Welcom Software, Project Manager Workbench фирмы Applied Business Technology Corporation.

Системы среднего класса предназначены для управления средними проектами, позволяющими планировать и управлять выполнением около 10000 задач. Наиболее распространенными программными продуктами данного класса являются Time-Line фирмы Symantec и Microsoft Project фирмы Microsoft. Эти системы ориентированы на использование непосредственно руководителями проекта. Основными отличительными чертами для систем данного класса является приемлемый интерфейс и простота оптимизационных алгоритмов, а также достаточный уровень совмещения в них функций экономического учета и анализ затрат на проекты.

Системы быстрого планирования проектов предназначены для менеджеров небольших организаций в бизнесе или разработке (уровень начальника группы или отдела). Часто они являются упрощенными версиями рассмотренных выше систем,

например On Taget фирмы Symantec, реализующими планирование только с использованием диаграмм Ганта.

Все большее внимание пользователей привлекают **программные средства, позволяющие организовать повседневную деятельность**: эффективно распорядиться рабочим временем, спланировать использование финансовых средств. Они обладают возможностью автоматизировать регулярные действия: составление персональных и групповых расписаний, ведение записной книжки. В их состав могут входить календарь, часы, калькулятор и т.п. Наибольшее распространение среди программных средств такого типа получил программный продукт Organizer фирмы Lotus.

Отдельно стоит отметить, что современные программные продукты, предназначенные для разработки программного обеспечения, имеют в своем составе средства планирования работ. При этом разработчики делали попытки реализовать принципы искусственного интеллекта: работу с нечеткими моделями данных, правилами принятия решений. Обычно такие системы предполагается разрабатывать с использованием специальных языков высокого уровня, позволяющих разрабатывать и, главное, сопровождать системы, начиная с функционального и информационного уровней: CASE-ORACLE.

Среди отечественных программных продуктов, реализующих методы сетевого планирования и управления, можно выделить систему ПЛАПС. Пакет прикладных программ "ПЛАПС" предназначен для прогнозирования технико-экономических показателей проекта, автоматизированного формирования плана разработки, формирования отчетных документов. Базой построения данной системы являются современные методы прогнозирования технико-экономических показателей и сетевого планирования (диаграммы Ганта).

2.3. Основные возможности MS PROJECT

Microsoft Project предназначен для управления средними проектами, позволяющими планировать и управлять

выполнением около 10000 задач. Эта система ориентирована на использование непосредственно руководителями проекта. Microsoft Project позволяет представить информацию о работах проектов в формах PERT, GANTT, календаря, позволяющих отобразить работы плана на одном-двух уровнях иерархии (рис. 1.3). Для каждой работы на диаграмме указывается длительность, сроки начала и конца, множество непосредственно предшествующих работ, исполнители, а также другие характеристики. Пользователю дополнительно предоставляются широкие возможности самому выбирать форму и совмещать их на экране монитора.

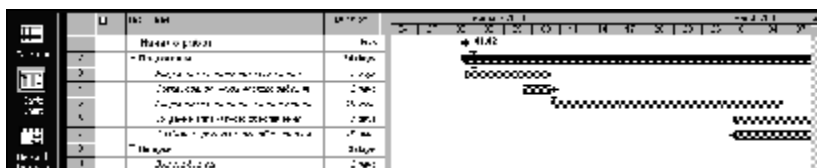


Рис. 1.3. Фрагмент типового плана (диаграмма Ганта)

В качестве основного вида ресурсов в Microsoft Project используются исполнители работ (трудовые ресурсы), на основе которых имеется возможность оценить использование финансовых ресурсов. Для этих целей предусмотрены три типа диаграмм.

Microsoft Project имеет широкий набор готовых для использования отчетов (25 видов), разбитых на шесть групп, а также позволяет пользователю самостоятельно создать тот или иной отчетный документ по выполняемым работам, ресурсами т.д.

Основными отличительными чертами для систем данного класса является приемлемый интерфейс и простота оптимизационных алгоритмов, а также достаточный уровень совмещения в них функций экономического учета и анализ затрат на проекты.

Руководителю проекта необходимо следить за множеством деталей проекта, не упуская из виду конечную цель. Может ли помочь в осуществлении этой задачи программа

Microsoft Project? Во-первых, в базе данных Microsoft Project сохраняются все сведения о проекте. Потом эти сведения используются в Microsoft Project для расчета календарного плана и учета затрат проекта; таким образом создается план проекта. Чем больше задано сведений, тем более точным будет план.

Введенные и рассчитанные в Microsoft Project сведения сохраняются в полях, которые содержат определенные типы данных, например названия или длительности задач. В Microsoft Project каждое поле обычно отображается в виде столбца.

Результаты расчетов в Microsoft Project немедленно отображаются в формате, аналогичном электронной таблице. Если введены все сведения о задаче, можно узнать начальную и конечную даты данной задачи, потребности в ресурсах и конечную дату проекта.

Планирование проекта

После создания плана проекта Microsoft Project рассчитывает и создает рабочий график на основании предоставленных сведений о задачах, которые необходимо выполнить, сотрудников, работающих над этими задачами, оборудовании и материалах, используемых для выполнения задач, а также произведенных затратах.

При создании плана проекта вводятся следующие типы сведений:

- задачи
- длительность
- зависимости задач
- ресурсы
- затраты

На основании этих сведений в Microsoft Project могут быть рассчитаны календарный план, затраты и загрузка ресурсов.

Отслеживание проекта

После создания плана проекта необходимо управлять проектом и следить за ходом его выполнения. Отслеживая

проект, можно вносить необходимые изменения, чтобы придерживаться календарного плана и оставаться в рамках бюджета.

При управлении проектом могут быть использованы следующие средства:

- настраиваемые поля для ввода особых сведений;
- представления и таблицы для отображения точных сведений, требуемых для просмотра;
- фильтры и группы для выделения сведений, требующих внимания.

Вводится фактический ход выполнения задач, фактические данные сравниваются с исходным планом, отслеживается баланс загрузки ресурсов. Все это позволяет эффективно решать возникающие проблемы.

Публикация плана

После создания плана проекта и в процессе управления может потребоваться предоставить сведения о проекте другим пользователям, например заинтересованным сторонам или лицам, назначенным для выполнения работы. Для предоставления сведений могут быть использованы следующие средства:

- печать и создание отчета для предоставления другим пользователям сведений о проекте на бумаге;
- публикация в формате HTML или сохранение плана проекта на веб- сервере для предоставления другим пользователям доступа к сведениям о проекте на веб- узле;
- сервер Microsoft Project Server или система электронной почты для предоставления сведений о проекте другим пользователям.

Совместная работа в группе

Можно упростить и автоматизировать отслеживание загрузки ресурсов и их взаимодействие с помощью средств интерактивной совместной работы в группе, основанных на электронной связи. Кроме того, можно включить в проект обновляемые сведения, т. е. учитывать в текущем плане

последние данные. Предусмотрено два средства для интерактивной совместной работы в Microsoft Project: сервер Microsoft Project Server и его компонент веб-интерфейс Microsoft Project Web Access — дополнительные программные продукты, которые могут быть использованы для организации интерактивного взаимодействия между членами группы, руководителями проекта и другими заинтересованными сторонами.

С помощью сервера Microsoft Project Server и веб-интерфейса Microsoft Project Web Access члены группы могут просматривать собственный список задач или весь проект. Они могут создавать новые задачи для плана проекта и делегировать задачи другим членам. Руководитель проекта может получать обновления задач и текстовые отчеты о состоянии. Заинтересованным сторонам предоставляется доступ к сведениям о проекте и возможность просмотра общей картины.

Электронная почта.

Если требования для совместной работы группы достаточно просты, для обмена основными сведениями о задаче и о состоянии дел с членами группы можно использовать существующую MAPI-совместимую 32-разрядную систему электронной почты. При наличии электронной почты для назначения задач, отправки обновлений и текстовых отчетов о состоянии проекта группа может использовать специальные сообщения, создаваемые Microsoft Project.

Просмотр необходимых сведений.

В базе данных проекта содержится множество сведений, но обычно в данный момент времени необходима только их часть. Для доступа к сведениям используются следующие средства.

- представления показывают часть сведений о проекте в формате, простом для восприятия. Например, на диаграмме Ганта основные сведения представляются столбцами и отрезками на график;
- таблицы определяют отображаемые столбцы;

- фильтры предназначены для отбора определенных задач или ресурсов.

Подобно каналам телевидения, в каждом представлении приводится разная информация. Таблицы и фильтры обеспечивают тонкую настройку. Так же, как при переключении каналов, информация не удаляется, но смена представлений, таблиц или фильтров может скрыть часть сведений. Все они продолжают храниться в базе данных и обновляться. Кроме того, для просмотра распределенных или упорядоченных сведений их можно группировать и сортировать нужным образом.

Лабораторная работа №1. Знакомство с MS Project.

Цель работы: произвести запуск приложения, ознакомиться с интерфейсом программы и основными элементами настроек.

Содержание работы.

Запуск программы Microsoft Project.

Запустите Windows.

Выберите команду меню *Программы - Microsoft Office - Microsoft Office Project 2003.*

Как и у всех программ Windows , в первой строке, рядом с названием программы указывается имя созданного или открытого файла-проекта. По умолчанию Microsoft Project создает первый проект с именем Project1 (Проект1). Как и у всех программ Windows , в первой строке, рядом с названием программы указывается имя созданного или открытого файла-проекта. По умолчанию Microsoft Project создает первый проект с именем Project1 (Проект1).

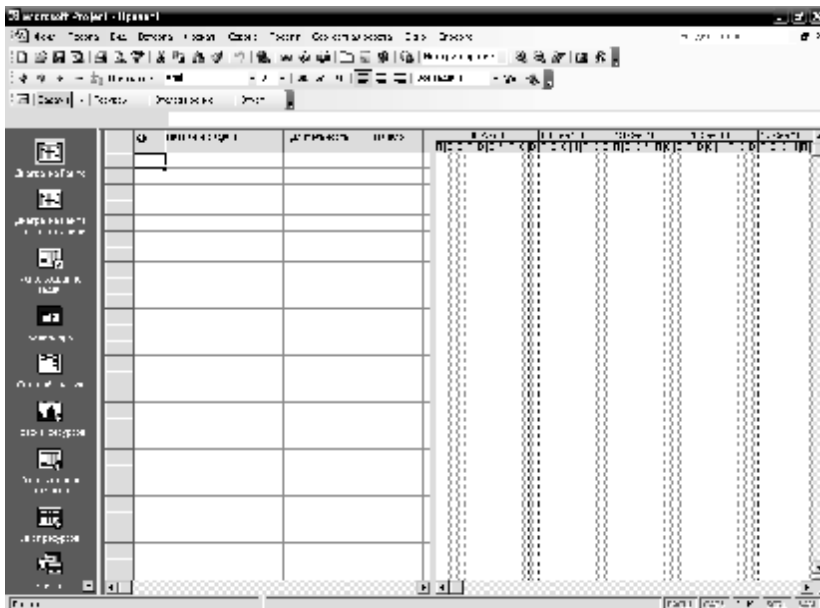


Рис. 1.4. Рабочее окно Microsoft Project.

Когда проект будет сохранен, здесь появится новое имя. Справа от него указывается имя автора проекта - то имя, которое вы ввели при установке программы. (рис.1.4. и 1.5).

Слева от заголовка программы находится значок системного меню, предназначенный для управления окнами. Системное меню содержит команды, позволяющие переместить, свернуть, восстановить окно, изменить его размер, закрыть, переключиться на другую запущенную программу.

Справа - кнопки: для сворачивания, для разворачивания, для закрытия окна. Под заголовком программы расположена полоса меню, предоставляющего доступ ко всем командам, управляющим работой программы. Ниже полосы меню располагаются две основные панели инструментов: Standard (Стандартная) и Formatting (Форматирование).

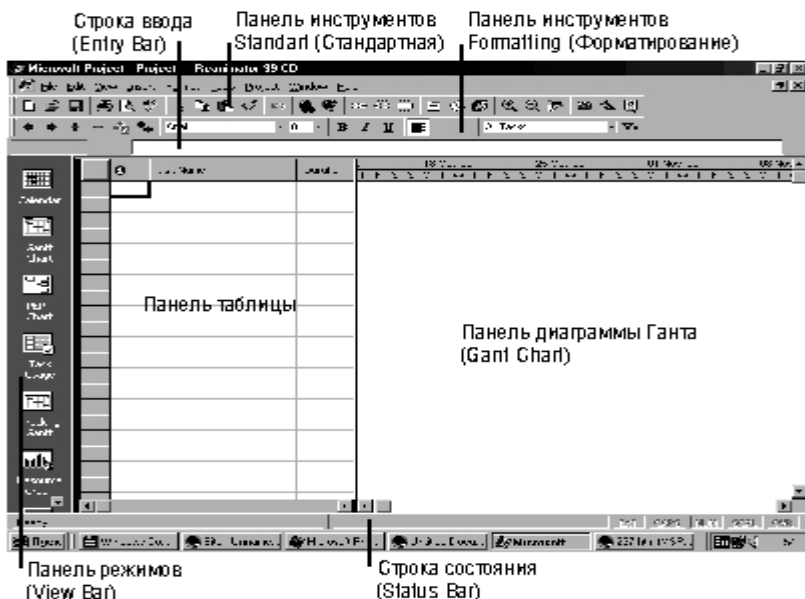


Рис.1.5. Рабочее окно Microsoft Project

Каждая кнопка здесь соответствует одной из наиболее часто применяемых команд меню. Нажатие кнопки позволяет значительно ускорить выбор команды по сравнению с выбором из меню. Панели инструментов можно добавлять из раскрывающегося списка стандартным для Windows способом.

В панели инструментов «**Стандартная**» добавлен ряд кнопок, ниже на рисунке они представлены в порядке следования по номеру:

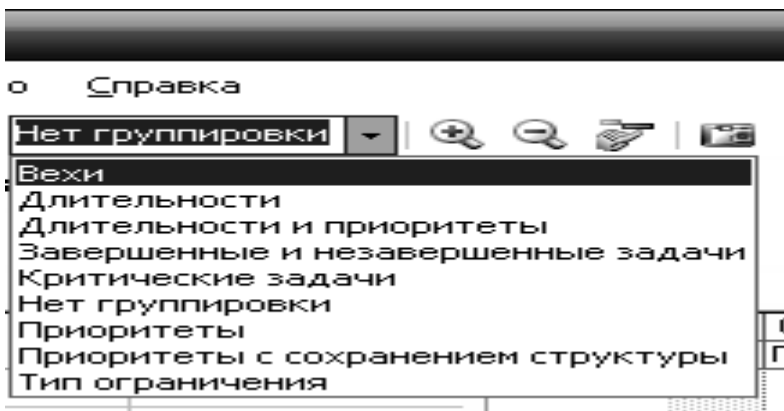
- 1.Связать задачи
- 2.Разорвать связи задач
- 3.Прервать задачу
- 4.Сведения о задаче
- 5.Заметки к задаче
- 6.Назначить ресурсы
- 7.Опубликовать все данные
- 8.Увеличить шаг настройки диаграммы Ганта
- 9.Уменьшить шаг настройки диаграммы Ганта

10. Перейти к выделенной задаче

11. Копировать рисунок

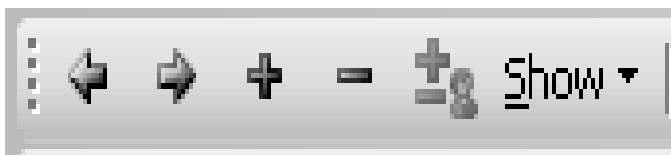


Кнопка **Группировка** содержит раскрывающийся список, позволяющий группировать ресурсы проекта:



- **Завершенные и незавершенные задачи**
- **Тип ограничения**
- **Критические задачи**
- **Длительности**
- **Длительности и приоритеты**
- **Вехи**
- **Нет группировки**
- **Приоритеты**
- **Приоритеты с сохранением структуры**

Панель «**Форматирование**» имеет следующие дополнительные кнопки:



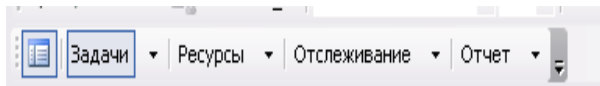
- **На уровень выше**
- **На уровень ниже**
- **Показать подзадачи**
- **Скрыть подзадачи**
- **Скрыть назначения**

Раскрывающийся список **Показать** позволяет выделить задачи определенного уровня: **Все подзадачи**, **Уровень 1 ...**, **Уровень 9**.

Далее на панели «**Форматирование**» следуют:

- Раскрывающийся список **Фильтр**, который позволяет просмотреть определенное подмножество сведений о проекте, отражая только те данные, которые удовлетворяют условиям фильтра. В исходном состоянии установлен фильтр «**Все задачи**» или «**Все ресурсы**»
- **Автофильтр**
- **Мастер диаграммы Ганта**

В исходном состоянии на рабочем окне установлена панель «**Консультант**», с помощью кнопок которой можно контролировать ход проектирования. Первые четыре кнопки **Задачи**, **Ресурсы**, **Отслеживание**, **Отчет** вызывают перечень необходимых действий на поле области задач. Следуя показанным указаниям можно выполнить все необходимые действия по разработке и управлению проектом.



Кнопка «**Показать или скрыть Консультант**» удаляет с экрана окно задач. Для восстановления режима «**Консультант**» достаточно щелкнуть по любой кнопке действий для выполнения проекта на панели.

Под панелями инструментов расположена строка ввода с полем для ввода и редактирования информации. У левого края рабочего окна находится панель режимов, с помощью которой можно выбрать один из множества режимов отображения информации.

В нижней части рабочего окна расположена строка состояния, в которой отображается информация о командах или кнопках, текущих операциях и др. В правой части строки состояния указывается, включен или выключен тот или иной режим:

- **EXT** (Расширение выделения). Если этот режим включен нажатием клавиши [F8], вы можете расширять выделение с помощью стрелочных клавиш.
- **CAPS** - с помощью клавиши [Caps] зафиксирован верхний регистр, что позволяет вводить заглавные буквы, не нажимая клавишу [Shift].
- **NUM** - с помощью клавиши [Num Lock] активизирован числовой блок клавиш в правой части клавиатуры; это позволяет ускорить ввод большого объема числовых данных.
- **SCRL** - если включен режим прокрутки клавишей [Scroll Lock], то с помощью стрелочных клавиш можно осуществлять панорамное перемещение по таблице, а не переход к другой ячейке, как при выключенном режиме прокрутки.
- **OVR** - нажатием клавиши [Insert] включен режим замены. Вводимые с клавиатуры символы будут заменять те, которые находятся справа от курсора, а не отодвигать их, как в режиме вставки.

Остальное пространство рабочего окна программы предназначено для отображения текущего проекта в различных режимах. По умолчанию при первом запуске Microsoft Project устанавливается режим диаграммы Гантта, который используется для составления списка и графика работ. В этом режиме окно проекта делится на две панели: левая отображает информацию в виде таблицы, а правая - в виде горизонтальных полосок-диаграмм на временной шкале.

Если предполагается планирование от конечной даты, то в указанном списке следует выбрать **Дата окончания**. При этом в центре диалога отобразится сообщение **Все задачи начинаются как можно позже**. Затем в открывающемся списке **Дата окончания** следует установить конечную дату.

В открывающемся списке **Дата начала** следует установить дату начала проекта. (Аналогично для конечной даты). По умолчанию здесь предлагается текущая дата. Допустим, мы же предполагаем начать выполнение нашего проекта 5 сентября 2010 года (Каждый устанавливает свою дату начала проекта). При необходимости начальную дату в любой момент можно будет изменить, выбрав команду меню **:Проект - Сведения о проекте**.

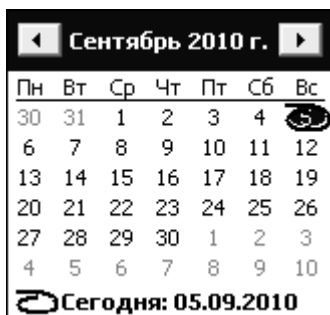




Рис. 1.7. Календарь текущего месяца

Нажмите кнопку  у правой границы поля открывающегося списка **Дата начала**. В окне диалога **Сведения о проекте** для «Проект2» появится календарь текущего месяца (рис. 1.7).

Используя кнопку  на календаре, выберите месяц сентябрь 2010.

Щелчком мыши выберите на календаре дату 5 (5 сентября 2010 года). В поле открывающегося списка **Дата начала** отобразится выбранная дата: **Воскресенье, 5 сентября 2010 года**.

В поле открывающегося списка **Текущая дата** выводится текущая дата. Вы можете изменить ее при необходимости. В открывающемся списке **Календарь** можно выбрать один из

типов предлагаемых программой календарей: **Стандартный ,24 часа, Ночная смена.**

Нажмите кнопку ОК, чтобы закрыть диалог **Сведения о проекте для «Проект2»**. Установки, сделанные в нем, будут использоваться для созданного проекта.

Microsoft Project создает график выполнения работ на основе встроенного календаря. По умолчанию - это базовый календарь **Стандартный (Календарь проекта)**, параметры которого можно легко изменять. Кроме того, вы можете создать новые, индивидуальные календари для каждого работника (ресурса) или группы ресурсов. Например, одна бригада может работать без выходных дней на сдельной оплате труда, а другая - на повременной оплате со всеми выходными днями. Для каждой из них может быть создан индивидуальный календарь, на основании которого будет планироваться выполнение работ.

Лабораторная работа № 2. Общая характеристика проекта ИС

Цель работы:

- научиться определять этапы и работы жизненного цикла проектируемой ИС и их последовательность и взаимосвязи между ними;
- научиться пользоваться стандартами при определении трудоемкости проекта.

Задача: Необходимо разработать программу для решения задач, представленных в виде информационно-логического графа с определением начала и окончания проекта, критического пути, ранних и поздних сроков окончания работ).

Эта задача относится к методоориентированным задачам и не привязана к какой-либо предметной области и разработанное ПО может быть использовано как самостоятельно при решении задач данного типа в любой предметной области либо как

модуль в другом ПО (ГОСТ 19.101-77). Расчетный характер задачи предопределяет систему программирования, в которой она может быть реализована, наиболее приемлемой является система программирования – DELPHI.

Определим ее место в стандартах, определяющих последовательность и трудоемкость выполнения работ проекта., к таким стандартам относятся : ГОСТ 34.601.90 и ГОСТ 4.071.030

В нашей задаче можно все работы отнести к одному этапу: технический проект (ГОСТ 34.601.90) с выделением нескольких подэтапов. Основные работы по реализации проекта представлены в таблице 2.1.

Таблица 2.1. Основные этапы и работы проекта

№ этапа	№ работы	Наименование этапа, работы
1		Технический проект
1.1		Постановка задачи и разработка алгоритма
	1	разработка функциональной схемы программного комплекса;
	2	разработка основных структур данных
	3	разработка алгоритмов определения логической несовместимости операторов и множества взаимно независимых операторов (ВНО)
	4	разработка алгоритмов нахождения ранних и поздних сроков окончания выполнения операторов
	5	разработка алгоритма нахождения необходимого числа процессоров
1.2		Разработка программных модулей
	6	разработка программных модулей для определения логической несовместимости операторов
	7	разработка программных модулей нахождения ранних и поздних сроков окончания выполнения операторов
	8	разработка программного модуля нахождения необходимого числа процессоров

1.3	Разработка интерфейса, тестирование и отладка программных модулей	
	9	тестирование и отладка отдельных модулей
	10	разработка интерфейса программного комплекса
	11	тестирование и отладка всего программного комплекса
	12	Разработка программной документации

Согласно ГОСТ19.101.77 для программ, рассматриваемых как отдельный компонент в качестве документации достаточно представить в качестве спецификации текст программы с необходимыми комментариями и руководство пользователя.

Расчет трудоемкости проекта

Согласно ГОСТ 4.071.030 нормативы трудоёмкости на проектирование ИС дифференцированы в зависимости от:

- технологии обработки информации;
- степени новизны создания ИС;
- сложности задач.

Технология обработки информации зависит от использования:

- локальных массивов;
- баз данных (БД).

Степень новизны определяется характером разработки ИС:
1-я степень - индивидуальная разработка задач с целью развития ИС, разработка типовых проектных решений (ТПР) ИС.

2-я степень - создание ИС на основе внедрения ТПР при условии их частичной доработки; развитие ИС за счет заимствования проектных решений в случае неполного соответствия их условиям предприятия;

3-я степень - привязка ТПР или перевод уже имеющейся программы в новую программно-техническую среду.

Сложность задач и реализующих их программ различают четырьмя группами. Характеристики групп сложности задач и реализующих их программ определяются логической структурой указанных задач и программ (таблицы 2.2 и 2.3).

Таблица 2.2

Группа сложности задачи	Характеристики задачи
1	Алгоритмы, позволяющие решать задачи: оптимального планирования; многофакторного анализа; расчёта нормативов; прогнозирования.
2	Алгоритмы, позволяющие решать задачи: оперативного планирования производством; текущего планирования и регулирования трудовыми и материальными ресурсами; управление технической подготовкой производства; нормативного и аналитического учета
3	Алгоритмы, позволяющие решать задачи синтетического бухгалтерского учета и отчётности.
4	Алгоритмы, позволяющие осуществлять организацию массивов нормативно-справочной информации.

Таблица 2.3

Группа сложности программы	Характеристика программы
1	Сложные логические программы, объединённые в систему программ; организующие программы; программы типа "Диспетчер"
2	Программы сравнительно сложной логической структуры, отличающиеся разнообразием форм входных или выходных документов или сложностью получения данных.
3	Программы несложной логической структуры с простой формой входных и выходных документов, не требующие модификации разработанных алгоритмов для ЭВМ.
4	Программы расчетного характера, реализующие несложные алгоритмы вычислительного характера, не требующие модификации разработанных алгоритмов; программы ведения массивов.

По трудоемкости наша задача может быть оценена по таким показателям:

- технологии --- к задачам с использованием локальных массивов;
- новизне — к задачам 3 степени новизны, так как алгоритм и программа расчета параметров сетевых графиков могут считаться типовыми, ее развитие может быть только в направлении перевода на язык более высокого уровня и совершенствования интерфейса;
- по сложности алгоритма и программирования – к задачам 4-й группы сложности.

Лабораторная работа № 3 ***Определение параметров проекта.***

Цель работы:

- **изучить метод критического пути, позволяющий оптимизировать проектные работы;**
- **составить опорный план проекта вручную с использованием сетевого графика;**
- **составить календарно- сетевой план с использованием MS Project**
- **сравнить результаты двух методов расчета и оценить возможности оптимизации основных параметров проекта при применении MS Project**

Перед использованием MS Project составляется опорный план проекта вручную, который должен содержать следующие параметры: состав этапов и работ, их взаимосвязь между собой, трудоемкость отдельной работы и этапа, общая трудоемкость, расчет количества исполнителей. Эти исходные данные вводятся затем в MS Project и подвергаются оптимизации.

Значение трудоемкости этапов проекта рассчитывались с помощью ГОСТ 4.071.030, а отдельных работ методом экспертных оценок (таблицы 2.4). По ГОСТ определялось количество

нормочасов, для определения объема работ в нормоднях предполагалось, что он включает 8 нормочасов.

Таблица 2.4.

Этап	№ работы	Содержание работы	Нормо часы	Нормо дни
1	Технический проект		759	95
1.1	Постановка задачи и разработка алгоритма		230	29
	1	разработка функциональной схемы программного комплекса;	70	8,8
	2	разработка основных структур данных	14	1,8
	3	разработка алгоритмов определения логической несовместимости операторов и ВНО	50	6,3
	4	разработка алгоритмов нахождения ранних и поздних сроков окончания выполнения операторов	35	4,4
	5	разработка алгоритма нахождения необходимого числа процессоров	61	7,6
1.2	Разработка программных модулей и документации		320	40
	6	разработка программных модулей для определения логической несовместимости операторов	112	14
	7	разработка программных модулей нахождения ранних и поздних сроков окончания выполнения операторов	115	14,4
	8	разработка программного модуля нахождения необходимого числа процессоров	37	4,6

1.3	Разработка интерфейса, тестирование и отладка программных модулей		209	26
	9	тестирование и отладка отдельных модулей	128	16
	10	разработка интерфейса и комплексное тестирование программного комплекса	81	10
	11	Разработка программной документации	56	7

* **Примечание:** Трудоемкость работы № 11 включается в итог этапа 1.2. В целях упрощения схемы эта работа поставлена в конце перечня работ. По ГОСТ 34.601-90 каждый этап завершается формированием соответствующей технической документации и в конце проекта происходит ее окончательное оформление.

Используя данные таблицы, можно определить предварительную длительность выполняемого проекта $Q_p = 759$ нормочасов или **95** нормодней.

Определение численности исполнителей. Средняя численность исполнителей при реализации проекта разработки и внедрения ПО определяется соотношением:

$$N = Q_p / F$$

где Q_p — затраты труда на выполнение проекта (разработка и внедрение ПО), F — фонд рабочего времени.

Величина фонда рабочего времени определяется соотношением: $F = T F_M$, где T — время выполнения проекта в месяцах; F_M — фонд времени в текущем месяце, который рассчитывается из учета общего числа дней в году, числа выходных и

$$F_M = t_p \times \frac{D_K - D_B - D_N}{12},$$

праздничных дней.

где t_p — продолжительность рабочего дня, D_K — общее число дней в году, D_B — число выходных дней в году, D_N — число праздничных дней в году.

Фонд времени в текущем месяце может составить:

$$F_M = 8 * (366 - 104 - 10) / 12 = 168 \text{ часов,}$$

тогда отведя на время выполнения проекта три месяца, определяем, что фонд рабочего времени $F_M = 3 * 168 = 504 \text{ часа}$

Средняя численность исполнителей равна:

$$N = 95 * 8 / 504 = 1,51$$

Таким образом, есть необходимость использовать до двух исполнителей в проекте .

Продолжительность отдельных работ при одновременном выполнении их несколькими исполнителями (T_i) определяется

из соотношения:

$$T_i = T_{pp} / (W_{исп} * K_n), \text{ где}$$

T_{pp} – расчетная продолжительность работы, $W_{исп}$ — количество исполнителей, K_n — коэффициент выполнения нормы.

Продолжительность работ рассчитывается исходя из того, что одной работой занят один исполнитель, а коэффициент выполнения нормы равен единице.

Для точного установления числа исполнителей и целесообразного их распределения по работам, т.е. ролевым назначениям необходимо построить и проанализировать сетевой график проекта.

Сетевая модель проекта и ее анализ

Построение сетевого графика предполагает использование метода сетевого планирования, на базе которого разрабатывается информационно-динамическая модель процесса выполнения проекта. Построение сетевой модели включает оценку степени детализации комплекса работ, определения логической связи между отдельными работами и временные характеристики выполнения этапов проекта

Система сетевого планирования основана на безмасштабном графическом изображении комплекса операций, показывающем технологическую последовательность и логическую

взаимозависимость между всеми работами, направленными на достижение определенной цели.

Всякий намеченный комплекс работ, необходимый для достижения некоторой цели, как мы уже установили, называют **проектом**.

Каждая отдельная работа, входящая в комплекс (проект) требует затрат определенного времени. Некоторые работы могут выполнять только в определенном порядке.

При выполнении комплекса работ всегда можно выделить ряд фаз, т.е. итогов какой-то деятельности, позволяющих приступить к выполнению следующих работ.

Если каждому событию поставить в соответствие вершину графа, а каждой работе – ориентированное ребро, то получится некоторый граф. Он будет отражать последовательность выполнения отдельных работ и наступлений событий в едином комплексе.

Упорядоченная группа дуг, в которой каждая вершина (исключая первую и последнюю) является общей точкой для двух дуг в группе, называется **путем**.

Термин **Работа** в сетевом планировании используется в широком смысле:

- **Действительная работа** – любой трудовой процесс, требующий затрат труда, времени и материальных ресурсов.
- **Ожидание** – пассивный процесс, не требующий затрат труда и материальных ресурсов, но требующий затрат времени (твердение бетона, сушки штукатурки, всходы посевов).
- **Фиктивная работа** – чисто условная зависимость между событиями, которая вводится только для удобства изображения сети. Фиктивная работа не связана с затратами труда, времени, ресурсов. На графике она обычно изображается пунктиром.

Событие определяет факт получения результата. Оно имеет продолжительность во времени, свершение события есть лишь фиксация окончания какого-либо процесса, определяемого данным событием.

- **Исходное событие** - начало выполнения проекта.
- **Завершающее событие** - достижение конечной цели проекта.
- **Промежуточное событие** - результат выполнения одной или нескольких работ, позволяющий приступить к выполнению последующих работ.

Любая стрелка на сетевом графике соединяет только две вершины и отражает процесс перехода от одного события к другому. Поэтому любая работа может быть зашифрована парой чисел, соответствующих предшествующему и последующему событиям.

Время, необходимое для выполнения работы $\langle i, j \rangle$, называют продолжительностью работы и обозначают $t\langle i, j \rangle$. Время проставляется над соответствующей дугой.

Если над ребрами поставить время, необходимое для завершения соответствующей работы, то получится так называемая **сеть**. Изображение такой сети называется **сетевым графиком**.

Сеть, моделирующая определенный процесс, называется сетевой моделью данного процесса, при этом ориентация дуг графа осуществляется в соответствии с логикой данного процесса.

Ориентация дуг, соединяющая вершины графа, осуществляется в направлении достижений результата при реализации данного процесса.

В основе построения сетевого графика лежат три основных понятия: **путь, работа, событие**.

Метод критического пути позволяет рассчитать возможные календарные графики выполнения комплекса работ на основе описанной логической структуры сети и оценок продолжительности выполнения каждой работы, определить критический путь проекта. Максимальный по продолжительности полный путь в сети называется **критическим**; работы, лежащие на этом пути, также называются **критическими**.

В табл.2.4 представлены основные события и работы проекта и связь между ними. Графическое отображение сетевой модели (сетевой график содержит окружности, отображающие основные события проекта, и векторы, соединяющие эти окружности и определяющие необходимость выполнения соответствующих работ. Реальные работы изображаются сплошной линией, фиктивные – штриховой, а работы, лежащие на критическом пути — линией двойной толщины (рис.2.1).

Таблица 2.4 Основные события и работы проекта

№	Событие	Код работы	Работа	Время, t	
0	Начало работ	0-1	Разработка функциональной схемы программного комплекса	70	8,8
1	Функциональная схема программного комплекса разработана	1-2	Разработка основных структур данных	14	1,8
2	Разработаны основные структуры данных	2-3	Разработка алгоритмов определения логической несовместимости операторов и нахождения множества ВНО	50	6,3
		2-4	Разработка алгоритмов нахождения ранних и поздних сроков окончания выполнения операторов	35	4,4
3	Алгоритмы определения несовместимости и нахождения множества ВНО разработаны	3-5	Разработка алгоритма нахождения необходимого числа процессоров	61	7,6

		3-6	Разработка программных модулей для определения логической несовместимости операторов и множества алгоритмов	112	14
4	Алгоритмы нахождения ранних и поздних сроков окончания выполнения операторов разработаны	4-5 4-7	Ожидание окончания работы 3—5. Разработка программных модулей для нахождения ранних и поздних сроков окончания выполнения операторов	0 115	0 14,4
5	Алгоритм нахождения необходимого числа процессоров разработан	5-8	Разработка программного модуля нахождения необходимого числа процессоров	37	4,6
6	Программные модули определения логической несовместимости операторов и множества ВНО разработаны	6-9	Тестирование программных модулей определения логической несовместимости операторов и нахождения множества алгоритмов	48	6
7	Программные модули для нахождения ранних и поздних сроков окончания выполнения операторов	7-9	Тестирование программных модулей для нахождения ранних и поздних сроков окончания выполнения операторов	48	6
8	Программный модуль нахождения необходимого числа процессоров разработан	8-9	Тестирование программного модуля нахождения необходимого числа процессоров	32	4

9	Закончено тестирование и отладка отдельных модулей	9-10	Разработка интерфейса и комплексное тестирование	81	10
10	Интерфейс программного комплекса разработан и проведено тестирование	10-11	Разработка программной документации	56	7
11	Завершена разработка руководства пользователя				

На соответствующих сегментах окружностей показаны номера событий, а на векторах — продолжительность работ, приведенная в столбце *чел./дни* таблицы основных событий и работ проекта (см. табл.2.4). На рис.2.1 показана графическая модель сетевого графика. Следует признать графическое представление графика и его анализ возможны только для проектов, включающих небольшое количество работ и событий, однако этот метод наглядно иллюстрирует логику расчетов, поэтому в пособии он будет использован для расчетов основных параметров сетевой модели.

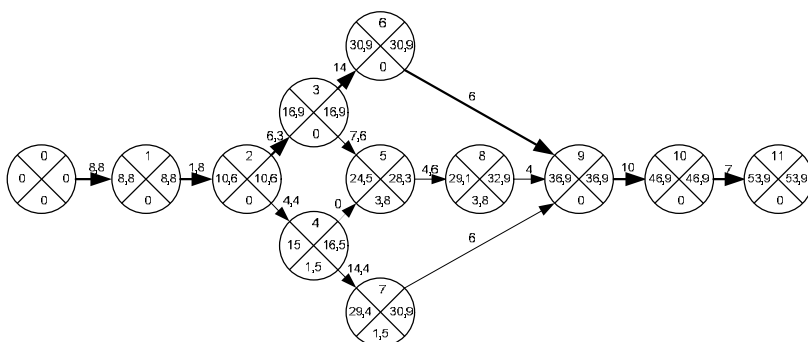


Рис. 2.1 Сетевой график проекта

Графический метод расчета параметров предполагает выполнение всех расчетов непосредственно на графике. Для этого необходимо выполнить некоторое изменение представления сетевой модели (рис.2.2.)

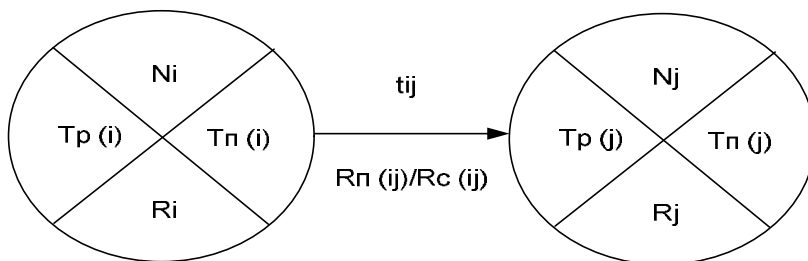


Рис. 2.2. Элемент сетевой модели

Графическое отображение сетевой модели содержит окружности, отображающие основные события проекта, и векторы, соединяющие эти окружности и определяющие необходимость выполнения соответствующих работ. Реальные работы изображаются сплошной линией, фиктивные – штриховой, а работы, лежащие на критическом пути, – линией двойной толщины.

Окружности разделены на четыре сектора, в которых указаны номер данного события (в верхнем секторе), значения раннего срока наступления текущего события (в левом секторе), значение позднего срока наступления события (в правом секторе) и значение резерва времени текущего события (в нижнем секторе).

Обозначения основных элементов графика:

Ni, Nj – номер события,

$t_p(i)$ – ранний срок наступления события i ;

$t_n(i)$ – поздний срок наступления события i ;

Ri – резерв времени события i ;

t_{ij} – продолжительность работы $i - j$;

$R_n(ij)$ – полный резерв времени работы $i - j$;

$R_c(ij)$ – свободный резерв времени работы $i - j$.

При создании сетевого графика в соответствующих сегментах окружностей следует записать номера событий, а на векторах – продолжительность работ. После построения графической модели следует рассчитать оставшиеся параметры сети: сроки наступления событий, резервы времени событий и полный и свободный резервы времени работы.

Расчет параметров события

Для определения резервов времени по событиям сети рассчитывают наиболее ранние $t_p(i)$ и наиболее поздние $t_n(i)$ сроки свершения событий. Любое событие не может наступить прежде, чем свершиться все предшествующие ему события и не будут выполнены все предшествующие работы. Поэтому *ранний* (или *ожидаемый*) срок $t_p(i)$ свершения i -ого события определяется продолжительностью максимального пути, предшествующего этому событию:

$$t_p(j) = \max [t_p(i) + t(i,j)] \quad (1)$$

Здесь максимум берется по всем работам, входящим в событие j , U_j – множество таких работ: $j = 1, 2, \dots, m$. Таким образом, для вычисления раннего срока и наступления события нужно уже вычисленный ранний срок предшествующего события $t_p(i)$ сложить с длительностью работы $t(i,j)$, идущей от предшествующего события к данному.

Задержка свершения события i по отношению к своему раннему сроку не отразится на сроке свершения завершающего события (а значит, и на сроке выполнения комплекса работ) до тех пор, пока сумма срока свершения этого события и продолжительности (длины) максимального из следующих за ним путей не превысит длины критического пути. Поэтому *поздний* (или *предельный*) срок $t_n(i)$ свершения i -ого события равен:

$$t_n(i) = t_{kp} - \max t(L_{ci}) \quad (2)$$

где L_{ci} - любой путь, следующий за i -ым событием, т.е. путь от i -ого до завершающего события (c) сети.

Если событие i имеет несколько последующих путей, а следовательно, несколько последующих событий j , то поздний срок свершения события i удобно находить по формуле:

$$t_n(i) = \min [t_p(j) - t(i,j)] \quad (3)$$

Резерв времени $R(i)$ i -ого события определяется как разность между поздним и ранним сроками его свершения:

$$R(i) = t_n(i) - t_p(i) \quad (4)$$

Резерв времени события показывает, на какой допустимый период времени можно задержать наступление этого события, не вызывая при этом увеличения срока выполнения комплекса работ.

Критические события резервов времени не имеют, так как любая задержка в свершении события, лежащего на критическом пути, вызовет такую же задержку в свершении завершающего события. Таким образом, определив ранний срок наступления завершающего события сети, мы тем самым определяем длину критического пути.

Расчет параметров работ.

Отдельная работа может начаться (и окончиться) в ранние, поздние и другие промежуточные сроки. При оптимизации графика возможно любое размещение работы в заданном интервале.

Очевидно, что ранний срок $t_{pn}(i,j)$ начала работы (i,j) совпадает с ранним сроком наступления начального (предшествующего) события i , то есть:

$$t_{pn}(i,j) = t_p(i). \quad (5)$$

Тогда ранний срок $t_{po}(i,j)$ окончания работы (i,j) определяется по формуле

$$t_{po}(i,j) = t_p(i) + t(i,j). \quad (6)$$

Ни одна работа не может окончиться позже допустимого позднего срока своего конечного события j . Поэтому поздний

срок $t_{но}(i,j)$ окончания работы (i,j) определяется соотношением:

$$t_{но}(i,j) = t_n(j), \quad (7)$$

а поздний срок $t_{пн}(i,j)$ начала этой работы – соотношением

$$t_{пн}(i,j) = t_n(j) - t(i,j). \quad (8)$$

Прежде чем рассматривать резервы времени работ, обратимся к резерву времени пути. Такие резервы имеют все некритические пути. Резерв времени пути определяется как разность между длиной критического и рассматриваемого пути:

$$R(L) = t_{кр} - t(L). \quad (9)$$

Он показывает, насколько в сумме могут быть увеличены продолжительности всех работ, принадлежащих этому пути. Любая из работ пути L на его участке, не совпадающем с критическим путем (замкнутым между двумя событиями критического пути), обладает резервом времени.

Полный резерв времени $R_n(i,j)$ работы (i,j) показывает, насколько можно увеличить время выполнения данной работы при условии, что срок выполнения комплекса работ не изменится. Полный резерв $R_n(i,j)$ определяется по формуле:

$$R_n(i,j) = t_n(j) - t_p(i) - t(i,j). \quad (10)$$

Полный резерв времени работы равен резерву максимального из путей, проходящего через данную работу. Этим резервом можно располагать при выполнении данной работы, если ее начальное событие свершится в самый ранний срок, и можно допустить свершение ее конечного события в самый поздний срок. Важным свойством полного резерва времени работы является то, что он принадлежит не только этой работе, но и всем полным путям, проходящим через нее.

Свободный резерв времени R_c работы (i,j) представляет собой часть полного резерва времени, на которую можно увеличить продолжительность работы, не изменив при этом раннего срока ее конечного события. R_c находится по формуле:

$$R_c(i,j) = R_n(i,j) - R(j). \quad (11)$$

$$\text{или} \quad R_c(i,j) = t_p(j) - t_p(i) - t(i,j). \quad (12)$$

Для критического пути резервы всех его работ равно нулю. На

других путях имеются работы с нулевым резервом времени.

Если увеличить время выполнения работы t_{ij} на её полный резерв $R_n(i,j)$, то работа станет критической. Весь резерв времени при этом будет затрачен на этой работе и остальные работы пути лишатся своих резервов.

Если увеличить время выполнения работ t_{ij} на величину свободного резерва $R_c(i,j)$, то эту работу можно завершить не ранее раннего наступления завершающего события (j). Этот резерв можно использовать также на предшествующих работах (ij) не критического пути. При этом можно заканчивать предшествующие работы раньше раннего срока завершения конечных событий этих работ и за тем поджидать завершения других работ, входящих в данные собрания. Это равносильно увеличению продолжительности работ на величину ожидания.

Работы, лежащие на критическом пути, так же, как и критические события резервов времени не имеют.

В соответствии с содержанием таблицы основных событий и работ проекта построена графическая модель сетевого графика.

В результате исследования определяется критический путь на сетевом графике — путь, имеющий наибольшую суммарную длительность работ. Применительно к данному проекту критический путь проходит через вершины: 0-1-2-3-6-9-10-11 и имеет длину $T_k = 54$ дня.

Ограничения, влияющие на расписание проекта

Если в сетевой модели не учитывать никаких связей и ограничений, все работы могут выполняться одновременно (параллельно), поэтому продолжительность комплекса работ будет равна продолжительности наиболее длинной работы.

Ограничения на последовательность выполнения работ (а тем самым и на сроки) устанавливаются с помощью связей между работами, а только на сроки — с помощью непосредственно накладываемых **временных ограничений** (constraint) Увеличение количества накладываемых ограничений приводит к увеличению общей продолжительности работ.

Установка связей и ограничений на выполнение комплекса работ является важнейшим этапом планирования проекта. Действительно, недостаточно просто перечислить все работы и указать их характеристики, необходимо также правильно охарактеризовать взаимосвязи между работами.

Одна связь может быть установлена только между двумя событиями. Следует помнить, что сроки начала и окончания работ также являются событиями. При этом работа, которую необходимо выполнить прежде рассматриваемой работы, называется **предшествующей** (predecessor), а работа, которая выполняется позже рассматриваемой, называется **последующей** (successor). Сетевая модель позволяет учитывать несколько видов связей:

А. По характеру связываемых событий :

- **Окончание-начало** (ОН - Finish to Start) – между моментами окончания предшествующей работы и начала последующей, это наиболее распространенный тип связи – одна работа следует за другой;
- **Начало-начало** (НН – Start to Start) - между моментами начала предшествующей работы и начала последующей, две работы могут выполняться одновременно;
- **Начало-окончание** (НО – Start to Finish) – между моментами начала предшествующей работы и окончания последующей, одна работа не может окончиться, пока не начнется другая;
- **Окончание-окончание** (ОО – Finish to Finish) – между моментами окончания предшествующей работы и окончания последующей, одна работа не может завершиться, пока не закончится другая.

Б. По соотношению между связываемыми событиями :

- **Связи непосредственного следования** (с нулевым растяжением) – **последующее** событие может наступить сразу же после свершения предшествующего;
- **Связи с задержкой** (lag - с положительным растяжением) – последующее событие может наступить не

ранее, чем через некоторый период времени после свершения предшествующего события. Связь с задержкой указывается с плюсом в единицах времени или в процентах по отношению к длительности работы. Например: +2дня или +20%.

- **Связи с опережением** (lead - с отрицательным растяжением) - последующее событие может наступить за некоторый период времени после свершения предшествующего события. Связь с опережением указывается с минусом в единицах времени или в процентах к длительности. Например: -2дня или -20%

Создание календарного графика выполнения проекта с применением MS Project.

Календарный график последовательности проводимых работ проекта построим в MS Project. За дату начала проекта примем – **01.10.07**. (В дальнейшем при использовании учебного пособия будем использовать текущую дату)

Календарь стандартный. Эти данные необходимо ввести в **Сведения о проекте**

На основании таблицы 2.4 создадим таблицу задач проекта, удобную для ввода в MS Project (табл. 2.5.)

Таблица 2.5.

№ п/п	Наименование задачи	Длительность		Связь с предшествующими задачами и тип связи
		в часах	в днях	
1	Начало	0	0	
Разработка архитектуры ПО				
2	Разработка функциональной схемы программного комплекса	70	8,8	1 ОН
3	Разработка основных структур данных	14	1,8	2 ОН
Разработка алгоритмов				
4	Разработка алгоритмов определения логической несовместимости операторов и нахождения множества ВНО	50	6,3	3 ОН
5	Разработка алгоритмов нахождения ранних и	35	4,4	3 ОН

	поздних сроков окончания выполнения операторов			
6	Разработка алгоритма нахождения необходимого числа процессоров	61	7,6	4, 5 ОН
Разработка программных модулей				
7	Разработка программных модулей для определения логической несовместимости операторов и множества ВНО	112	14	4 ОН
8	Разработка программных модулей для нахождения ранних и поздних сроков окончания выполнения операторов	115	14,4	5 ОН
9	Разработка программного модуля нахождения необходимого числа процессоров	37	4,6	6 ОН
Тестирование программных модулей				
10	Тестирование программных модулей определения логической несовместимости операторов и множества ВНО	48	6	7 ОН
11	Тестирование программных модулей для нахождения ранних и поздних сроков окончания выполнения операторов	48	6	8 ОН
12	Тестирование программного модуля нахождения необходимого числа процессоров	32	4	9 ОН
Комплексное тестирование, создание интерфейса и документации				
13	Разработка интерфейса и комплексное тестирование	81	10	10,11,12 ОН
14	Разработка программной документации	56	7	13 ОН
15	Завершение проекта	0	0	14 ОН

Примечание: В проекте между всеми задачами установлен тип связи ОН (окончание-начало).

Результат процесса разработки – календарно-сетевой график показан на рис.2.3. (диаграмма Ганта).

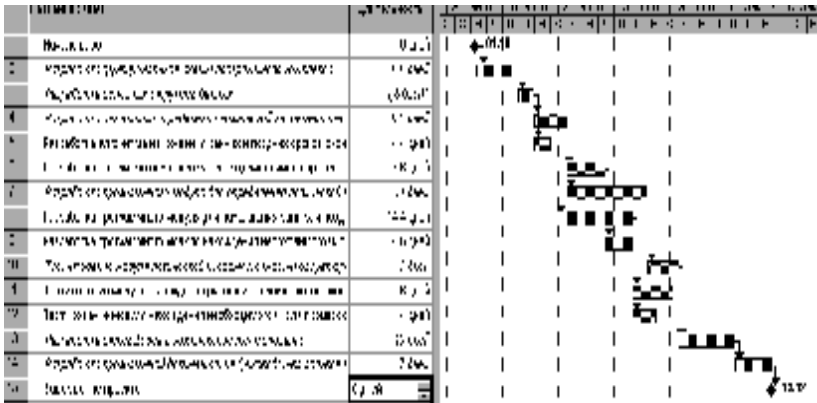


Рис.2.3. Календарно-сетевой график проекта (диаграмма Ганта)

Его данные в основном совпадают с данными сетевого графика, разница – в более точном расчете резервов времени.

Рекомендации по созданию календарно- сетевого графика.

Ввод работ.

Добавление работ и их продолжительности рассмотрим на примере первой задачи – **Разработка функциональной схемы программного комплекса (табл.2.5)**. Выделите ячейку щелчком активной клавиши манипулятора в поле **Task name (Название задачи)** первой строки. Введите первую задачу «**Разработка функциональной схемы программного комплекса**». Клавишами управления курсора переместите табличный указатель в соседнюю ячейку **Duration (Длительность)** и замените, выставленную по умолчанию, продолжительность работы 1 день (**1 day**) на **8,8 дней**. Таким же образом введите все работы из таблицы.

Примечание: Задачи «**Начало**» и «**Завершение проекта**» являются не функциональными задачами, а контрольными точками и вводятся позже.

Добавление контрольных точек.

Чтобы иметь возможность получения промежуточных итогов проекта необходимо добавлять контрольные точки – фиктивную работу длительностью 0 дней. Для этого переместите табличный курсор в первую строку поля **Tasks** (Наименование задачи). Выберите **Insert** (Вставить)– **New Task** (Новую задачу). В таблице появится новая строка . Введите значение для поля **Tasks** (Наименование задачи) **Duration** (Длительность) – **Начало работ** и **0** дней. Переместите курсор в конец списка работ и создайте такую же контрольную точку **Завершение проекта**.

В результате Вы получите скелетный план проекта, в котором будут отражены последовательность работ и их длительность.

Сохраните проект.

Создание графика работ.

В любом проекте все виды работ взаимосвязаны и должны выполняться в определенной последовательности. В MS Project создание графика работ сводится к установлению связи между задачами. Связи между задачами представлены в таблице 2.5 в последней графе, в ней указаны номер предшествующей задачи и тип связи.

Формирование связей между задачами.

Существует несколько способов формирования связей. Ниже писаны самые распространенные способы.

Щелчком активной клавиши манипулятора выделите первую строку **Начало работ**. Удерживая клавишу **Shift**, выделите вторую строку – **Разработка функциональной схемы программного комплекса**. Щелчком по кнопке **Link Tasks** (Связь задач) в панели инструментов создайте связь **Finish-to-Start (FS)** (Окончание-начало(OH)).

Перейдите в ячейку поля **Predecessors** (Предшественники) строки **Разработка основных структур данных**. Двойным щелчком в ячейке откройте окно **Task Information** (Сведения о задаче). Откройте выпадающий список ячейки **Task Name** (Название задачи) первой строки табличной части

диалога. Выберите предшествующую задачу по номеру (2) – **Разработка функциональной схемы программного комплекса** – и установите тип связи (**Окончание-начало(ОН)**). Закройте окно, сохранив изменения.

Подобным образом установите связи с предшественниками для всех задач проекта.

Выделение контрольных точек (wex).

Установите курсор в поле **Task Name** (Название задачи) строки *Начало работ*. Вызовите диалог **Font** (Шрифт) из меню **Format** (Формат). Измените цвет шрифта на **Зеленый**, начертание на **Жирный**, размер на **10**. Закройте окно **Font** (Шрифт) кнопкой **Ок**.

Оставив табличный курсор в строке *Начало работ*, нажмите кнопку **Format Painter** (Формат Художник) на инструментальной панели **Standart** (Стандартная) и щелчком манипулятора в строке *Завершение проекта* перенесите формат текста.

Выделение критических работ.

Откройте диалог **Text Styles** (Стили Текста) меню **Format** (Формат). В качестве значения параметра **Item to Change** (Изменяемый элемент) выберите **Critical Tasks** (Критические задачи). Измените цвет шрифта на **Красный**, а начертание на **Курсив**. Сохраните изменения.

Выделение критического пути на диаграмме.

Двойным щелчком на панели диаграммы вызовите диалог **Bar Styles** (Стили бруска). Установите табличный курсор в строку **Task** (Задача) поля **Name** (Название). Щелчком манипулятора по кнопке **Insert Row** (Добавить строку) добавьте в таблицу новую строку. В поле **Name** (Название) введите имя – **Critical Task** (Критическая задача). Стрелками управления курсором перейдите в поле **Appearance** (Вид). Используя раздел **Middle** (Середина) закладки **Bar** (Отрезки), замените цвет заполнения на **Red** (Красный). Перейдите в поле **Show for...Tasks** (Показывать для задач) и выберите значение **Critical** (Критическая). Опустите табличный курсор на строку **Task**

(Задача) и измените параметр **Show for...Tasks** (Отображать для след. задач) на **Noncritical** (Не критические). Закройте диалог **Bar Styles** (Стили бруска) с сохранением изменений. Сравните полученный результат с рисунком 2.5. У Вас сформировался календарный план в виде диаграммы Ганта, его можно рассматривать как опорный план, на основе которого можно определять и оптимизировать остальные параметры календарно-сетевого плана.

Анализ сетевого графика и назначение исполнителей. На данном этапе руководителя проекта интересуют параметры, которые представлены в таблицах 2.5 и 2.6. Их можно получить из таблиц и отчетов MS Project:

Данные для таблицы 2.5 – Проект\Сведения о проекте\Статистика; Вид\Отчеты\Обзорные\Критические задачи.

Данные для таблицы 2.6. – Вид\Таблицы\ Календарный план

Таблица 2.5.

№ п\п	Параметры проекта	Значения параметров
1	Дата начала проекта	1.10.07
2	Дата окончания проекта	13.12.07
3	Длительность проекта:	
	в календарных днях	74
	в нормоднях	54
4	Критический путь	1-2-3-4-7-10-13-14-15

Таблица 2.6.

Название работ	Свободный резерв времени	Общий резерв времени
Начало работ	0 дней	0 дней
Разработка функциональной схемы программного комплекса	0 дней	0 дней
Разработка основных структур данных	0 дней	0 дней
Разработка алгоритма определения логической несовместимости операторов и ВНО	0 дней	0 дней
Разработка алгоритма нахождения ранних и поздних сроков окончания выполнения операторов	0 дней	1,5 дней

Разработка алгоритма нахождения необходимого числа процессоров	0 дней	3,8 дней
Разработка программного модуля для определения логической несовместимости операторов и множества алгоритмов	0 дней	0 дней
Разработка программного модуля для нахождения ранних и поздних сроков	0 дней	1,5 дней
Разработка программного модуля нахождения необходимого числа процессоров	0 дней	3,8 дней
Тестирование модуля логической несовместимости операторов и ВНО	0 дней	0 дней
Тестирование модуля нахождения ранних и поздних сроков окончания выполнения операторов	1,5 дней	1,5 дней
Тестирование модуля нахождения необходимого числа процессоров	3,8 дней	3,8 дней
Разработка интерфейса и комплексное тестирование	0 дней	0 дней
Разработка программной документации (руководства пользователя)	0 дней	0 дней
Завершение проекта	0 дней	0 дней

Выполним анализ сетевого графика (**Вид\Сетевой график**) или используется **Панель представлений\Сетевой график**.

Анализ графика с технологической точки зрения показал, что в нем можно выделить следующие блоки (рис3.1).:

- 1. Постановка задачи.** Он включает разработку функциональной схемы ПК и разработку структуры данных (задачи 2 и 3).
- 2. Разработка программного модуля определения логической несовместимости операторов и ВНО.** Он включает: разработку алгоритма, программы и ее тестирование (задачи 4, 7, 10).
- 3. Разработка программного модуля для нахождения ранних и поздних сроков окончания выполнения операторов.** Он включает: разработку алгоритма, программы и ее тестирование (задачи 5,8,11)..
- 4. Разработка программного модуля нахождения необходимого числа процессоров.** Он включает: разработку алгоритма, программы и ее тестирование (задачи 6,9,12)..
- 5. Сборка ПК и разработка программной документации.**

Он включает: разработку интерфейса и комплексное тестирование и разработку документации (задачи 13 и 14)

Для удобства контроля за выполнением работ рекомендуется разбить их на группы прямо на диаграмме Ганта.

На сетевом графике видно, что после выполнения задач первого блока процесс распараллеливается на три пути: 1 путь – 4-7-10; 2 путь – 5-8-11; 3 путь – 6-9-12) (рис. 2.4).

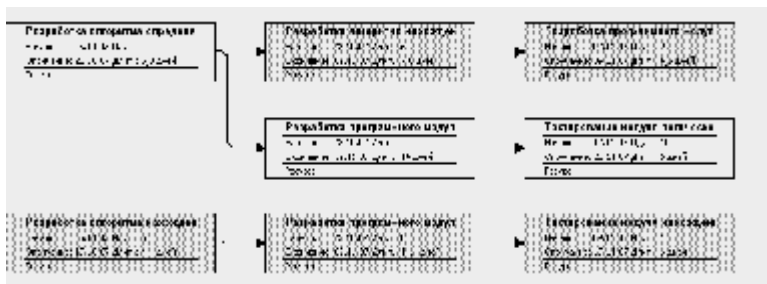


Рис. 2.4. Фрагмент сетевого графика, где процесс распараллеливается

Анализ сетевого графика дает нам возможность выполнить ролевое распределение работ следующим образом:

1. Выделить ведущего программиста, он же и руководитель проекта, ему будет поручено выполнение наиболее важных блоков 1, 2 и 5 и как руководителю – общая ответственность за проект. Он будет принят на эти роли на все время выполнения проекта.
2. Выделить 2 программистов для выполнения блоков 3 и 4, они будут приняты в проект только на время выполнения работ блоков 3 и 4.

Распределение ролей исполнителей, время работы в проекте и тарифные ставки показаны в таблице 3.7.

№ п/п	Должности	Даты начала и окончания работы (доступность ресурса)	Перечень задач, в которых участвует исполнитель	Стандартная ставка, грн/час	Степень использования (макс. единиц)
1	Руководитель ведущий программист	01.10.07 – 13.12.07	2-3-4-7-10-13-14	40	100%
2	1-й программист	15.10.07 – 19.11.07	5-8-11	25	100%
3	2-й программист	23.10.07 – 15.11.07	6-9-12	25	100%

Примечание. Расчет тарифной ставки программистов был выполнен по формуле: $S_{\text{тариф}} = S_{\text{оклад}} / T_{\text{время}}$, где $S_{\text{тариф}}$ – почасовая тарифная ставка; $S_{\text{оклад}}$ – месячный оклад (4500 грн); $T_{\text{время}}$ – месячный фонд рабочего времени в нормочасах. Ведущему программисту добавлено за руководство проектом.

Назначение исполнителей и определение трудозатрат

На первом этапе необходимо заполнить **Лист ресурсов** по указанным колонкам, для этого использовать данные таблицы 2.7. Дополнительные данные для заполнения таблицы следующие: тип – трудовой; группа – люди; начисление – пропорциональное.

Затем надо определить время участия исполнителей в проекте.

На втором этапе не выходя из режима **Лист ресурсов** необходимо определить время участия в проекте первого и второго программистов, для чего: **Проект/ Сведения о ресурсе** в таблице **Доступность ресурса** для каждого программиста отдельно установить **Доступен с** и **Доступен по**. Данные взять из таблицы 2.7. Можно войти в форму **Сведения о ресурсе** не выходя из **Листа ресурсов** двойным щелчком по строке нужного исполнителя.

На третьем этапе следует выполнить назначение специалистов на конкретные задачи. Перечень задач по исполнителям

указаны в таблице 2.7. Для выполнения операции выполните следующие действия:

1. Откройте диаграмму Ганта
2. Установите курсор на графу «**Название ресурсов**» на нужную по номеру задачу, например – **Разработка функциональной схемы программного комплекса** – и щелчком из списка выберете исполнителя (Ведущий программист). Подобным образом выполните назначение исполнителей по всем задачам. Если необходимо ввести несколько ресурсов, то следует двойным щелчком вызвать форму **Сведения о ресурсе** и там построчно ввести все ресурсы на данную работу.

Анализ использования рабочего времени исполнителей, их трудозатрат и оплаты труда.

Следует прежде всего установить, нет ли перегрузки ресурсов, для этого необходимо проанализировать **Лист ресурсов, График ресурсов и Использование ресурсов** на отсутствие перегрузки исполнителей в таблицах и на графике (перегрузка отмечается красным цветом). В нашем примере роли были распределены точно и перегрузка ресурсов отсутствует. Если бы в проекте возникла перегрузка ресурсов, то необходимо применить методы выравнивания ресурсов. Для руководителя проекта важна информация по исполнителям, по исполнителям и времени, ресурсы с превышением доступности и другие, их можно получить в виде отчетов в **Вид/Отчеты/Назначения**. Данные по исполнителям и времени представлены в таблице 2.8.

Исполнитель	№ задачи	Единицы в %	Трудозатраты, час	Начало	Окончание
Ведущий программист			431,2		
	2	100	70,4	Пн01.10.07	Чт11.10.07
	3	100	14,4	Чт11.10.07	Пн15.10.07
	4	100	50,4	Пн15.10.07	Вт23.10.07
	7	100	112	Вт23.10.07	Пн12.11.07

	10	100	48	Пн12.11.07	Вт20.11.07
	13	100	80	Вт20.11.07	Вт04.12.07
	14	100	56	Вт04.12.07	Чт13.12.07
Программист 1			198,4		
	5	100	35,2	Пн15.10.07	Пт19.10.07
	8	100	115,2	Пн22.10.07	Пт09.11.07
	11	100	48	Пт09.11.07	Пт19.11.07
Программист 2			126,6		
	6	100	60,8	Вт23.10.07	Пт02.11.07
	9	100	36,8	Пт02.11.07	Пт09.11.07
	12	100	32	Пт09.11.07	Чт15.11.07

Для анализа степени перегруженности ресурсов можно воспользоваться возможностями панели инструментов «Анализ» «Анализ повременных данных в Excel» и получить график использования ресурсов (рис.2.5).

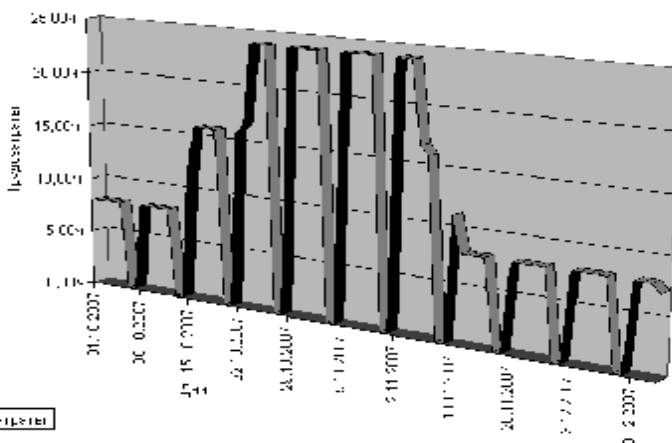


Рис. 2.5. График использования трудовых ресурсов

Расчет стоимости проекта

Есть несколько методик планирования стоимости проекта: по аналогии, по параметрам и снизу вверх.

Определение стоимости проекта *по аналогии* (analogous estimating) можно применять, когда планируемый проект аналогичен ряду других, выполняющихся в организации ранее. В таком случае общая стоимость проекта определяется исходя из накопленного опыта, а затем общая стоимость распределяется между задачами. Эта методика наименее точна, но ее применение занимает меньше всего времени.

Оценка стоимости *по параметрам* (parametric modeling) довольно популярная методика. Типичным примером является оценка стоимости строящегося дома по площади. В отношении программного обеспечения можно оценить примерную стоимость проекта по количеству команд программы. Точность этой методики и, соответственно, трудозатраты на ее использование зависят от числа оцениваемых параметров.

Методика оценки стоимости проекта *снизу вверх* (bottom-up estimating) заключается в расчете стоимости отдельных задач проекта и формировании общей стоимости отдельных задач проекта и формировании общей стоимости проекта, исходя из суммарной стоимости всех работ. Эта методика является наиболее точной и именно на ее использование ориентирована программа MS Project. В своих решениях мы будем ориентироваться на эту методику, хотя вышеназванные методики можно применять в комплексе. Например, если известно, что затраты на тестирование программы составляют 25% от затрат на проект разработки программного обеспечения, то можно оценить стоимость всех работ по проекту снизу вверх, и, исходя из этого, определить общую стоимость фазы тестирования и уже спланировать затраты на задачи этой фазы, такой метод определения затрат применяется и в ГОСТ 4.071.030 для некоторых фаз проекта ПО.

Структура затрат на выполнение проекта и методы их расчета

В MS Project общая стоимость проекта складывается из фиксированной стоимости ресурсов и задач и стоимости назначений, которая, в свою очередь, определяется ставками ресурса, трудозатратами и стоимостью использования ресурса. Как мы уже знаем, для каждого ресурса проекта можно определить почасовую ставку или стоимость за использование. Стоимость назначения определяется стоимостью ресурса, умноженной на длительность назначения (при почасовой ставке), либо фиксированной стоимостью ресурса. При назначении ресурса на задачу программы определяют его стоимость и стоимость задачи, складывая стоимость всех ее назначений и добавляя к ним фиксированную стоимость задачи, если она указана. Суммарная стоимость задач определяет стоимость проекта в целом.

Лабораторная работа № 4. Расчет стоимости проекта.

Затраты на выполнение нашего проекта будут состоять из затрат на основную заработную плату исполнителям, дополнительную заработную плату, отчислений в социальные фонды в % от зарплаты, затрат на закупку или аренду оборудования, на организацию рабочих мест и на накладные расходы.

Расчет заработной платы и отчислений в социальные фонды

Как мы уже установили, для работ в рамках проекта привлекаются три разработчика со знанием языка Delphi. Средний месячный оклад персонала подобной квалификации — 4500 грн.. По данным таблицы «Затраты» MS Project сумма основной заработной платы по выполненным назначениям составляет 25448 грн. Расходы на дополнительную заработную плату учитывают все выплаты непосредственно исполнителям за время, не проработанное на производстве, но

предусмотренное законодательством, в том числе: оплата очередных отпусков, компенсация за недоиспользованный отпуск и др. Величина этих выплат составляет примерно 20% размера основной заработной платы. Отчисления с заработной платы состоят из уплаты единого социального налога. Согласно налоговому законодательству Украины из фонда оплаты труда(основной и дополнительной оплаты) производятся выплаты в Пенсионный фонд, Фонд по безработице, Фонд охраны труда, Фонд социального страхования, в общем эти выплаты составляют 38% от фонда оплаты труда.

Таблица 3.8.Месячная заработная плата исполнителей и отчисления

№	Должность	Заработная плата и отчисления		
		основная по назначениям	дополнительная, 20%	отчисления в социальные фонды, 38%
1	Вед. программист	17248	3450	7865
2	1-й программист	4960	992	2261
3	2-й программист	3240	648	1477
Итого		25448	5090	11603

Таким образом, в стоимость проекта должна войти зарплата с отчислениями в сумме 42141 грн. Основная зарплата вошла в проект в процессе назначения исполнителей на выполнение работ. Осталось ввести дополнительную оплату и отчисления.

Пример ввода дополнительной оплаты:

Если дополнительная оплата зависит от тарифной ставки, то для ее ввода можно использовать инструмент изменения стандартной ставки, который предусмотрен в **форме Сведения о ресурсе / Затраты**. Здесь можно для каждого исполнителя выполнить изменение предыдущей ставки, указав в следующей строке процент изменения со знаком «+» (в сторону

повышения) или со знаком «–» (в сторону понижения). Сделать это можно следующим образом, покажем на примере ввода дополнительной оплаты ведущему программисту:

1. Открыть форму **Сведения о ресурсе** \ закладка. **Затраты**
2. В свободной строке таблицы А в графе **Стандартная ставка** ввести со знаком «+» процент дополнительной оплаты. В нашем случае надо ввести +20%.
3. Обязательно надо указать дату, с которой будет действовать изменение для исполнителя – для ведущего программиста – с начала проекта (01.10.2007). Начисление затрат оставить – **Пропорциональное**
4. Закрывать форму

Произойдет изменение ставок и автоматический пересчет фонда заработной платы. В результате мы получим следующие затраты фонда заработной платы: по ведущему программисту – 20697,6 грн, по 1 Программисту – 5952 грн, по 2 Программисту -- – 3888 грн. Эти данные можно просмотреть и проанализировать в таблице **Затраты** и в отчетах группы **Затраты**.

Пример ввода отчислений в социальные фонды.

Отчисления в социальные фонды выполняются от всей оплаты труда (основной и дополнительной). Обычно они отчисляются в фонды после выплаты зарплаты, поэтому их можно внести как фиксированные затраты в **Лист ресурсов** по группе **Материальные**, создав подгруппу **Деньги** в графу **Затраты на использование, По окончании**. Затем в режиме **Диаграмма Ганта** открыть таблицу **Затраты** и внести эту сумму (11603 грн) по строке: **Завершение проекта**, графа: **Фиксированные затраты**, в графе **Начисление фиксированных затрат** выбрать: **По окончании**. Проконтролируйте включение этой суммы в стоимость проекта используя таблицы и отчеты.

Расчет материальных ресурсов на выполнение проекта.

Затраты на расходные материалы

Затраты на расходные материалы определяются перечнем минимально необходимых для организации работ расходных материалов. Примерный перечень приведен в табл 2.9.

Таблица 2.9.

№ п\п	Наименование	Цена	Количество	Сумма, грн
1	CD-RW, шт	10	10	100
2	Бумага офисная , пачка	5	25	125
	Итого			225

Расходные материалы полностью расходуются в процессе выполнения проекта, поэтому их стоимость полностью входит в его стоимость. Впишем данные в **Лист ресурсов** по типу **Материальные** в группу **Материалы** и общую стоимость в графу **Затраты на использование**. Расходными материалами распоряжается ведущий программист, он должен их получить в начале проекта, поэтому выполним назначение расходных материалов на первую задачу, которую выполнит ведущий программист (**Разработка функциональной схемы ПК**). Технология ввода данных такая же как при назначениях исполнителей на конкретные работы.

Затраты на обеспечение работ оборудованием

Затраты, связанные с обеспечением работ оборудованием определяются составом оборудования и необходимостью его закупки или аренды. В данной разработке целесообразно иметь до проекта следующее необходимое оборудование (табл.2.10) и в стоимость проекта включить затраты на амортизацию оборудования.

Таблица 2.10.

№ п/п	Оборудование	Цена, грн	Количество	Сумма, грн
1	ПЭВМ, шт	4000	3	12000
2	Принтер лазерный, шт	800	1	800
4	Сетевое оборудование, комплект	200	2	400
5	Стол компьютерный, шт	500	3	1500
6	Стулья	300	4	1200
	Итого			15900

Затраты, связанные с амортизацией, определяются по следующей формуле:

$$S_{\text{аморт}} = \sum_i ((C_{\text{оби}} * D_{\text{ii}}) / D_{\text{раби}}), \text{ где}$$

$C_{\text{оби}}$ – затраты на приобретение i -го средства производства;
 D_{ii} – время использования i -го средства в днях;

$D_{\text{раби}}$ – полный ожидаемый срок эксплуатации i -го средства в днях.

Принимая за полный срок эксплуатации ПЭВМ 3 года (с учетом морального старения), а для прочего оборудования – 10 лет и учитывая, что в году приблизительно 255 рабочих дней, а рабочее время проекта составляет 95 дня, определим:

$$S_{\text{аморт}} = ((4000 * 3) \text{ грн} * 95 \text{ дня}) / (255 \text{ дня/год} * 3 \text{ года}) + ((800 + 200 * 2 + 500 * 3 + 300 * 4) \text{ грн} * 95 \text{ дня}) / (255 \text{ дня/год} * 3 \text{ года}) = 1490 + 484 = 1974 \text{ грн}$$

Эту сумму мы введем как фиксированную сумму в Лист ресурсов и в таблицу Затраты в Начало работ.

Затраты на накладные расходы

Накладные расходы, связанные с выполнением проекта, следует вычислять, ориентируясь на расходы по основной заработной плате. Обычно они составляют от 60 до 100% расходов на основную заработную плату. Таким образом:

$$S_{\text{накл}} = 25448 * 0,6 = 15269 \text{ грн}$$

Определение общей стоимости проекта.

По данным отчета о бюджете общая стоимость проекта составила 59609 грн. В таблице 2.11 проведем анализ структуры затрат.

Таблица 2.11.

№ п\п	Статьи затрат	Сумма затрат, грн	Удельный вес, %
1	Зарплата основная	25448	42,8
2	Зарплата дополнительная	5090	8,5
3	Отчисления в соцфонды	11603	19,5
4	Расходные материалы	225	0,3
5	Затраты на оборудование	1974	3,3
6	Накладные расходы	15269	25,6
		59609	100,0

Таким образом, общие затраты на выполнение проекта равны 59609 грн. Данные о бюджете можно получить: Вид\Таблица\затраты или Вид\Отчеты\затраты или используя Панель представлений.

В ряде случаев продажа ПО предполагает его настройку под условия эксплуатации, выдачу рекомендаций для конкретного использования ПО и др. Вся совокупность затрат на эти мероприятия определяется как затраты на внедрение ПО. В рассматриваемом примере эти затраты не рассматриваются.

Контрольные вопросы

1. Какие тенденции изменили подходы к проектированию информационных систем (ИС)?
2. Понятие жизненного цикла ИС. Какой компонент является основной и ключевой составляющей жизненного цикла ИС?
3. Опишите кратко этапы классического жизненного цикла ИС. Недостатки и достоинства классического жизненного цикла ИС

4. Что такое прототипирование, необходимость его использования при проектировании ИС. Что понимается под прототипом?
5. Какие формы может принимать прототип? Этапы прототипирования. Достоинства и недостатки прототипирования
6. Какие стратегии используются при разработке ПО?
7. Что такое спиральная модель жизненного цикла ИС, ее достоинства и недостатки?
8. Что понимается под тяжеловесными и облегченными процессами разработки ПО?
9. Что понимается под технологией программирования, какие требования к ней предъявляются?
10. Какие специалисты входят в проектную группу разработки ПО и их функции?
11. Охарактеризуйте кратко сущность CASE-технологий. Что Вы знаете об унифицированном процессе и языке моделирования, его обеспечивающем?
12. Дайте краткую характеристику основным стандартам проектирования ИС.
13. Что такое проект, перечислите основные этапы проекта.
14. Из каких основных элементов состоит план проекта, опишите основные свойства элементов проекта.
15. Что такое «треугольник проекта», как влияют изменения одного из сторон треугольника на другие его стороны?
16. Дайте краткую характеристику методу управления проектом – СРМ
17. Каково назначение пакета Microsoft Project, опишите основные возможности Microsoft Project.
18. Какова последовательность проектирования работ в Microsoft Project?

ЧАСТЬ 3. ПРИМЕНЕНИЕ КИБЕРНЕТИЧЕСКИХ АНАЛОГИЙ В ФОРМИРОВАНИИ ПЕДАГОГИЧЕСКИХ СИСТЕМ

РАЗДЕЛ 1. ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ОБУЧЕНИЯ

1.1.Предпосылки применения кибернетики в построении обучающих систем

Практически во всех развитых странах с 80-х гг с нарастающей активностью формируется мысль, что существующая модель образования исчерпала себя, мир приблизился к осознанию необходимости радикального пересмотра оснований образовательной практики и педагогической деятельности.

В решении названных проблем свою важнейшую роль должны сыграть новейшие информационные технологии, основанные на кибернетических и синергетических аналогиях. Как мы уже установили, кибернетика это наука об управлении сложными динамическими системами, способными воспринимать, хранить и перерабатывать информацию и использовать её для управления. Процессы и системы, изучаемые кибернетикой, имеют ряд черт, присущих также деятельности и поведению человека. В частности, обучение и воспитание могут рассматриваться как регулирование развития личности обучаемого путём целенаправленного управления его деятельностью и поведением.

Согласно теории обратной связи в кибернетике различают два типа управления. При цикличном управлении управляющая система получает сведения о ходе процесса и может оказывать воздействие на его течение, т.е. осуществляется обратная связь и коррекция. При разомкнутом управлении обратная связь и коррекция отсутствуют.

Эффективность процесса обучения, имеющего вероятностный характер, обеспечивается лишь цикличным управлением. Оно предполагает разработку двух видов

программ - основных и корректирующих. Первая разрабатывается до начала функционирования системы обучения; вторая - как до начала процесса обучения, так и по ходу его.

На стыке научного взаимодействия педагогики и кибернетики возникло новое педагогическое течение, получившее название кибернетическая педагогика. В своё время идеи кибернетики существенно повлияли на господствующую в обществе парадигму образования, внося в неё точный смысл управления сложной динамической системой, а также требования внедрения количественных измерительных процедур и математических моделей.

Необходимым условием эффективного управления процессом обучения является чёткое и точное определение целей педагогических воздействий. Внедрение кибернетических аналогий в педагогику сопровождается углубленным использованием измерительных процедур и математических моделей. Одним из достижений кибернетики являются информационные технологии, наиболее популярными из них стали обучающие программы.

1.2. Этапы становления кибернетической педагогики

Термин "кибернетическая педагогика" введен в 1966 году, академиком Бергом А.И. Выделяются три этапа ее развития.

1.Кибернетическая педагогика (наивная) 1950-1990 годы
Предмет исследования: процессы управления обучением
Направление исследования: программированное обучение

2.Кибернетическая педагогика (современная) 1990-2000. годы
Предмет исследования: управление учебными и образовательными процессами
Направление исследования: интеллектуальные системы управления учебными и образовательными процессами
Цель исследования: повышение эффективности учебных и образовательных процессов за счет использования в вузах и управлениях образованием и наукой искусственного интеллекта

3.Кибернетическая педагогика (будущего) 21 век. Предмет исследования: управление познавательными и учебными процессами с ускоренной динамикой. Адаптивное управление образовательными процессами. Направление исследования: интеллектуальные системы управления учебными и образовательными процессами на основе искусственного коллективного разума Цель исследования: повышение эффективности взаимодействия образовательной и экономической систем государства

Наивная кибернетическая педагогика

Программированные пособия, написанные для изучения различного учебного материала, имели ряд недостатков, которые сократили их практическое применение и широкого использования не нашли. К их недостаткам можно отнести слабую дидактическую проработку предлагаемого учебного материала, а также тривиальные контрольные задания, которые предполагали выбор ответов из 3-5-ти заданных вариантов

Обучающие машины были предназначены в основном для тренировок элементарных навыков обучающихся. Интерес к программированному обучению постепенно (к 80-м годам прошлого столетия) начал уменьшаться из-за отсутствия ощутимых успехов в обучении при его применении

Результат: приобретен первоначальный опыт использования технических и программных средств в области образования, а также выявлены возможные пути совершенствования учебного процесса

Развитие современной кибернетической педагогики (направления)

- 1.Исследования в области дистанционного образования и обучения
2. Исследования по созданию интеллектуальных обучающих систем по конкретной учебной дисциплине
- 3.Исследования по созданию образовательных систем на основе гибридного интеллекта

4. Исследования по созданию электронных учебников
5. Исследования по повышению эффективности обучения за счет использования ИТ при традиционном обучении
6. Исследования по автоматизации управления образованием и наукой

Итоги этого этапа

Результаты исследований показывают, что основные усилия ученых посвящены исследованию влияния информационных технологий на познавательные процессы обучающихся. Исследование познавательной деятельности научно-педагогических работников остается еще слабо изученным.

Крайне мало работ посвященных разработке и исследованию экспериментальных методов на современном этапе развития кибернетической педагогики. Например, в полной мере не разработана методология планирования и проведения педагогических экспериментов

Задача создания высокоэффективной системы управления учебными процессами и вузами в целом на основе информационных технологий остается по-прежнему актуальной.

Таким образом, необходима систематизация результатов исследований, связанных с внедрением информационных технологий в образовательную сферу, а также разработка такого формального аппарата, который позволил бы на основе единых принципов моделировать педагогические и образовательные процессы и явления. Это задача будущего развития кибернетической педагогики

1.3. Компьютерная кибернетическая технология обучения

Структура процесса обучения.

Обучение есть информационный процесс формирования знаний у субъекта обучения под управлением преподавателя.

Субъект обучения - индивидуум, группа индивидуумов, система искусственного интеллекта.

Знание есть активная информация, способная генерировать новую информацию.

Уровни знания

В процессе обучения выделяем *уровни (формы) знания*:

- владение информацией (содержанием предмета);
- понимание;
- умение решать типовые задачи предмета;
- умение решать прикладные задачи предмета;
- умение синтезировать междисциплинарные связи;
- умение практической работы по специальности;
- умение исследовательской работы;

Семантические модели, формируемые в сознании обучаемого на каждом этапе обучения

- структурная модель предмета;
- понятийно-сущностная модель предмета;
- алгоритмическая (вычислительная) модель предмета;
- проблемная модель предмета;
- модель знаний специалиста;
- опыт в области специальности;
- новые знания - как результат исследовательской работы.

Результат обучения – компетентность.

Для достижения вышеуказанных уровней знаний должна быть разработана концептуальная модель управления образовательным процессом, которая должна включать, на наш взгляд, следующие компоненты: цели обучения, степень восприятия, содержание дисциплины, способы, методы и формы обучения и контроля, деятельность преподавателя и студента в процессе обучения и диагностика обучения.

Цели обучения как общеобразовательные, так и по каждой дисциплине, они рассматриваются как вход в систему обучения.

Для определения степени подготовленности студентов по каждой учебной дисциплине выделяют объем знаний, которые необходимы для усвоения в соответствии с учебной программой, которая составляет базовый объем знаний. Базовые

знания представляют минимум государственного образовательного стандарта. Но и среди базовых знаний выделяют те, которые должны оставаться в памяти из любой дисциплины, и в совокупности образовывать мировоззренческие знания. Для правильной постановки целей обучения преподавателю важно полнее анализировать условия, в которых будет находиться специалист после учебы и к которым он должен быть обстоятельно подготовленный.

На всех уровнях усвоения знаний можно выделить четыре вида умений:

- умение распознавать объекты, понятия, факты, законы, модели.
- умение действовать по образцу, по типовому алгоритму, по правилу.
- умение проводить анализ ситуации, выделять главное и строить из освоенных операций процедуры, которые позволяют получить решение тестового задания.
- умение и способность находить оригинальные решения.

Содержание дисциплины. При использовании компьютерных технологий обучения появляется возможность руководить процессом усвоения знаний на основе четкой систематизации и структуризации курса. Этот подход позволяет заложить в каждую составную часть учебной программы взвешивающий коэффициент и на этом построить системный подход к оценке знаний. Программно целевой подход к содержанию обучения, а затем и систематизация и структуризация предмета, по мнению специалистов, способствует следующему:

- формированию у студентов системных знаний;
- повышению объективности самооценки и оценки знаний;
- возможности более объективного и глубокого анализа степени усвоения отдельных фрагментов учебной программы.

В соответствии с выделенными учебными элементами, предварительно установленными дидактичными целями можно

автоматизировать процесс объективной и беспрестанной оценки знаний.

Оценка результатов обучения играет определенную роль в корректировке и направлении результата обучения соответственно поставленным целям. В этом случае оценка знаний становится эффективным инструментом повышения познавательной активности студента. Появляется возможность самоконтроля знаний, а также необходимость разработки экспертно учебной системы, а затем и рейтинговой системы контроля знаний. Создание экспертно учебных и рейтинговых систем контроля знаний в какой-то степени помогает выбрать направление в решении еще одной из важнейших проблем – в выработке единственного подхода к оценке профессионализма выпускника учебного заведения. В настоящее время единственной объективной оценкой качества подготовки специалиста является их оценка предприятиями и организациями. Этот метод неприемлем для использования в процессе подготовки специалиста, поэтому разрабатываются целые учебно-методические комплексы управления качеством подготовки, которые включают в себе следующие задачи:

- формирование эталонов качества подготовки специалистов;
- разработка средств контроля на базе эталонов качества;
- разработка, проведение процедуры сравнения достигнутого уровня подготовки с эталоном качества;
- разработка системы управляющих влияний условий и факторов, которые определяют достигнутое качество, с целью минимизации обнаруженных отклонений.

Основные компоненты концептуальной модели интерактивной системы обучения представлены на рисунке.

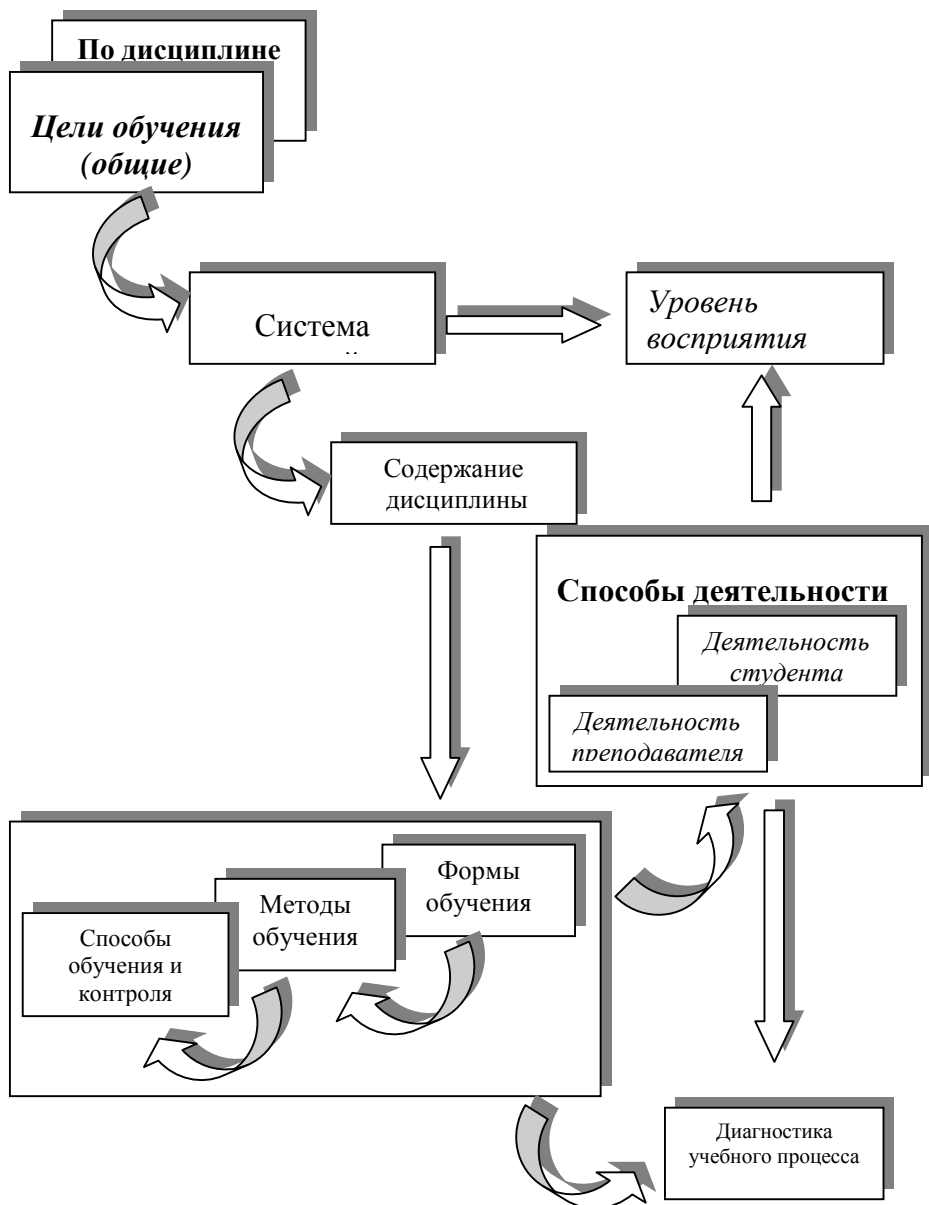


Рис.1.1. Концептуальная модель обучения

Обзор основных моделей, используемых при создании системы автоматизированного обучения.

Система автоматизированного обучения (САО) – это взаимодействующие в процессе решения педагогической задачи преподаватель, обучаемый и компьютер. При этом, компьютер выступает и как универсальное средство обучения, и как субъект педагогического процесса.

Модели и модельные задачи

Шесть моделей считаются основными в кибернетике, в явном или неявном виде эти модели используются при постановке педагогических задач:

- модель массового обслуживания и надежности;
- игровая модель;
- модель распознавания образов;
- графовая модель;
- алгебраическая модель.
- лингвистическая (семантическая) модель

Из всех перечисленных моделей более подробно остановимся на семантической модели и семантическом программировании.

Семантической моделью предметной области называем ее смысловое содержание, направленное на выражение того или иного уровня знания и представленное вне сознания некоторым формальным аппаратом.

Термином "**семантическое программирование**" обозначаем процедуры создания и применения (здесь - для целей обучения) семантических моделей предметной области. Формальный аппарат семантического программирования – это создание фреймовой структуры изучаемого предмета, которая может включать три типа элементарных фреймов: классификационные, смысловые связи и директивные.

Классификационные фреймы позволяют упорядочивать ключевые понятия предметной области под понятием - категорией, которое выступает как имя фрейма.

Директивный фрейм есть приказ для пользователя на проведение какого-либо действия или терминальный текст.

Смысловая связка содержит сказуемое, входной и выходной аргументы, являясь заготовкой предложения естественного языка.

Элементарные фреймы объединяются во фразы, совокупность которых представляет собой семантическую модель на ограниченном естественном языке. Эта модель может быть представлена графом.

Семантические сети были разработаны Ричардом Риченсом в 1956 году в рамках проекта Кембриджского центра изучения языка по машинному переводу. Процесс машинного перевода подразделяется на 2 части: перевод исходного текста в промежуточную форму представления, а затем эта промежуточная форма транслируется на нужный язык. Такой промежуточной формой именно и были семантические сети. Позже, в начале 1960-х, Робертом Симмонсом были начаты более серьезные исследования. В Интернете Семантическая сеть известна под названием Семантическая паутина.

Семантическую сеть можно представить с помощью ориентированного графа $G(X, U)$, множество вершин которого X отвечает множеству исследуемых понятий M , а множество ребер U определяют отношения включения понятий.

Основные элементы организации системы автоматизированного обучения

Электронный учебник

Электронные интерактивные учебно-методические материалы (пособия, учебники и т.д.) - это набор взаимозависимых HTML-документов, объединенных в логическую структуру: элементы текста, которые включают в себя статические и динамические изображения, аудио и

видеоматериалы, элементы меню и навигации, а также средства тестирования и самоконтроля. Каждое электронное пособие представляет собой автоматизированную обучающую среду со всем набором дидактического материала.

Формализация процесса обучения, позволила обоснованно сформировать системную структуру электронного учебника. Она должна быть модульной: каждый модуль поддерживает соответствующий этап процесса обучения, имеет необходимый набор алгоритмов, реализующих операции обучения и семантическое наполнение.

Предметное наполнение модулей составляют семантические модели, описывающие цели соответствующих этапов обучения, процедуры и средства их достижения, а также тексты, являющиеся терминалами семантических моделей.

Семантические модели совместно с алгоритмами используются для управления процессом обучения.

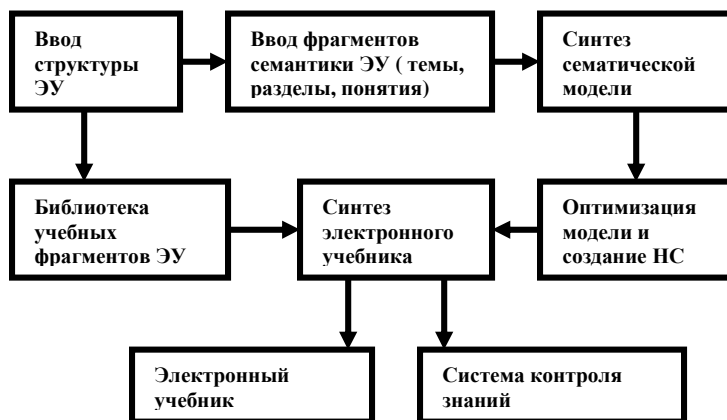


Рис.1.1 Этапы проектирования электронного учебника (ЭУ)

Лингвистический робот

Лингвистический робот, с одной стороны, является управляющей частью электронного учебника (его навигатором), с другой - универсальной оболочкой для построения электронных учебников. Как средство управления

лингвистический робот функционирует на основе принципа обратной связи, как универсальная оболочка - способен воспринять смысловые модели конкретных предметов и содержит типовые алгоритмы обучения, реализующие соответствующие методики обучения.

Настройка лингвистического робота как оболочки на конкретный предмет заключается в построении проекта электронного учебника в виде модели требуемых знаний на языке внешнего представления семантических моделей и вводе его в робот.

Дистанционное обучение

Под дистанционным обучением понимается процесс формирования знаний у субъекта обучения с использованием компьютерных технологий и средств телекоммуникации, которая обеспечивает интерактивный удаленный диалог обучаемого, находящегося в терминальном пункте обучения (возможно по месту жительства) с центром обучения - ВУЗом в соответствии с его индивидуальным графиком обучения

Процесс дистанционного обучения представляет собой информационный процесс, построенный по принципу обратной связи, включающий следующие этапы:

1. Планирование процесса обучения на семестр, осуществляемый деканатом.
2. Подготовки учебно-методического комплекса (УМК) по дисциплинам кафедры.
3. Компьютерное обучение дисциплинам, когда обучаемый самостоятельно работает над комплексом дисциплин, используя учебно-методические материалы на бумажном носителе и компьютерные курсы (КК) по дисциплинам, выполненные на базе лингвистического робота, осуществляющего управление процессом компьютерного изучения дисциплины в соответствии с индивидуальными характеристиками обучаемого его текущим уровнем знаний.
4. Работы с преподавателем, когда обучаемый работает под руководством преподавателя (при личном контакте),

который осуществляет консультации, дополнительное обучение, оценку знаний обучаемого.

Процесс обучения как управляемый процесс и подходы к проектированию САО

Управление процессом обучения происходит по принципу обратной связи: учебные задания, разъяснения вырабатываются преподавателем по определенным алгоритмам обучения, исходя из цели обучения, на основе сравнения модели требуемых знаний и модели текущих знаний обучаемого.

Процесс обучения, как управляемый по принципу обратной связи процесс, состоящий из операций, носит *дуальный характер*: в процессе обучения формируется не только его выход - знания обучаемого, но и реализация самого процесса обучения.

Алгоритмы управления обучением интерпретируются как алгоритмы выработки учебных заданий, разъяснений. Они должны определяться объективными законами познания и используемыми технологиями, методологиями, методами, методиками, приемами обучения. Свойство формализуемости программы обучения дает возможность подходить к учебному процессу как объекту проектирования и применить все те правила и закономерности, которые характерны для проекта.

Проблема проектирования учебного процесса стала особенно актуальной в период перехода к системе автоматизированного обучения (САО), основой которого является индивидуальный подход к работе с каждым студентом. В литературе дискутируются отдельные детали проектирования на разных этапах учебного процесса, но еще не создан системный, целостный подход к формированию модели обучения.

Первые попытки в переходе к новой технологии учебного процесса потребовали пересмотра ряда методологических основ обучения. Однако создание новой методологии педагогики отстает от практики ее технического

первооружения, а это приводит к тому, что автоматизация учебного процесса налагается на тот перечень методов и форм учебы, который использовался в традиционной педагогике, хотя применение ЭВМ позволяет значительно расширить ее возможности, появляется реальная перспектива перехода к интерактивным методам управления учебным процессам. Важную, если не основную роль в решении этого вопроса должны сыграть математические методы и модели. Модельное проектирование предусматривает наличие общей информационной модели объекта проектирования, аппарата ее совершенствования и возможности ее параметризации.

Таким образом, назрела необходимость использовать математическое моделирование на всех стадиях проектирования системы автоматизированного обучения (САО), а для этого весь учебный процесс должен быть описан четким языком математических символов и формул.

Выделение модели и выбор метода решения педагогической задачи часто взаимосвязаны, поскольку метод решения должен быть адаптирован к модели. Более того, для многих моделей вполне подходят типовые, модельные задачи с хорошо разработанными методами их решения (такая ситуация характерна для кибернетики). Поэтому разумно поставить и решить вопрос о возможности использования кибернетических моделей и методов решения модельных задач в САО. В явном или неявном виде некоторые кибернетические модели используются при постановке педагогических задач. Например, широко используется графовая модель при автоматизированном составлении расписания учебных занятий или распределении времени на изучение отдельных дисциплин в рамках профессиональной подготовки будущего специалиста. На базе игровой модели разрабатываются деловые и дидактические игры. Разные исследователи интуитивно чувствуют важность модели распознавания образов при решении задач педагогического контроля.

При исследовании была поставлена цель сделать попытку освещения *технологии обучения* как комплексной интерактивной системы и провести оценку возможностей

использования кибернетических моделей для проектирования этой системы.

Технология обучения – это системная категория, ориентированная на дидактичное применение научного знания, научные подходы к анализу и организации учебного процесса, с учетом эмпирических инноваций преподавателей и направленности на достижение высоких результатов в профессиональной компетенции и развитии личности студентов. Структурными составляющими такой системы является:

- цель обучения;
- содержание обучения;
- средства педагогического взаимодействия, в том числе мотивация и средства;
- организация учебного процесса;
- студент;
- преподаватель;
- результат деятельности (в том числе уровень профессиональной подготовки).

Таким образом, технология обучения допускает управление дидактичным процессом, который включает у себя организацию деятельности того, кого учат, и контроль этой деятельности. Эти процессы беспрестанно взаимодействуют: результат контроля влияет на содержание управляющих действий, то есть изменяет последующую организацию деятельности в интересах достижения целей, определенных на основе образовательных стандартов.

Исследование разных подходов к технологии учебного процесса позволило сделать вывод, что наиболее производительным подходом является деятельностный подход, он наилучше удовлетворяет требованиям к определению целей учебы и построения модели. Деятельностный подход дает принципиально другое видение предметных целей учебы. Его главные понятия: **деятельность, умение, задача** идеально отвечают принципам построения кибернетической модели обучения. **Умение** — приобретение студентом способности осуществлять конкретную деятельность в будущей профессии через систему полученных **знаний**, оно может рассматриваться

как цель некоторого этапа учебного процесса (вход в систему), то есть, сначала определяются цели – умения, а затем отбираются знания, необходимые для умений. *Деятельность* — процесс достижения цели – получение необходимых знаний – в процессе учебы благодаря действиям преподавателя, студента и учебного материала (компоненты системы). *Задача* — имитация ситуации, в которой студент может доказать степень приобретения умения (обратная связь). Следовательно, чтобы сформировать модель учебы, нужно выполнить структуризацию выделенных элементов и установить соотношение между ними (рис.1.2).

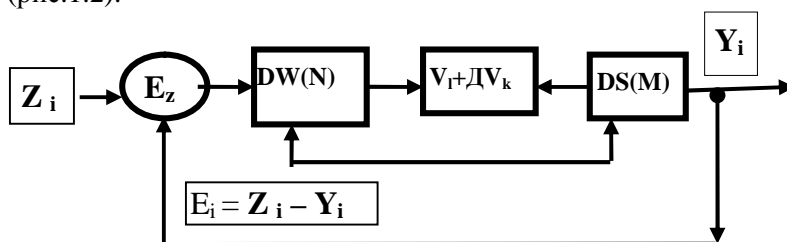


Рис.1.2. Структурная схема системы автоматизированной обучения (CAO).

Условные обозначения:

Z_i – отдельная цель обучения из множества $Z = \{Z_1, Z_2, \dots, Z_m\}$, что означает систему умений, которыми должны овладеть студенты при изучении конкретного учебного предмета;

E_z – критерии оценки степени достижения цели, это звено в модели реализует принцип обратной связи;

$DW(N)$ – подмножество(N) действий преподавателя, которое направлено на эффективное овладение студентом системой знаний и умений для поставленной цели ;

$V_1 + DV_k$ – система предварительно накопленных знаний (V_1) и знаний, которые будут приобретены при изучении данной дисциплины (DV_k);

$DS(M)$ – подмножество (M) действий студента, которое направлено на приобретение системы знаний и умений в рамках поставленной цели;

Y_i – отдельная задача из множественного числа $B = \{ Y_1, Y_2, \dots, Y_i, \dots, Y_s \}$, какое означает систему учебных заданий профессионального характера, которые должен уметь решать студент, степень умения контролируется звеном обратной связи на основе сравнения: $E_i = Z_{ii} - Y_i$; $E_i \in E_z$.

В кибернетике есть аналог представленной схемы САО, это модель автоматической системы регулирования (АСР). Принципы ее построения очень хорошо разработаны и используются в технических системах, в традиционных педагогических системах, где превалирует человеческий фактор, она может быть полезна как концептуальная модель, но при проектировании САО увеличивается роль технических систем (информационных технологий) и АСР может занять свое место в процессе формирования САН. Это направление уже приобретает развитие в педагогической науке под названием – кибернетическая педагогика, которая занимается математическим моделированием учебного процесса.

Сформулируем наши подходы к проектированию каждого элемента представленной модели.

Цель – начало проектирования образовательного процесса. Для того, чтобы спроектировать цели обучения, нужно описать и оценить то, что *делает* человек, потому что цель обучения — это всегда в конечном счете формирование умения что-то *делать*. Описывать и оценивать то, что человек делает, нужно с определенной степенью точности, строгости, систематичности. Цель задана корректно, если используемые понятия удовлетворяют следующим требованиям:

- определение и их признаки настолько точно описаны, что понятие всегда адекватно соотносится с его объективным проявлением (то есть с тем, что оно помечает);
- определено ее место в учебном плане подготовки специалиста, то есть она является составной частью системы целей;
- проявления и факторы, которые отражаются понятием,

имеют категорию меры, то есть их величина поддается прямому или косвенному измерению;

- результаты измерений могут быть соотнесены с определенной шкалой оценки.

Следовательно, задать цели обучения по учебной дисциплине – значит выделить и сформировать систему умений, которыми должны овладеть студенты.

Для системной постановки познавательной цели (**Z**) нужно, чтобы она была точно описана (**O**), поддавалась измерению (**W**) и, чтобы существовала шкала ее оценки (**Ez**). Отсюда выходит формула корректности цели: $Z = O + W + Ez$.

При формировании целевого блока для конкретной дисциплины нужно помнить, что она является частью предметного комплекса дисциплин, которые изучают студенты данной специальности, потому цели изучения конкретной дисциплины должны отвечать общим профессиональным целям – $Z_1 = \{Z_{11}, Z_{12} \dots Z_{1n}\}$.

Обучение – это информационный процесс формирования знаний у субъекта обучения под управлением преподавателя и с помощью современных средств. Информационный процесс состоит из операций. Операция характеризуется алгоритмом, входной и выходной информацией. Будем различать операции сложные и элементарные. В информационном процессе могут быть выделены этапы как совокупность операций, которые имеют общее целевое назначение.

Все эти операции нужно структурировать и оптимизировать при проектировании учебной программы CAO, где должны найти свое место действия преподавателя и студента (**DW**, **DS**), содержание учебника (**DVk**), диагностические измерения уровня знаний (**B**) и необходимые средства при формировании будущего специалиста. Действия преподавателя и студента и использование современных средств обучения можно представить как алгоритм управления учебным процессом, который состоит из элементарных педагогических действий, каждая из которых начинается с конкретной цели обучения и завершается финальным результатом, назовем это элементарное педагогическое действие оператором учебной программы CAO – **QJ** ($Q = Q_1, Q_2 \dots, Q_n$).

Проектирование учебной программы как алгоритма управления САО.

Учебную программу в настоящий момент рассматривают как документ, где в соответствии с требованиями к облику будущего специалиста определены содержание учебы и наиболее целесообразные средства организации его усвоения студентом. Но это определение учебной программы недостаточно раскрывает ее суть и ее педагогическую целесообразность и не показывает пути ее совершенствования. Несоввершенство традиционных учебных программ с точки зрения САО такие:

- цели обучения не отвечают тем критериям, которые задекларированы выше;
- программа регламентирует действия только одного участника педагогического процесса – преподавателя, который считается единоличным его руководителем, а студенту отводится роль только ведомого, что не допустимо при САО;
- она жестко детерминирована, у нее нет выбора возможных переходов в зависимости от уровня знаний студента, его способа мышления и тому подобное;
- блоки учебного материала очень укрупнены, их очень трудно связать с конкретными целями учебы, действиями преподавателя и студента и конечными результатами;
- она не должна существовать отдельно от учебного материала и действовать в пакетном режиме, а должна быть вмонтированной в семантическую сеть системы знаний и в интерактивном режиме регулировать процесс освоения студентом отдельных понятий учебного предмета;

По нашему мнению, в основе программы обучения должна быть модель, суть которой заключается в представлении поведения того, кого обучают, как конечного детерминированного автомата с графом возможных переходов.

База знаний определяет доступные блоки знаний, часть которых обозначена как финальные с выходом на конечные результаты. Граф доступных переходов должен сохраняться в базе знаний. В каждый момент обучения студенту доступен результат работы как функции отображения параметров состояния модели того, кого обучают. В результате выбора последующего пути обучения и ответов на поставленные вопросы меняется и состояние модели того, которого обучают. Учеба заканчивается по желанию студента в одной из финальных вершин. Следовательно, в основе построения учебной программы следует поставить систему знаний, которой должен овладеть студент при изучении конкретной дисциплины.

Под **знанием** понимаются понятие и системы понятий, которые отображают содержание учебного материала (учебника) по дисциплине. Все множество понятий разбивается на k подмножеств, каждое из которых отвечает уровню понятий. При структуризации учебного предмета будем такие блоки понятий помечать как учебные элементы дисциплины – V_i , где $(V_i \in \bar{I} \rightarrow DV_k)$. Для формального определения разбивки используется семантическая сеть. Семантическая сеть отображает семантику дисциплины в виде понятий и отношений. Формальный аппарат семантического программирования - фреймовые структуры. Семантическую сеть можно представить с помощью ориентированного графа $G(V, U)$ [], где V – узлы графа, U – его ребра. В нашем примере семантическая сеть будет выглядеть как графическая классификационная схема, в узлах которой находятся учебные элементы, – V_i , а дуги этой схемы отображают связки между ними, конечной вершиной каждой ветви схемы должен стать оператор учебной программы (Q_j), который будет играть роль регулятора учебного процесса на данном этапе (рис.1.3). Такую семантическую модель можно положить в основу навигационной системы обучения.

Семантическая модель является воплощением основных принципов кибернетической лингвистической модели. Понятие «**лингвистический**» означает использование методов, которые развиваются в математической лингвистике, исследующих естественные языки методами алгебры. С помощью методов

структурной лингвистики создается грамматика поведения системы с определенными синтаксическими правилами и семантика с использованием процедуры учебы на известных примерах поведения.

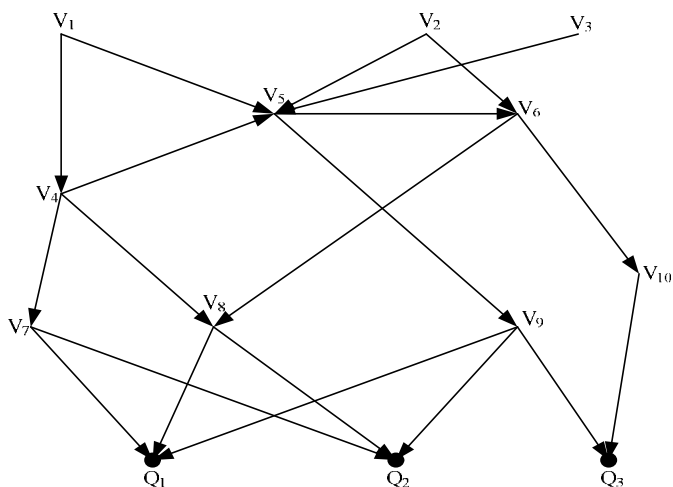


Рис.1.3. Граф-модель учебной программы CAO

Нуждается в более детальных объяснениях формирование системы операторов в алгоритме управления CAO. Каждый учебный элемент нуждается в усилиях участников учебного процесса: преподавателя и студента, то есть программа должна включать разработку алгоритма их действий (управление учебным процессом).

Алгоритм действий зависит от того, какую роль играет в подготовке специалиста эта дисциплина и ее отдельные элементы. Это в педагогической науке обуславливается системой уровней усвоения студентом учебного материала. Эту систему мы будем означать как множественное число: $S = \{S1, S2 \dots Sh\}$. Ей должны подчиняться действия и преподавателя, и студента, в педагогической практике, например, выделяют такие уровни усвоения учебного элемента:

- *уровень знакомства (S1)* : студент способен сделать опознание, различение и соотношение понятий

дисциплины;

- *уровень воссоздания (S2)* : студент может воспроизвести (повторить) информацию, операции, действия, решить типичные задачи, рассмотренные при учебе;
- *уровень умений и навыков(S3)* : студент умеет выполнить действия, общая методика и последовательность (алгоритм) которых изученные на занятиях, но содержание и условия их выполнения не новые.
- *уровень творчества (S4)* : студент умеет самостоятельно добывать новые знания и умения.

Для достижения любого уровня усвоения запланированного учебного элемента студент должен осуществить конкретную учебную деятельность (оператор – **Qj**), которая может состоять из трех видов действий: ориентированных (**Od**), исполнительских (**Wd**) и контрольных (**Kd**), выполняемых студентом обычно с помощью преподавателя: **Qj = Od + Wd + Kd**.

Ориентированные действия заключаются в получении студентами необходимой информации, осознания поставленной перед ними задачи усвоения. С помощью инструктажа и ориентирования, которые даются преподавателем, они выбирают пути, средства и методы ее решения.

Исполнительские действия отражаются в интеллектуальной переработке полученной информации и выполнении упражнений для усвоения знаний, умений и навыков. Студент выполняет программу, разработанную в ходе ориентированных действий.

Контрольные действия – действия, с помощью которых проверяются полнота, правильность и качество выполнения преподавателем и студентами ориентировочных и исполнительских действий, в случае необходимости корректируются ошибочно выполнены операции.

Задачу создания операторов управления учебным процессом облегчает наличие детерминированного соотношения между уровнем усвоения знаний и совокупностью

действий преподавателя и студента, которые уже сложились в определенной педагогической системе, которая дает возможность создать нормативную матрицу операторов учебного процесса (см. табл.). В таблице каждому уровню усвоения поставлены в соответствие совокупность действий преподавателя и студента на всех этапах учебной деятельности, эта совокупность и составляет суть j -го оператора (Q_j).

Уровни усвоения знаний	Ориентированные и исполнительные действия		Контрольные действия	
	преподавателя	студента	преподавателя	студента
1	DW (K)	DS (A)	DW (O)	DS (F)
2	DW(L)	DS (B)	DW (P)	DS (G)
3	DW(M)	DS (C)	DW (R)	DS (H)
4	DW(N)	DS (D)	DW (U)	DS (S)

Заключительным этапом проектирования учебной программы является назначение каждому финальному учебному элементу уровня усвоения, стыковку всех элементов модели учебы с целями учебы и ее оптимизация. Модель учебной программы в заключительном виде представляет собой интегрированный граф, где естественно объединены система знаний и алгоритм управления ее усвоением. Графическая форма модели позволяет успешно использовать для ее оптимизации методы теории графов, которые способны выделить количество тактов приобретения знаний и умений, определить окончательно конечные вершины, которые можно связать с операторами и целями учебы, упорядочить этапы учебного процесса и рассчитать некоторые его параметры.

РАЗДЕЛ 2. ИСПОЛЬЗОВАНИЕ КИБЕРНЕТИЧЕСКИХ МОДЕЛЕЙ В СИСТЕМЕ ОБУЧЕНИЯ ПРИ СОЗДАНИИ ЭЛЕКТРОННОГО УЧЕБНИКА.

2.1. Исследование дидактических особенностей учебного курса

Использование кибернетических моделей при построении ЭУ рассмотрим на примере учебного курса «Компьютеры в экономике и бизнесе», который входит в учебные планы специальности «Информатика» на 4 курсе, когда студенты уже проходят учебные практики и сталкиваются с программированием решения практических, чаще всего экономических задач. Будущий специалист начинает понимать, что понятие «информационная система» в современном представлении не сводится только к созданию программного обеспечения, а представляет собой сложный комплекс использования информационных технологий в управлении конкретным объектом (производство, бизнес, банки и т.п.) Следовательно, информатики как специалисты в области информационных технологий обязаны владеть знаниями и умениями их использования в создании и поддержки информационных систем, обслуживающих конкретные бизнес-процессы.

Важным направлением современного преподавания этой дисциплины следует считать расширение возможностей применения проектного метода обучения, при котором студент, выполняя индивидуальный проект учится создавать программную и информационную среду для конкретного бизнес-процесса. Следовательно, при проектировании структуры ЭУ по курсу «Компьютеры в экономике и бизнесе» следует уделить внимание практическому освоению приложений, которые автоматизируют процесс управления в области конкретной функции, например, бухгалтерский учет (ППП «1С:Предприятие»). Усвоение такой сложной программы без преподавателя потребует включения мультимедийных средств имитации действий при работе с «1С:Предприятие».

Проектируемый электронный учебник должен быть предназначен как для использования преподавателями при чтении лекций в мультимедийных аудиториях и студентами в рамках учебного процесса, так и для самостоятельной работы обучающихся, т.е. по своему прямому назначению. (в том числе, вне учебного заведения). Поэтому страницы учебника, созданные на основе коллекции информационных объектов, должны быть снабжены системой меню и указателей, а также поисковым механизмом. Он должен быть прост в использовании преподавателями и студентами и представлять собой систему, открытую для изменения и дополнения. Для этого преподавателю надо предусмотреть наличие простого и доступного инструмента (оболочки) для совершенствования и дополнения учебного материала, а студенту возможность копирования ресурсов ЭУ, чтобы впоследствии воспроизводить их при помощи стандартных программных средств.

2.2. Принципы разработки электронного учебника на основе семантической сети.

Качество подготовки студентов во многом определяется качеством учебных пособий, используемых ими. К недостаткам традиционных электронных учебников стоит отнести отсутствие средств контроля усвоения знаний в процессе работы с ними, а также ориентации на определенный уровень знаний студентов. В итоге студенту выдается для изучения строго определенный учебный материал в строго определенной последовательности.

Для повышения эффективности обучения необходимо создание учебников, которые настраиваются на уровень знаний студента и выдают ему материал в заданном объеме и последовательности.

В этом случае, ЭВМ уделяется не пассивная роль воссоздания текста, графики и другого иллюстративного материала, а активная роль, свойственная преподавателю.

Допустим, имеется дисциплина, для которой необходимо разработать электронное учебное пособие. Учебный материал

по этой дисциплине представлен некоторой моделью знаний. Определены цели обучения - исследуемые темы и понятия.

Задача заключается в создании подхода к разработке интеллектуального электронного учебного пособия (для заданной дисциплины), что позволяет выполнять следующие функции:

- определять уровень знаний студента, исходя из цели обучения;
- синтезировать план обучения;
- обучать, путем выдачи учебного материала, в соответствии с учебным планом .

Человеческие знания в целом и каждый отдельный индивидуум имеют сложную иерархическую структуру, подобную структуре вещественного мира.

Как структура материи от элементарных частиц восходит к сложным объектам окружающего нас мира, точно также и знание, основываясь на самых простых понятиях, восходит к сложным понятиям, ранее не известных закономерностей окружающего нас мира (открытий). Причем, как материя непостижима и не имеет границ, так и наши знания развиваются в сторону более сложных понятий и при этом происходит уточнение самых простых понятий.

Процесс обучения можно рассматривать, как процесс, который происходит в одном человеке, но при этом в главном ничем не отличается от приобретения знаний человечеством. Главное отличие - это ограниченность и определенность знаний при их изучении, в отличие от безграничности и непредсказуемости знаний о природе.

При изучении дисциплины происходит изучение более сложных понятий на основе уже известных понятий. В целом, весь процесс обучения строится по таким же принципам: от основных дисциплин к более сложным и специализированным.

Учебное пособие, которое создается на принципах системы искусственного интеллекта, должно содержать базу знаний (удерживающую знание по исследуемой дисциплине), механизм вывода и перечень возможных целей изучения (исследуемых тем и понятий).

Обучение начинается с задания студенту цели обучения - исследуемой темы. Учебное пособие должно выдать все неизвестные знания по этой теме. Для этого учебное пособие осуществляет тестирование приобретенных знаний, и потом по результатам тестирования формируется и выводится следующая порция неизвестного учебного материала.

Предусматривается, что учебный материал представлен определенным набором понятий. Для определения понятия необходимо знание других понятий, непосредственно с помощью которых, определяется исследуемое понятие. (Например: Чтобы выучить понятие **информационная система**, необходимо когда-то выучить следующие понятия – **система**, **сложная система**, **система управления**, **информация** и тому подобное).

Между понятиями существует отношение включения. Каждому понятию, отвечает некоторое подмножество понятий, с помощью которых оно определяется и с которыми находится в отношении включения. Отношение включения является частично упорядоченным. Оно определяет частично упорядоченное множество понятий. Это отношение обуславливает разбивка всех понятий на подмножества (уровни знаний).

Нижний уровень составляют базовые знания. Понятия этого уровня предполагаются известными обучаемому, и не требуют последующего определения. На основе этого уровня определяются понятия на уровень выше.

На основе понятий этих уровней определяется следующий уровень понятий и так далее. Все множество понятий разбивается на k подмножеств, каждое из которых отвечает уровневые понятия.

Для формального определения разбивки используется семантическая сеть. Семантика изучает содержание предложений, а сеть представляет собой набор вершин графа, соединенных дугами. В *семантической сети* роль вершин исполняют понятия, а *направленные* дуги задают отношения между ними. Таким образом, семантическая сеть представляет семантику предметной области в виде понятий и отношений.

Следует отметить, что разработка семантической сети является творческим процессом. Поэтому, аналогично тому, как после составления программы она требует отлаживания, так и вновь созданная семантическая сеть требует анализа и оптимизации.

Качество разработанной семантической сети можно оценить с помощью ряда показателей, используемых для оценки качества, это:

- достоверность;
- кумулятивность;
- противоречивость.

Под кумулятивностью понимается свойство данных небольшого объема достаточно полно (точно) отображать действительность.

Достоверность - степень безошибочности данных.

Противоречивость - это наличие двух взаимоисключающих понятий.

Приведенные показатели позволяют только оценить наличие явных ошибок в сети, которые возникают при кодировке и введении сети в ЭВМ, а также оценить возможное присутствие более сложных логических ошибок. Окончательное решение за человеком.

Количественные характеристики этих параметров можно получить, анализируя граф семантической сети.

Студент стремится выучить заданные ему темы с наибольшим пониманием и в кратчайшие сроки. Поэтому, разработанную сеть, после ее анализа и доработки необходимо оптимизировать так, чтобы время изучения было минимально, а понимание наиболее глубоким.

Для уменьшения времени изучения необходимо оптимизировать семантическую сеть, таким образом, чтобы уменьшить число анализа знаний (тестирование) и сократить (если это возможно, без нарушения понимания) число понятий. Для этого существуют следующие средства:

- замена определения понятия на более простое (с меньшим числом понятий, для объяснения основного понятия);

- если одно и то же понятие используется для объяснения нескольких понятий одного уровня, то можно обойтись одним его тестированием, а в других случаях уже брать готовый результат;
- путем сокращения типов понятий, которые объясняют понятие высшего уровня.

Следует отметить, что оптимизация не является формальной процедурой.

После того, как семантическую сеть разработали на бумаге, отдельные фрагменты сети вводятся в ЭВМ. Компьютер из этих фрагментов "сам" собирает всю сеть и анализирует качество, подсчитывая значение параметров. Если сеть некачественна, то отдельные фрагменты корректируются, и повторяется анализ.

После того как семантическая сеть будет отвечать некоторому уровню качества (отсутствие грубых ошибок, значения параметров лежат в заданных диапазонах), ее можно оптимизировать. После этапа оптимизации сети на основе библиотеки учебных фрагментов и тестов, синтезируется учебник.

2.3. Принципы структурирования учебного материала для электронного учебника

В основе структурирования учебного материала должна быть положена семантическая модель дисциплины, созданная на этапе построения навигационной системы. Она является результатом формализации процесса, то есть построения четкого формального описания процесса с необходимой степенью приближения к действительности, в результате которого мы должны получить библиотеку учебных фрагментов, которая привязывается к созданной семантической модели дисциплины.

Структуру определяют как совокупность стойких связей объектов, которые обеспечивают его целостность и тождественность самому себе, то есть сохранение основных

свойств, при разных внутренних и внешних изменениях. Способы структуризации могут быть разными в зависимости от исследуемого объекта, процесса, явления.

Структуризация:

- разбивка исследуемого объекта или явления на взаимосвязанные части (элементы)
- выбор совокупности характеристик
- построение системы отношений между выбранными элементами

В первую очередь необходимо решить вопрос о разбивке исследуемого объекта или явления на взаимосвязанные части (элементы).

Каждый элемент в модели должен быть обеспечен уникальным и значимым именем. Таким образом, обозначение, наименование объекта - это элементарная процедура, лежащая в основе информационного моделирования. Элемент является простой информационной моделью.

Следующий шаг - выбор совокупности характеристик, параметров (свойств) так, чтобы они обеспечивали удобство определения искомых элементов при исследовании объектов, процессов, явлений методом моделирования и давали возможность получить достаточно простую модель.

Последний шаг структуризация объекта - построение системы отношений (взаимоотношений) между выбранными элементами. Отношения могут быть пространственные и временные, причинно-следственные, части и целого, формы и содержания, внешнего и внутреннего и тому подобное. Для принятого нами подхода к построению ЭУ его семантическая структура представлена на рисунке 2.1.

В соответствии с нашей концепцией учебный материал прежде всего должен быть представлен как целостный текст, созданный в электронном варианте в традиционной структуре по темам, разделам и т.д., который затем структурируется под навигационную систему (табл.2.1.).

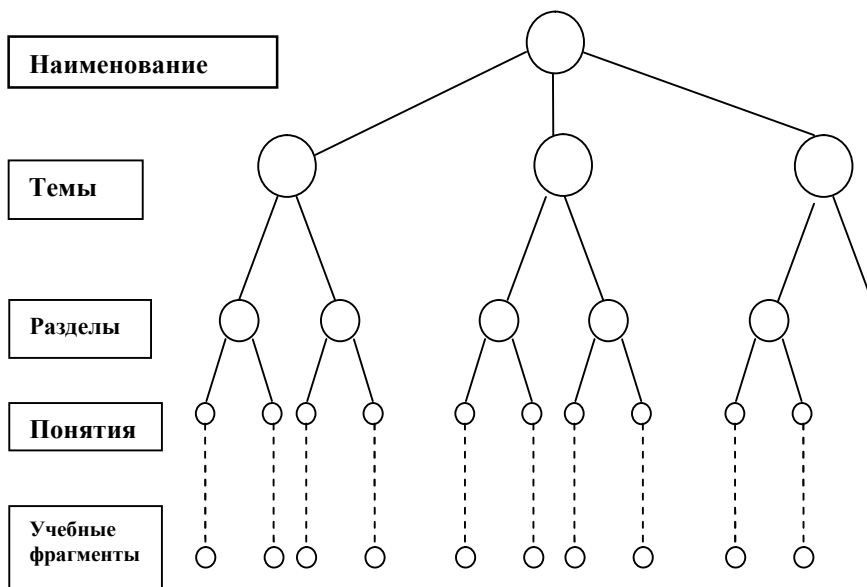


Рис.2.1 Семантическая структура электронного учебника

Таблица 2.1. Фрагменты учебного материала

№ п\п	Учебные фрагменты
1	Общий текст в электронном варианте
2	Общие рекомендации по изучению дисциплины
3	Семантическая связь входных знаний с изучаемыми по заданной дисциплине
3	Система входного контроля: вопросы, задачи, программы, имитационные материалы и т.д.
4	Рекомендации к изучению каждого понятия

5	Глоссарий
6	Порция информации для освоения выбранного понятия
7	Система контроля: вопросы, задачи, программы, имитационные материалы и т.д.

Первое требование к компоновке текста – это его максимальное «сжатие». Он не должен походить на традиционный «бумажный» учебник в пространной словесной форме. В русле концепции инженерии знаний рассматриваются всевозможные типы моделей представления знаний в «сжатом», компактном, удобном для использования виде. Среди них логическая модель, продукционная модель, фреймовая модель, модель семантической сети.

Технологическое сжатие учебной информации может быть достигнуто различными методическими приемами. Наиболее действенными из них зарекомендовали себя следующие методические приемы: моделирование в предметной, графической и знаковой форме, укрупненное упражнение, структурные блок-схемы, генеалогическое дерево и др. Вместе с тем следует учитывать тот факт, что при осуществлении «сжатия» учебного материала наибольшая прочность освоения достигается при подаче учебной информации на четырех кодах: рисуночном, числовом, символическом и словесном.

Второе условие – разделение материала учебника на отдельные модули при соблюдении его общей целостности. Это требование должно быть учтено уже при его разбиении на темы, разделы и подразделы. Принципом разбиения должна стать неперемность выхода раздела или подраздела на узловые понятия семантической модели, в крайнем случае они должны объединяться общей темой.

Третье условие – использование при формировании библиотеки учебных фрагментов общепринятых и известных непросвещенному пользователю программных продуктов. Это условие обуславливается тем, что подготовку учебных фрагментов выполняют преподаватели, часто не знакомые с

технологией программирования, но использующие в своей работе общедоступные программы Microsoft Office, поэтому желательно при создании лингвистического робота для формирования ЭУ это учитывать.

2.4. Построение семантической модели курса

Создание семантической модели курса выполнялось путем установки связи глоссария курса с содержанием дисциплины. Семантический анализ дисциплины выполнялся в следующей последовательности (таблицы 2.2. – 2.4.)

Таблица 2.2. Состав тем и разделов в курсе

№	Темы	№	Раздел
1	Роль ИС в управлении современной экономикой и бизнесом	1	Сущность и структура системы управления
		2	Информационная система: понятие, место в системе управления, функции
		3	Информация как предмет и продукт функционирования ИС
2	Понятие автоматизированной ИС (АИС), этапы развития, принципы построения.	4	Понятие автоматизированной ИС (АИС), этапы развития АИС, принципы построения АИС
3	Состав и структура ИС	5	Структура АИС
		6	Функциональные компоненты АИС
		7	Компоненты системы обработки данных (СОД)
		8	Организационные компоненты АИС
4	Классификация ИС	9	Классификация ИС
5	Тенденции в развитии ИС в соответствии с потребностями современного бизнеса.	10	Развитие подходов к технической и программной реализации ИС
		11	Развитие телекоммуникационных средств передачи данных
		12	Развитие информационных технологий в экономике
6	Обзор возможностей	13	Бухгалтерские ИС

современных ИС для конкретных бизнес-процессов	14	ИС управление бизнес-проектами
	15	Банковские ИС
	16	Электронная коммерция

Создание семантической модели следует выполнять поэтапно:

1 этап Должны быть изучены требования ОКХ (образовательно-квалификационная характеристика) и ОПП (образовательно-профессиональная программа) для студентов-информатиков. В результате была за основу принята структура электронного учебника, представленная в таблице 2.2.

2 этап. На основе изучения глоссария курса был определен его основной понятийный аппарат, который затем должен составить семантический граф дисциплины. Выделяется он из разделов курса, затем уставляются между понятиями семантические отношения, определяются «узловые» понятия путем определения наличия и числа входов в каждое понятие (вершину графа) и точки проведения контроля усвоения этих понятий. Проведенное исследование представлено в таблице 2.3.

Таблица 2.3. Семантическая модель тезауруса курса

№ понятия	Понятия	№ раздела	Вход в понятие	Число входов	Точки контроля
1	сложная система	1	7		
2	управление	1	7		
3	структура управления	1	7		
4	уровень управления	2	7		
5	прямая связь	1	7		
6	обратная связь	1	7		
7	информационная система	2	0	5	7
8	функция учета	2	22		
9	функция планирования	2	22		
10	функция анализа и регулирования	2	22		
11	информация,	3	7	6	11

12	данные	3	11		
13	синтаксический уровень информации	3	11		
14	семантический уровень информации	3	11		
15	прагматический уровень информации	3	11		
16	количественная мера информации	3	11		
17	тезаурус	3	11		
18	информационная технология (ИТ)	4	19	10	18
19	автоматизированная информационная система (АИС)	4	7	5	
20	автоматизированное рабочее место (АРМ)	4	24		
21	программное обеспечение (ПО) СОД	9	19		
22	функциональные компоненты АИС	7	19	6	22
23	компоненты системы обработки данных (СОД)	7	19	4	23
24	организационные компоненты АИС	7	19	4	24
25	функциональная структура АИС	8	22		
26	электронный документооборот	8	22		
27	функциональная задача	8	22		
28	информационное обеспечение (ИО) СОД	9	23		
29	правовое обеспечение СОД	9	23		
30	лингвистическое обеспечение СОД	9	23		
31	техническое обеспечение СОД	9	23		
32	одиночные ИС	11	24		
33	групповые ИС	11	24		
34	корпоративные ИС (КИС)	11	24		
35	системы оперативной обработки данных — OLTP	11	18		
36	системы поддержки и принятия решений — DSS	11	18		
37	хранилище данных	11	18		
38	система обработки транзакций	11	18		
39	профессиональные системы	11	18		
40	информационно-справочные системы – ИСС	11	18		
41	офисные АИС	11	18		
42	мультисервисные сети	13	18		
43	системы поддержки исполнения решений (EPSS)	14	18		
44	МА-технологии	12	18		

Таблица 2.4. Окончательная семантическая модель курса

Понятия										
№ понятия	Понятие	№ раздела	Вход	Число входов	Точки контроля	Источник	Цель	Контроль	ПО	Видео
1	сложная система	1	7			1_1	Ц1_1			
2	управление	1	7			1_1	Ц1_1			
3	структура управления	1	7			1_1	Ц1_1			
4	уровень управления	2	7			1_1	Ц1_1			
5	прямая связь	1	7			1_1	Ц1_1			
6	обратная связь	1	7			1_1	Ц1_1			
7	информационная система	2	0	5	7	1_1	Ц1_1	K1		
8	функция учета	2	22			1_1	Ц1_1			
9	функция планирования	2	22			1_1	Ц1_1			
10	функция анализа и регулирования	2	22			1_1	Ц1_1			
11	информация,	3	7	6	11	1_1	Ц1_1	K1		
12	данные	3	11			1_1	Ц1_1			
13	синтаксический уровень информации	3	11			1_1	Ц1_1			
14	семантический уровень информации	3	11			1_1	Ц1_1			
15	прагматический уровень информации	3	11			1_1	Ц1_1			
16	количественная мера информации	3	11			1_1	Ц1_1			
17	тезаурус	3	11			1_1	Ц1_1			
18	информационная технология (ИТ)	4	19	10	18	1_3	Ц1_2	K2		
19	автоматизированная информационная система (АИС)	4	7	5	19	1_2	Ц1_2	K2		

Понятия

№ понятия	Понятие	№ раздела	Вход	Число входов	Точки контроля	Источник	Цель	Контроль	ПО	Видео
20	автоматизированное рабочее место (АРМ)	4	24			1_2	Ц1_2			
21	программное обеспечение (ПО) СОД	9	19			1_3	Ц1_3			
22	функциональные компоненты АИС	7	19	6	22	1_3	Ц1_3	КЗ		
23	компоненты системы обработки данных (СОД)	7	19	4	23	1_3	Ц1_3	КЗ		
24	организационные компоненты АИС	7	19	4	24	1_3	Ц1_3	КЗ		
25	функциональная структура АИС	8	22			1_3	Ц1_3			
26	электронный документооборот	8	22			1_3	Ц1_3			
27	функциональная задача	8	22			1_3	Ц1_3			
28	информационное обеспечение (ИО) СОД	9	23			1_3	Ц1_3			
29	правовое обеспечение СОД	9	23			1_3	Ц1_3			
30	лингвистическое обеспечение СОД	9	23			1_3	Ц1_3			
31	техническое обеспечение СОД	9	23			1_3	Ц1_3			
32	одиночные ИС	11	24			1_4	Ц1_4			
33	групповые ИС	11	24			1_4	Ц1_4			
34	корпоративные ИС (КИС)	11	24			1_4	Ц1_4			
35	системы оперативной обработки данных — OLTP	11	18			1_4	Ц1_4			
36	системы поддержки и принятия решений —	11	18			1_4	Ц1_4			

Понятия

№ понятия	Понятие	№ раздела	Вход	Число входов	Точки контроля	Источник	Цель	Контроль	ПО	Видео
	DSS									
37	хранилище данных	11	18			1_4	Ц1_4			
38	система обработки транзакций	11	18			1_4	Ц1_4			
39	профессиональные системы	11	18			1_4	Ц1_4			
40	информационно-справочные системы – ИСС	11	18			1_4	Ц1_4			
41	офисные АИС	11	18			1_4	Ц1_4			
42	мультисервисные сети	13	18			1_5	Ц1_5			
43	системы поддержки исполнения решений (EPSS)	14	18			1_5	Ц1_5			
44	МА-технологии	12	18			1_5	Ц1_5			
45	MRP-системы	14	49			1_5	Ц1_5			
46	MRP II-системы	14	49			1_5	Ц1_5			
47	ERP-системы	14	49			1_5	Ц1_5			
48	CSRP-системы	14	49			1_5	Ц1_5			
49	экономическая АИС	2	19	9	49	1_6	Ц1_6	К4		
50	компьютерная система бухгалтерского учета (КСБУ)	15	49			1_6	Ц1_6			
51	управление проектами	16	49			1_6	Ц1_6			
52	автоматизированная банковская система (АБС)	17	49			1_6	Ц1_6			
53	электронная коммерция	18	49			1_6	Ц1_6			
54	электронный магазин	18	49			1_6	Ц1_6			

2.6. Программная реализация оболочки для создания семантической модели курса

Элементы представленной выше модели были реализованы с помощью использования программных возможностей СУБД ACCESS и Web-технологий.

Схема данных, которая реализует семантическую модель дисциплины, программно состоит из двух групп таблиц ACCESS :

- отражающих структуру курса;
- отражающих содержание курса.

Объединяет обе группы таблица «Понятия»

В первую группу входят:

- Ø Тема - содержит поля «№ Темы», «Название темы»;
- Ø Раздел – содержит поля «№ Раздела», «Название темы», «№ Темы»;
- Ø Понятия – содержит поля «№ понятия», «№ раздела», «Вход», «Число входов», «Точки контроля», и т.д;
- Ø Определения – содержит поля «№ определения», «Определение понятия»;

Информация, содержащаяся в первой группе таблиц, используется затем при построении интерфейса ЭУ. Структура таблиц представлена на рисунке 2.3. Вторая группа таблиц отражает фрагменты учебного курса и устанавливает связь с тезаурусом дисциплины. В совокупности первая и вторая группа таблиц представляет собой лингвистическую модель ЭУ. Предварительно учебный курс разбивается на фрагменты в соответствии с дидактикой преподавания дисциплины: рекомендации, материал, контроль, программное обеспечение, видео-клипы. Наполнение этих фрагментов может осуществляться с использованием различных инструментов по желанию и уровню компьютерной грамотности преподавателя: гипертекста, технологий мультимедиа, графических редакторов, языков программирования. Эти фрагменты хранятся в отдельных файлах и папках и затем привязываются к тезаурусу курса с помощью гипертекстовых ссылок.

Таким образом, с помощью MS Access мы имеем возможность экспортировать таблицы и запросы в формат HTML и создавать страницы доступа к данным, которую можно просматривать с помощью браузера Internet Explorer. Общая картина программной реализации лингвистической модели отражена в схеме MS Access на рисунке 2.5.

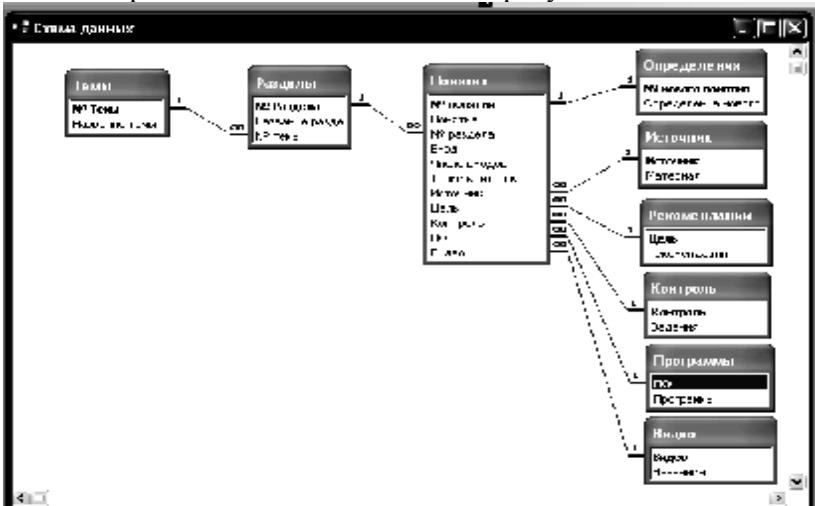


Рис.2.5. Схема данных, реализующая семантическую модель. Центральное место в схеме данных занимает таблица «Понятие», поэтому считаем целесообразным более детально пояснить ее структуру(рис.2.6)

Понятия: таблица

Имя	Тип данных	Комментарий
Иллюстрация	Ссылка	Ссылка на иллюстрацию
Исключения	Ссылка	Ссылка на исключения
Программа	Ссылка	Ссылка на программу
Индекс	Ссылка	Ссылка на индекс
Имя	Текст	Имя понятия
Многочлен	Текст	Составляющие понятия
Раздел	Текст	Ссылка на раздел
Тема	Текст	Ссылка на тему
Иллюстрация	Текст	Ссылка на иллюстрацию
Исключения	Текст	Ссылка на исключения
Программа	Текст	Ссылка на программу
Индекс	Текст	Ссылка на индекс

Рис. 2.6. Структура таблицы «Понятия»

Для удобства пользователя при формировании семантической модели курса создана специальная кнопочная форма (рис. 2.7)

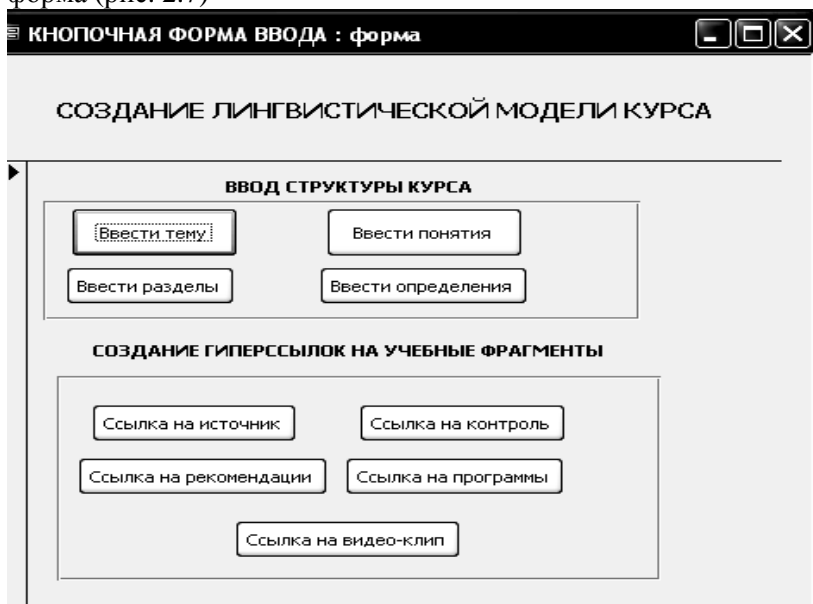


Рис. 2.7. Интерфейс создания семантической модели курса

2.7. Описание навигационной системы курса

В выбранной нами навигации ЭУ реализована следующая концепция построения интерфейса. Весь проект разбит на 2 логических составляющие:

Мастер создания курса, который помогает преподавателю сформировать учебный материал для электронного учебника на основе созданной лингвистической модели дисциплины.

Интерфейс курса, с помощью которого студент сможет освоить дисциплину.

Мастер создания курса

Весь процесс создания мастера интерфейса разбит на определенные шаги в виде схемы, представленной на рисунке 2.8.

Первый шаг: «Общие сведения» – дается «название курса» и создаются папки проекта и учебного материала. Обязательное условие: чтобы все поля были заполнены и каталог проекта сначала должен быть пустым. Для проверки соблюдения условия служит специальная процедура. После указания директорий проекта и материалов, необходимо скопировать предварительно созданные преподавателем учебные материалы в папку проекта, делается это специально, чтобы сделать курс автономным, не привязанным к определенному пути на жестком диске (или на другом носителе) Перед переходом ко второму шагу будет создана БД.

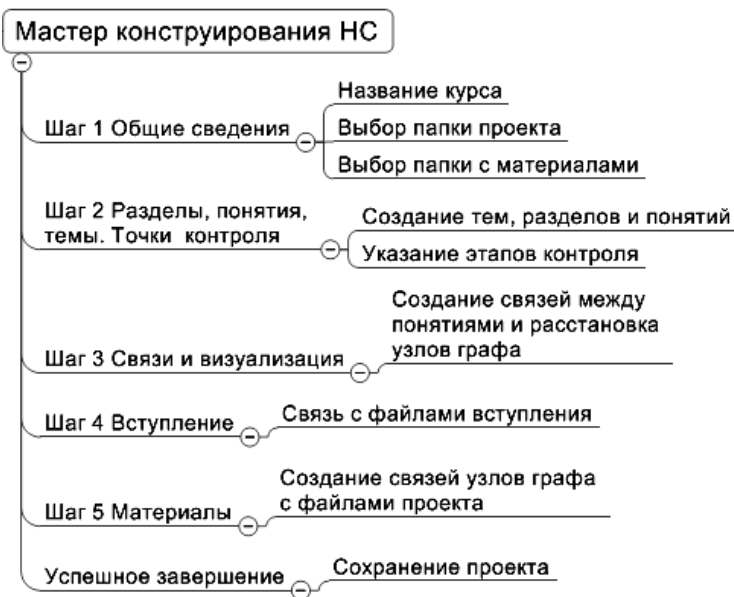


Рис.2.8. Схема процесса конструирования навигационной системы ЭУ

Второй шаг: «Разделы, понятия, темы. Точки контроля». На втором шаге проекта преподаватель вносит в

проект предварительно созданную им семантическую модель курса в виде тем, разделов и понятий и определяет этапы контроля усвоения материала. Для организации этой работы выбран компонент библиотеки VCL – “TTreeView”. Добавление тем разделов и понятий осуществляется с помощью дополнительного компонента “TMemo”. Так же для работы с деревом предусмотрены специальные процедуры редактирования, удаления и процедуры, которые контролируют внешний вид дерева.

Третий шаг: «Связи и визуализация». Для того, чтобы преподаватель смог представить понятийный аппарат в виде дерева, необходимо каким-то образом представить узлы графа в виде понятий с возможностью создания связей в экранном варианте. Третий шаг реализован с помощью специального компонента “ACTIVEХ” – “Flash”. Расстановка узлов графа (понятий) на экране происходит не в автоматическом режиме, а в ручном, что позволяет преподавателю точнее и логичнее расставить все узлы графа, в связи со спецификой конкретного предмета, что практически невозможно сделать в автоматическом режиме. Каждым узлом является круг, внутри которого стоит порядковый номер, соответствующий понятию, которое находится в левой части экранной формы программы (список понятий), так что преподавателю будет легко сориентироваться.

Четвертый шаг: «Вступление». После того, как понятия расставлены и все связи сформированы в экранном варианте, мастер переходит к следующему шагу – обеспечению связи НС со вступлением к учебному курсу, который преподаватель готовит отдельно и в котором дает студенту общие рекомендации для успешного усвоения материала.

Пятый шаг: «Материалы». На этом шаге происходит связь понятий с базой знаний, файлами контрольных точек и рекомендациями для усвоения. Все три поля (материалы для усвоения, материалы понятия, материалы контроля) для введения представлены в виде компонента TRzButtonEdit (RZ components), который отличается от стандартного компонента TEdit библиотеки компонентов VCL. Этот компонент как нельзя кстати подходит на роль выбора файла с его помощью,

поскольку у него есть специальная встроенная кнопка, вид которой можно редактировать в панели настроек компонента среды “CodeGear”. Диалоговое окно включает следующие вопросы, на которые надо дать ответ:

- Название понятия
- Название раздела
- Название темы
- Порядковый номер понятия
- Определение понятия
- Параметр усвоения понятия
- Координаты понятия (на графе)
- Ссылка на понятие (материалы усвоения и контроля)
- Связки понятия

На выполнении пятой процедуры работа с мастером заканчивается и происходит сохранение проекта. В результате работы мастера создается электронный учебник, навигатор которого ведет студента не по темам и разделам, а по тезаурусу дисциплины и студент может самостоятельно составить для себя план усвоения материала, исходя из уровня подготовленности в этой предметной области.

Описание интерфейса электронного учебника.

Организация интерфейса - очень важное потребительское свойство программного продукта учебного назначения. Предполагается, что с программой работает неквалифицированный пользователь, а число сеансов работы с одной программой обычно относительно невелико, так что особое значение приобретает скорость и легкость освоения управлением программой. Поэтому необходимо обеспечить возможность сконцентрироваться на исследуемом предмете и как можно меньше думать о способах общения с компьютером. Кроме того, интерфейс должен отражать специфику изучаемой дисциплины, в частности адекватно отображать принятый в дисциплине язык.

Основными факторами, которые определяют удобства работы пользователя в диалоговом интерфейсе, есть следующие:

1. Гибкость диалога.
2. Ясность диалога.
3. Легкость обучения и использования.
4. Надежность.
5. Стандартизация интерфейса.

Признаки удобного интерфейса:

- использование пиктограмм для организации управляющего диалога;
- меню-ориентированность;
- контекстно-зависимая справочная система (Help);
- возможность использования для ввода как мыши, так и клавиатуры с использованием "горячих клавиш", стрелок, клавиша табуляции и т.п.
- единые правила работы со всеми меню;
- ESC или игнорируется, или используется только для выхода из любого режима на более высокий уровень с отменой сделанных изменений;
- запрос подтверждения в опасных ситуациях (выход из программы, потеря информации и т.п.);
- возможность прерывания продолжительных процессов;
- корректная обработка аварийных ситуаций внешних устройств (например, неготовность принтера);
- обработка некорректных действий пользователя.

Интерфейс не должен содержать отвлекающих элементов. Если ЭУ предназначен для применения на занятии, то все внимание обучаемого должно быть сосредоточено только на исследуемом материале.

На наш взгляд, выбранный нами интерфейс учитывает все выше перечисленные требования.

Окно интерфейса является достаточно удобной системой навигации по курсу. Окно делится на 3 визуальных части (рис.2.9).

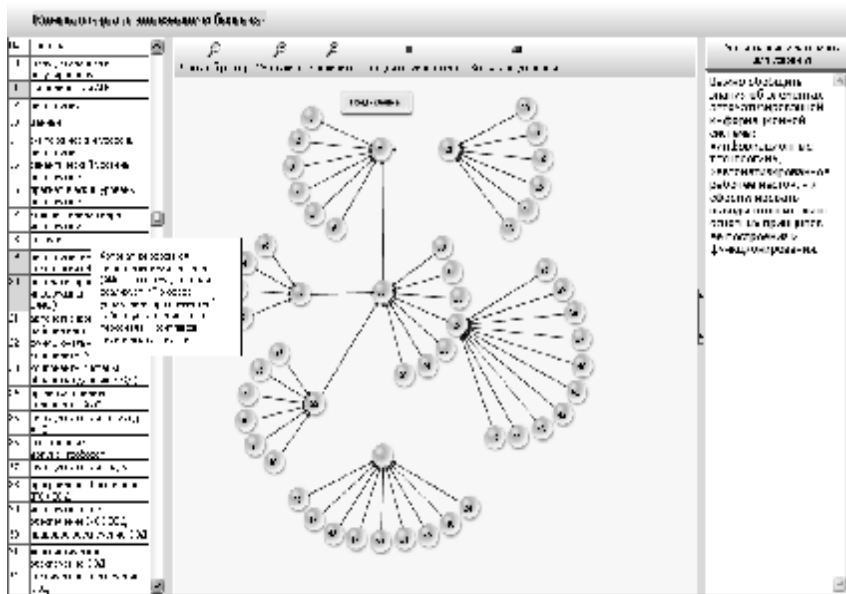
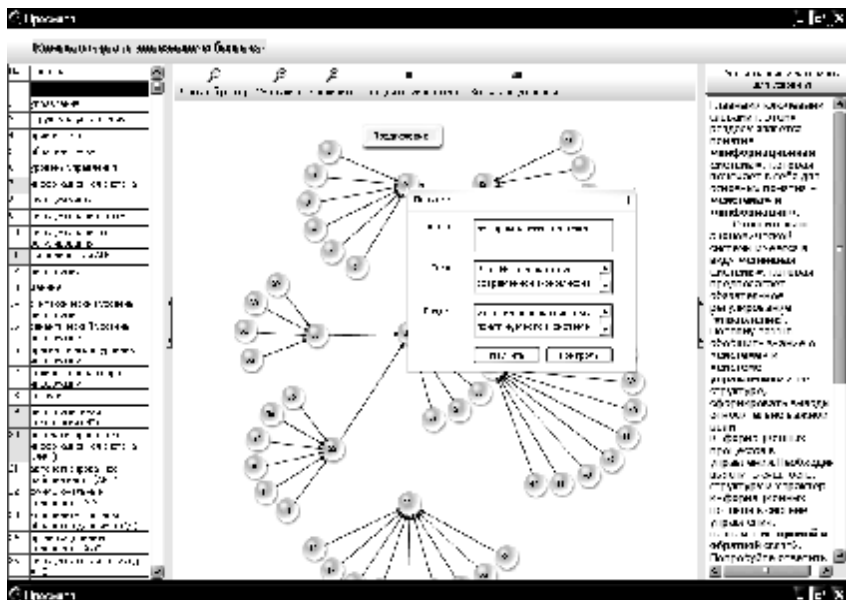


Рис. 2.9 Примеры интерфейса ДК, построенного на основе семантической модели и WEB-технологий

Часть 1 Список понятий (расположен в левой части окна)

Список состоит из номеров понятий и их названий. Точки контроля, которые существуют на определенных понятиях, окрашены в синий цвет, все пройденные понятия окрашивающиеся в зеленый цвет. При наведении курсора мыши на понятие, студент сразу получает определение этого понятия в виде выплывающей подсказки. Опираясь на данный список, студент легко может ориентироваться при работе с графом.

Часть 2 Визуальное представление системы понятий находится в центре. Это ключевой компонент, с помощью которого происходит навигация по курсу с помощью гиперссылок и диалоговых окон. Он представлен в виде графа, узлы которого являются понятиями. Каждое понятие связано с определенным фрагментом учебного материала, который позволяет его освоить, для этого необходимо вызвать диалоговое окно, щелкнув кнопкой мыши по вершине, соответствующей номеру понятия, после появления диалогового окна, необходимо нажать на кнопку «Изучить». Если это понятие является контрольным, тогда будет присутствовать кнопка «Контроль», при нажатии на которую отобразятся тесты или вопросы или программа для контроля (рис.2.10.).

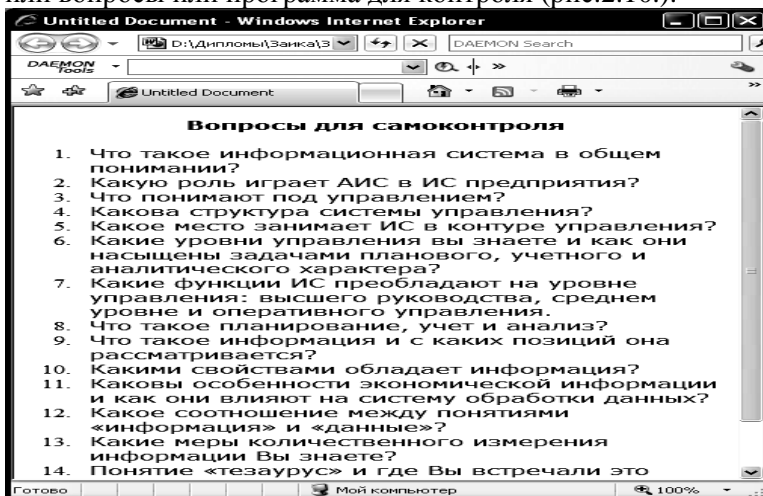
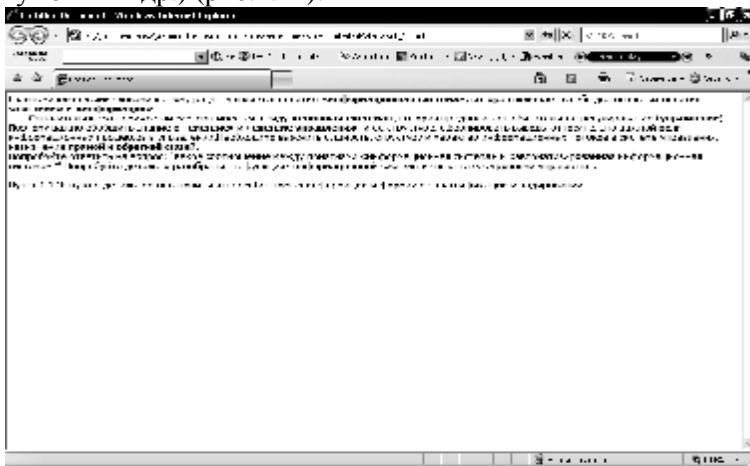


Рис.2.10. Пример вывода контролирующего фрагмента курса

Часть 3 Рекомендации для усвоения. Панель рекомендаций активируется при вызове диалогового окна любого из понятий. При этом в панели начинает отображаться информация для усвоения (гипертекстовый, текстовый, видео-документы и др.) (рис.2.11).



2.11. Пример представления рекомендаций

2.8. Технология создания навигационной системы курса.

Программа делится на два блока. Первая часть - администрирование, это конструктор, который помогает преподавателю составить весь курс, то есть создать для него навигатор. Вторая часть - это навигатор для всего курса, с помощью которого студент сможет выучить дисциплину.

Конструктор построен по принципу лингвистического робота, который способен воспринять смысловые модели конкретного предмета и содержать типовые алгоритмы обучения, реализующие соответствующие методики обучения.

Настройка лингвистического робота как оболочки на конкретный предмет заключается в построении проекта

электронного учебника в виде модели требуемых знаний на языке внешнего представления семантических моделей и вводе его в робот. В результате работы с ним создается навигационная система ЭУ.

На каждом этапе работы с конструктором выполняются определенные шаги, которые приводят к конечному результату, а именно созданию навигационной системы для курса. Рассмотрим процесс создания НС на примере курса “Компьютеры в экономике и бизнесе”.

В конструкторе всего существует 5 шагов.

Шаг 1: Общие сведения о дисциплине

В этом пункте устанавливаются базовые параметры курса, а именно расположение курса на жестком диске, путь к материалам курса и название курса.

Дальше (после того, как полностью отработает конструктор) расположение курса может меняться, то есть программа не привязывается к определенному пути на жестком диске.

Материалы курса - это все те файлы, которые будут использоваться в качестве базы знаний всего курса. Этими материалами могут быть абсолютно любые типы файлов (гипертекстовые документы, аудио-, видео-файлы, обычный текстовый документ и так далее) и даже ссылки на ресурсы Интернет. В нашем примере это будут отдельные фрагменты курса, созданные в виде гипертекстовых документов в Word. Материалы будут размещены в папке «Материалы».

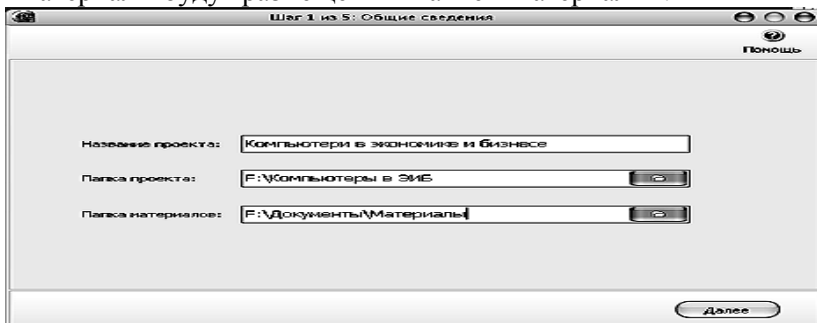


Рис. 2.11 Ввод общих сведений об электронном учебнике

Шаг 2: Создание тем, разделов и понятий

Окно этого шага делится на две части: верхняя, нижняя. В верхней части окна все ключевые моменты базы знаний структурируется в виде дерева знаний. Первый уровень дерева составляют темы, второй разделы и третий понятие.

Дерево может редактироваться после добавления у него пунктов. Пункт дерева может быть удален или изменено его название.

Наполнение дерева происходит с помощью поля, которое находится в нижней части экранной формы. Здесь есть возможность добавлять сразу несколько понятий тем или разделов одновременно (разделение названий происходит с помощью перенесения строки).

В этом же шаге устанавливаются точки контроля. Напротив каждого понятия можно поставить галочку, тем самым, отметив его как этап контроля. В шаге 5, к этому понятию, отмеченному галочкой, будет возможность добавлять контрольный материал, это могут быть вопросы для самоконтроля, тесты и любые другие материалы.

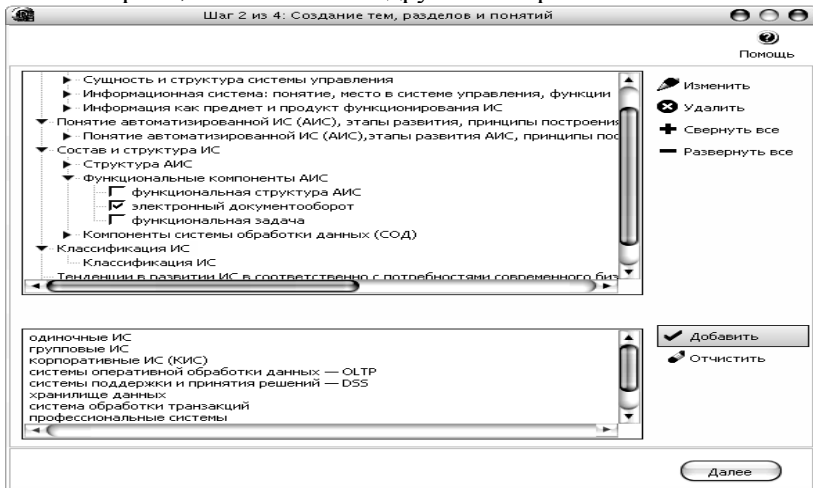


Рис.2.12 Создание тем, разделов и понятий и установка точек контроля

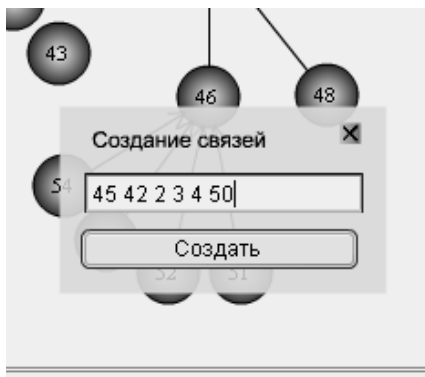


Рис.2.14 Диалоговое окно создания связей

Шаг 4: Связь с файлами вступления

Преподавателю необходимо выбрать файлы для связи со вступлением, а именно: файл материалы для вступления и файл входного контроля. (рис. 2.15)



Рис. 2.15 Связь с файлами вступления

Шаг 5 Связь с файлами и определение

Здесь преподавателю надлежит выбрать для каждого понятия соответствующий ему материал, а также файл рекомендаций для усвоения. Если это понятие контрольное, тогда программа предложит выбрать файл для контроля.

Перемещаться между понятиями можно с помощью списка понятий в левой части экранной формы программы.

Полет “Номер текущего понятия” сигнализирует о том, какое поле в настоящий момент активно и для какого понятия мы устанавливаем связки с материалами.

В нижней части формы необходимо также задать определение понятия (рис. 2.16).

Шаг 5 из 5: Связь с файлами

Помощь

№	Понятия
1	сложная система
2	управление
3	структура управления
4	прямая связь
5	обратная связь
6	уровень управления
7	информационная система
8	функция учета
9	функция планирования
10	функция анализа и регулирования
11	экономическая АИС
12	информация
13	данные
14	синтаксический уровень инфо
15	семантический уровень инфо
16	прагматический уровень инфо
17	количественная мера инфо

Номер текущего понятия: 5

Файлы понятия

Материал: Data\Materials\2.html

Рекомендации: Data\Materials\1_2.html

Контроль:

Определение понятия:

Обратная связь представляет собой поток отчетной информации о выполнении принятых решений, направляемый от объекта управления к управленческому аппарату

Сохранить

Далее

Рис.2.16 Связь с файлами

После того, как будет пройден последний 5й шаг, будет отображено последнее окно, которое сообщит, что проект успешно создан и готов к использованию. Будет выведен (как напоминание) полный путь к файлу проекта, где его можно найти (рис. 2.17).

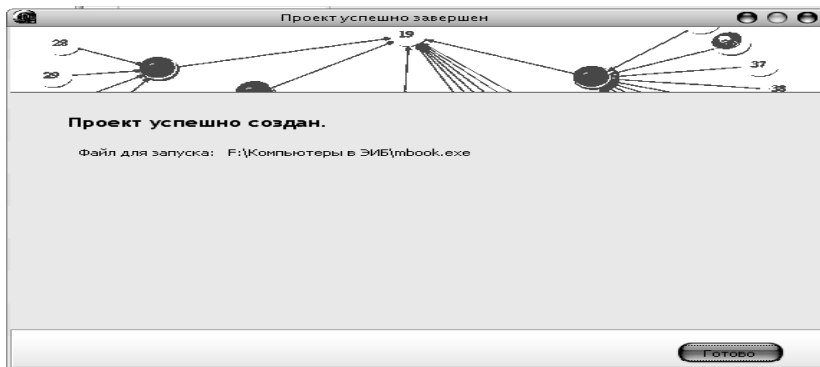


Рис 2.17 Финальное окно

На этом создание второй части - навигатора завершится.

Лабораторная работа №1 ***Создание семантической модели учебного курса***

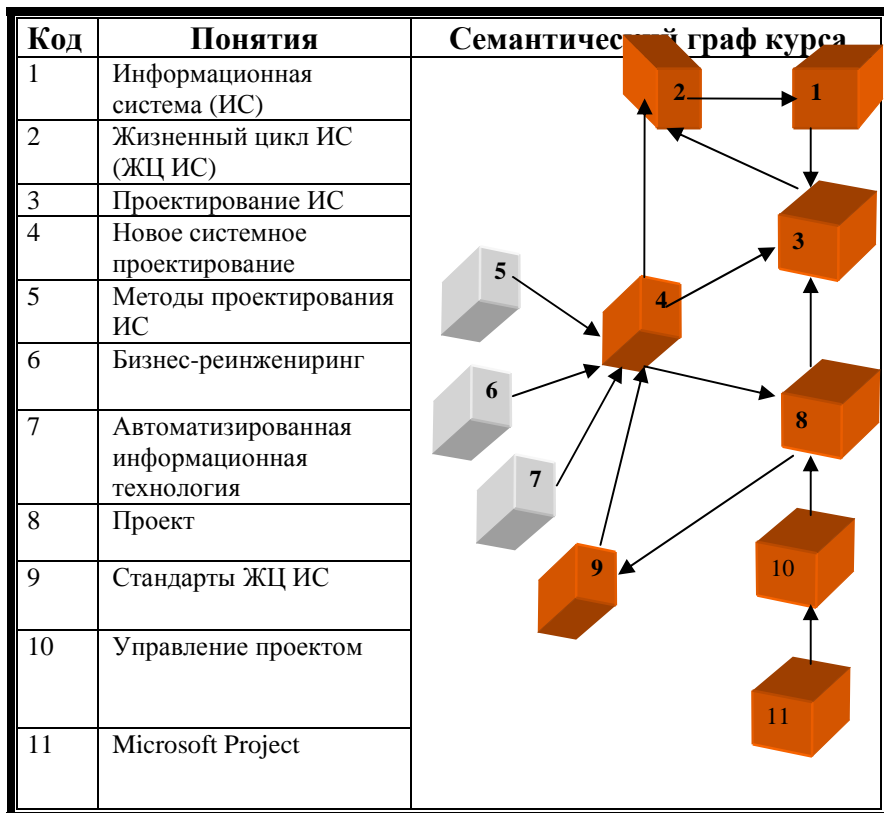
Цель работы:

Смоделировать технологию создания семантической модели части курса «Кибернетические основы информационных технологий», например, «Применение кибернетических аналогий в проектировании информационных систем (ИС

Используя содержание данного раздела выделить его структуру (темы и разделы), выделить понятийный аппарат и определить связи между ними и узловые понятия и составить глоссарий

раздела, определить точки контроля. Создать граф-модель раздела.

В качестве помощи предлагается начальный вариант граф-модели раздела, его следует дополнить.



Преподавателю представить полученное решение и его обоснование.

Лабораторная работа № 2

Тема. Создание фрагментов учебного материала и навигатора курса

Цель работы:

1. Представить вариант создания учебных фрагментов по разделу «Применение кибернетических аналогий в проектировании информационных систем (ИС)», в гипертекстовом варианте. Он должен включать список фрагментов, представленных в таблице.

№	Учебные фрагменты
1	Общий текст в гипертекстовом варианте
2	Общие рекомендации по изучению дисциплины
3	Семантическая связь входных знаний с изучаемыми по заданной дисциплине
3	Система входного контроля: вопросы, задачи, программы, имитационные материалы и т.д.
4	Рекомендации к изучению выбранного понятия
5	Глоссарий
6	Порция информации для изучения выбранного понятия
7	Система контроля: вопросы, задачи, программы, имитационные материалы и т.д.

2. Используя программу в Access создать семантическую модель курса и представить ее преподавателю.

3. Используя программу «Навигатор» создать навигатор по курсу

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.

ТЕМЫ РЕФЕРАТОВ

1. Истоки кибернетики и ее понятийный аппарат
2. Информационный подход к процессам управления в кибернетике и его приложение к ИТ
3. Основные черты кибернетики и их связь с информационными технологиями
4. Математические основы, модели кибернетики и их реализация в ИТ
5. Теория информационно-поисковых систем и их роль в Интернет
6. Этапы становления кибернетической педагогики и прогноз ее развития
7. Использование кибернетических моделей в системе обучения
8. Понятие, формы представления и количественная оценка информации
9. Семантический и прагматический подходы к оценке количества и качества информации и сферы их использования в ИТ
10. Информация, информационные процессы и информационные системы как основной понятийный аппарат современных ИТ
11. Теория классификации и кодирования информации и ее роль в СУБД, Интернет и других информационных технологиях
12. Кибернетическая система и роль процессов хранения информации в кибернетических системах
13. Направления влияния кибернетики на развитие информационных технологий.
14. Роль стандартов и кибернетических моделей в проектировании информационных систем
15. Обзор современных систем стандартов, регламентирующих проектирование программных комплексов

ПРОЕКТНЫЕ ЗАДАНИЯ.

Вариант 1. Внедрение новой информационной системы Виды работ

№ п/п	Наименование работ	Продолжительность	Предшествующая	Исполнитель
1	Начало работ	0d		
2	Назначение руководителя проекта	4w	1	Психологи
3	Поставить задачи	3w	2	Ученые[2]; Менеджеры
4	Собрать необходимую информацию	8w	3	Социологи [3]
5	Определить потребности	6w	3	Ученые[2]
6	Рассмотреть варианты решений	4w	5	Ученые[2]
7	Оценить варианты технических средств	3w	4; 6	Технологи[2]
8	Разработать программные решения	12w	4; 6	Ученые[2] Программист[1]
9	Разработать инструкции	8w	4; 6	Менеджеры [2]
10	Установить оборудование и ПО	6w	7; 8	Технологи[2]; - Наладчики; Программисты[2]
11	Опробовать компьютерную систему	4w	10	Программисты; Наладчики

12	Подготовить персонал	6w	9; 8	Менеджеры [2]
13	Полностью сдать систему в эксплуатацию	4w	12	Технологи[3]; Менеджеры [2]; Наладчик
14	Завершение проекта	0d	13	

Виды ресурсов

Имя_ресурса	Мах едниц	Почасовая ставка, грн/день	Стоимость, грн	Способ оплаты	Календарь
Ученые	200%		25 000,00	По окончании	стандартный
Социологи	300%	100,00		пропорционально	стандартный
Менеджеры	200%	150,00		пропорционально	стандартный
Программисты	200%		20 000,00	По окончании	стандартный
Наладчики	100%	80,00		пропорционально	стандартный
Технологи	300%	130,00		пропорционально	стандартный
Психологи	100%		5 000,00	По окончании	стандартный

Вариант 2.Создание программного обеспечения

Виды работ

№ п/п	Наименование работ	Продолжительность	Предшествующая	Исполнитель
1	Назначить руководителя проекта	1d		Директор[1]
2	Поставить задачи	3d	1	Руководитель[1]
3	Собрать необходимую информацию	8d	2	Менеджер [1]

4	Определить потребности	6d	2	Менеджер[1]
5	Рассмотреть варианты решений	6d	4	Руководитель[1], Программисты[2]
6	Оценить варианты технических средств	3d	3;5	Инженеры[2]
7	Разработать программное обеспечение	12d	3;5	Программисты [2]
8	Разработать инструкции	8d	3;5	Программисты [2]
9	Установить оборудование и ПО	6d	6;7	Инженеры[2]
10	Опробовать компьютерную систему	4d	9	Программисты [2], Инженеры [2]
11	Подготовить персонал	6d	8;10	Руководитель[1] Программисты [2]
12	Полностью сдать систему в эксплуатацию	4d	11	Руководитель[1] Программисты [2] Инженеры[2]

Ресурсы

Имя ресурса	Мах единиц	Почасовая ставка, грн	Способ оплаты	Календарь
Директор	1	50,00	Пропорционально	Стандартный
Руководитель	1	40,00	Пропорционально	Стандартный
Программист	2	35,00	Пропорционально	Стандартный
Инженер	2	35,00	Пропорционально	Стандартный

ЛИТЕРАТУРА

Литература к первой части учебного пособия

1. Винер Н. Кибернетика, или управление и связь в животном и машине /Н. Винер; [пер. с англ. И.В. Соловьева и Г.Н. Поварова]. - М.: «Советское радио», 1968.– 213 с.
2. Винер Н. Кибернетика и общество./Н. Винер; [пер. с англ. Е.Г.Панфилова].–М.: «Изд-во иностр. лит-ры», – М., 1958. – 30 с.
3. Винер Н. Мое отношение к кибернетике: Ее прошлое и будущее /Н. Винер; [пер. с англ. Г.Н.Поварова]. – М.: «Советское радио», 1969. -18 с.
4. Воловник А.А. Знакомьтесь, информационные технологии / А.А. Воловник. – СПб.: БХВ-Петербург, 2002. – 352 с.
5. Кузин Л.Т. Математические основы кибернетики. Учебное пособие для студентов вузов. Т.1 /Л.Т. Кузин. – М: «Энергия», 1973.- 504 с.
6. Математика и кибернетика в экономике. Словарь-справочник. - М.: Экономика, 1975.– 700с.
7. Ляпунов А.А. Проблемы теоретической и прикладной кибернетики /А.А. Ляпунов. – М.: Наука, 1980.– 335 с.
8. Турчин В.Ф. Феномен науки: Кибернетический подход к эволюции. Изд. 2-е /В.Ф. Турчин. - М.: ЭТС, 2000. - 368 с.
9. Урсул А.Д. Проблема информации в современной науке. Философские очерки /А.Д. Урсул. - М.: Наука, 1975. -287 с.
10. Фридланд А.Я. Основные ресурсы информатики: учебное пособие /А.Я. Фридланд. - М.: Астрель: Профиздат, 2005.– 283 с.
11. Чернавский Д.С. Синергетика и информация /Д.С. Чернавский. - М.: Знание, 1990. - 45 с.

Литература ко второй части учебного пособия

1. Богданов В. Управление проектами в Microsoft Project 2003 /В. Богданов. - Питер, 2004. - 608 с.

2. Гультьяев А.К. Microsoft Office Project 2003 Professional. Управление проектами: практическое пособие /А.К. Гультьяев. - Корона Принт – 2004. - 507 с.
3. Информационные системы /В.Н. Петров. – СПб.: Питер, 2002.–688 с.
4. Меняев М. Ф. Управление проектами. MS Project: учебное пособие /М. Ф. Меняев. - Омега-Л, 2005. - 156 с.
5. Орлов С.А. Технологии разработки программного обеспечения: учебник /С.А. Орлов. – СПб.: Питер, 2002. – 464 с.
6. Проектування інформаційних систем: посібник / За редакцією В.С.Пономаренка. – К.: Видавничий центр «Академія», 2002.– 488 с. (Альма-матер)
7. Скотт К. UML. Основные концепции /К..Скотт. – М.: Издательский дом «Вильямс», 2002. -144 с.
8. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г.Буч , Дж. Рамбо. – СПб.: Питер, 2002. – 496 с.;

Литература к третьей части учебного пособия

1. Зеневич А.М. Математическое моделирование процесса обучения: материалы конференции «Подготовка научных кадров высшей квалификации с целью обеспечения инновационного развития экономики» /А.М. Зеневич, С.Я. Жукович. – Мн.: ГУ «БЕЛИСА», 2006.– 146 с.
2. Семикин В.А. Семантическая модель контента образовательных электронных изданий. Автореферат диссертации на соискание ученой степени кандидата технических наук /В.А. Семикин. – Тюмень, 2004.–21 с.
3. Пашет О. Теория графов /О. Пашет. - М.: Наука, Главная редакция физико-математической литературы, 1980, - 336с.
4. Циганкова С.О. Підходи до створення навігаційної системи електронного підручника на основі семантичної моделі //С.О. Циганкова, А.П. Зайцев, В.О. Пчелов. –

Вісник Луганського національного університету імені Тараса Шевченка, №1(188) 2010, с. 252-258.

5. Цыганкова С.А. Концепция и методы создания электронного учебника на основе лингвистической модели и ее программной реализации //С.А. Цыганкова, Ю.В. Придорожко. – Вісник Східноукраїнського національного університету імені Володимира Даля, №2 (145) 2011, с. 271- 280.

ПРИЛОЖЕНИЕ

Семантика электронного учебника (ЭУ)

Таблица 1 Состав тем и разделов

№	Тема	№	Раздел
1	Роль ИС в управлении современной экономикой и бизнесом	1	Сущность и структура системы управления
		2	Информационная система: понятие, место в системе управления, функции
		3	Информация как предмет и продукт функционирования ИС
2	Понятие автоматизированной ИС (АИС), этапы развития, принципы построения.	4	Понятие автоматизированной ИС (АИС), этапы развития, принципы построения.
3	Состав и структура ИС	5	Структура ИС
		6	Функциональные компоненты АИС
		7	Компоненты системы обработки данных (СОД)
		8	Организационные компоненты АИС
4	Классификация ИС	9	Классификация ИС
5	Тенденции в развитии ИС в соответствии с потребностями современного бизнеса.	10	Развитие подходов к технической и программной реализации ИС
		11	Развитие телекоммуникационных средств передачи данных

		12	Развитие информационных технологий в экономике
6	Обзор возможностей современных ИС для конкретных бизнес-процессов	13	Бухгалтерские ИС
		14	ИС управление бизнес-проектами
		15	Банковские ИС
		16	Электронная коммерция

Таблица 2 Семантическая сеть ЭУ

№	Новые понятия	№ раздела	№ базового понятия	Входы в узловые понятия
1	сложная система	1	7	
2	управление	1	7	
3	структура управления	1	7	
4	уровень управления	1	7	
5	прямая связь	1	7	
6	обратная связь	1	7	
7	информационная система	2	16	7
8	информация,	3	7	6
9	данные	3	8	
10	синтаксический уровень информации	3	8	
11	семантический уровень информации	3	8	
12	прагматический уровень информации	3	8	
13	количественная мера информации	3	8	
14	тезаурус	3	8	
15	информационная технология (ИТ)	4	16	10
16	автоматизированная информационная система (АИС)	4	46	5
17	автоматизированное рабочее место (АРМ)	4	16	
18	программное обеспечение (ПО) СОД	9	20	
19	функциональные компоненты АИС	6	16	3
20	компоненты системы обработки данных (СОД)	6	16	5
21	организационные компоненты АИС	6	16	3

22	функциональная структура АИС	8	19	
23	электронный документооборот	8	19	
24	функциональная задача	8	19	
25	информационное обеспечение (ИО) СОД	9	20	
26	правовое обеспечение СОД	9	20	
27	лингвистическое обеспечение СОД	9	20	
28	техническое обеспечение СОД	9	20	
29	одиночные ИС	11	21	
30	групповые ИС	11	21	
31	корпоративные ИС (КИС)	11	21	
32	системы оперативной обработки данных — OLTP	11	15	
33	системы поддержки и принятия решений — DSS	11	15	
34	хранилище данных	11	15	
35	система обработки транзакций	11	15	
36	профессиональные системы	11	15	
37	информационно-справочные системы – ИСС	11	15	
38	офисные АИС	11	15	
39	мультисервисные сети	13	15	
40	системы поддержки исполнения решений (EPSS)	14	15	
41	МА-технологии	12	15	
42	MRP-системы	14	46	
43	MRP II-системы	14	46	
44	ERP-системы	14	46	
45	CSRP-системы	14	46	
46	экономическая АИС	2	19	9
47	компьютерная система бухгалтерского учета (КСБУ)	15	46	
48	управление проектами	16	46	
49	автоматизированная банковская система (АБС)	17	46	
50	электронная коммерция	18	46	
51	электронный магазин	18	46	

Таблица 3 Глоссарий ЭУ

№	Понятие	Определение нового понятия
1	сложная система	система, которая состоит из большого числа взаимосвязанных и взаимодействующих между собой элементов и способна выполнять сложную функцию.
2	управление	организирующее начало в сложной системе, которое образует систему управления
3	структура управления	совокупность объекта управления, и субъекта управления – управленческого аппарата
4	уровень управления	это отражение иерархии в процессе управления, выделяют три основных уровня: высший (стратегический), средний (тактический) и операционный.
5	прямая связь	выражается потоком директивной информации, направляемой от управленческого аппарата к объекту управления
6	обратная связь	представляет собой поток отчетной информации о выполнении принятых решений, направляемый от объекта управления к управленческому аппарату
7	информационная система (ИС)	коммуникационная система по сбору, передаче, переработки данных об объекте, снабжающая работников различного ранга информацией для реализации функции управления
8	информация,	совокупность полезных сведений являющихся объектом сбора, регистрации, хранения, передачи и преобразования.
9	данные	Понятие «данные» стало применимым в концепции баз данных и означает набор конкретных значений, параметров, характеризующих объект, условие, ситуацию или любые другие факторы
10	синтаксический уровень информации	связан с внешней формой и структурой информационных сообщений.
11	семантический уровень информации	определяет смысловое содержание информации.
12	прагматический уровень информации	отражает ценность информации для системы управления, ее полезность для выработки управленческих решений.
13	количественная мера информации	отражает ее объем на каждом из уровней ее структуризации (синтаксическом, семантическом и прагматическом)

14	тезаурус	содержит взаимосвязанные понятия, термины, определения, согласованные структуры данных логического уровня представления, т.е. «знания» о системе
15	информационная технология (ИТ)	это совокупность методов и средств обработки данных, которая обеспечивает целенаправленное преобразование данных в соответствии с закономерностями предметной области
16	автоматизированная информационная система (АИС)	реализует ИТ в сфере управления при совместной работе управленческого персонала и комплекса технических средств
17	автоматизированное рабочее место (АРМ)	проблемно-ориентированная диалоговая человеко-машинная система, программно-технические средства которой поддерживают деятельность пользователя при выполнении его профессиональных функций.
18	программное обеспечение (ПО) СОД	совокупность программ СОД и программных документов, необходимых для эксплуатации этих программ.
19	функциональные компоненты АИС	модели, методы и алгоритмы получения управляющей информации.
20	компоненты системы обработки данных (СОД)	это системы, реализующие типовые операции обработки данных
21	организационные компоненты АИС	совокупность методов и средств, реализующих организационную структуру объектов и управленческие функции в условиях функционирования СОД
22	функциональная структура АИС	совокупность функциональных подсистем, комплексов задач и процедур обработки информации, реализующих функции системы управления.
23	электронный документооборот	система управления электронными документами и организации документооборота АИС,

24	функциональная задача	это законченный комплекс обработки информации, обеспечивающий либо выдачу прямых управляющих воздействий, либо выдачу необходимой информации для принятия решений управленческим персоналом.
25	информационное обеспечение (ИО) СОД	это совокупность форм документов, нормативной базы и баз данных и другой информации, которая используется в информационной системе при ее функционировании
26	правовое обеспечение СОД	это совокупность норм, выраженных в нормативных актах, устанавливающих и закрепляющих организацию СОД, ее цели, задачи, структуру, функции и правовой статус
27	лингвистическое обеспечение СОД	совокупность языковых средств, используемых на различных стадиях создания и эксплуатации ИС для обеспечения общения человека и ЭВМ.
28	техническое обеспечение СОД	совокупность вычислительной техники, средств коммуникации и оргтехники, обеспечивающей автоматизированную обработку данных.
29	одиночные ИС	системы, которые реализуются на автономном персональном компьютере (сеть не используется).
30	групповые ИС	ориентированы на коллективное использование информации членами рабочей группы и чаще всего строятся на базе локальной вычислительной сети.
31	корпоративные ИС (КИС)	ориентированы на крупные компании и могут поддерживать территориально разнесенные узлы или сети. Они интегрируют все функции управления на оперативном, тактическом и стратегическом уровнях.
32	системы оперативной обработки данных — OLTP	это традиционные ИС учета и регистрации первичной информации
33	системы поддержки и принятия решений — DSS	ориентированы на реализацию сложных бизнес-процессов, требующих аналитической обработки информации, формирования новых знаний. Они основаны на технологии OLAP и решают задачи, которые плохо структурированы

34	хранилище данных	предметно-ориентированное, привязанное ко времени и неизменяемое собрание данных для поддержки процесса принятия управляющих решений. Оно является основой построения OLAP-технологий
35	система обработки транзакций	являются базовыми для обслуживания текущих операций предприятия. Они используют технологию OLTP и представляют собой системы, которые выполняют и регистрируют рутинные регулярные транзакции
36	профессиональные системы	обслуживают конструкторов, научных работников, технологов
37	информационно-справочные системы – ИСС	системы, основанные на гипертекстовых технологиях и средствах мультимедиа. Они хранят данные (текстовые, графические и другие документы), организуют их поиск и выдачу по запросам пользователей
38	офисные АИС	используются для повышения эффективности работы с данными, они обеспечивают связи с потребителями, поставщиками и внешними организациями.
39	мультисервисные сети	телекоммуникационная структура, которая позволяет оказывать пользователям услуги связи, различающиеся как по качественным, так и по количественным параметрам.
40	системы поддержки исполнения решений (EPSS)	система, обеспечивающая не только поиск нужного решения, но и внедрение принятого решения в практику управления
41	МА-технологии	являются объединением объектно-ориентированной технологии и методов искусственного интеллекта.
42	MRP-системы	MRP - системы(Material Requirements Planning) - это планирование потребностей в материалах
43	MRP II-системы	MRP II - системы(Manufacturing Resource Planning) это планирование производственных ресурсов

44	ERP-системы	ERP - системы(Enterprise Resource Planning ERP (Enterprise Resource Planning - планирование ресурсов предприятия
45	CSRP-системы	CSRP -системы (Customer Synchronized Resource Planning) - планирование ресурсов, синхронизированное с потребностями клиентов
46	экономическая АИС	система, которая реализует ИТ в сфере управления конкретным бизнес-процессом
47	компьютерная система бухгалтерского учета (КСБУ)	технологическая компьютерная систем, которая обеспечивает функции и задачи ведения бухгалтерского учета
48	управление проектами	процесс планирования, организации и управления задачами и ресурсами с целью достижения определенной уникальной цели
49	автоматизированная банковская система (АБС)	ехнологическая система, которая обеспечивает функционирование банковского учреждения.
50	электронная коммерция	использование Internet для проведения коммерческих операций между предприятиями или между предприятием и потребителями.
51	электронный магазин	место (адрес) в Интернет, в котором Вы рекламируете и продаете товары или услуги другим пользователям сети по всему миру

Навчально-методичне видання

ЦИГАНКОВА Світлана Олексіївна
СКВІРСЬКИЙ Віктор Давидович

КИБЕРНЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

*Навчально-методичний посібник
для магістрантів спеціальності
8.04030201 – «Інформатика»*

Навчально-методичний посібник орієнтовано на використання магістрантами спеціальності «Інформатика». Призначення посібника – придбати системне уявлення про джерела процесів функціонування та управління сучасних інформаційних технологій (ІТ) і оцінити ступінь їх зв'язку з методами кібернетики на прикладі використання конкретних ІТ у проектуванні інформаційних систем і електронного підручника

За редакцією авторів

Здано до склад. 04.04.2012 р. Підп. до друку 04.05.2012 р.
Формат 60x84 1/16. Папір офсет. Гарнитура Times New Roman.
Друк ризографічний. Ум. друк. арк.. 10,87. Наклад 300 прим. Зам. № 136.

Видавець і виготовлювач

Видавництво Державного закладу
«Луганський національний університет імені Тараса Шевченка»
вул.. Оборонна, 2, м. Луганськ, 91011, т/ф: (0642)58-03-20.
e-mail: alma-mater@list.ru
Свідоцтво суб'єкта видавничої справи ДК № 3459 від 09.04.2009 р.