

Н.В. БЕЗМЕНОВА, С.А.ШУСТОВ

**МЕТОДОЛОГИЯ
КОНЦЕПТУАЛЬНОГО
МОДЕЛИРОВАНИЯ IDEF0
В ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ
BPWin**

**(на примере жизненного цикла
двигателя летательного аппарата)**

2006



САМАРА

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»

Н.В. БЕЗМЕНОВА, С.А. ШУСТОВ

МЕТОДОЛОГИЯ КОНЦЕПТУАЛЬНОГО
МОДЕЛИРОВАНИЯ IDEF0
В ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ VPWin
(на примере жизненного цикла двигателя
летательного аппарата)

*Утверждено Редакционно-издательским советом университета
в качестве учебного пособия*

САМАРА
Издательство СГАУ
2006

УДК 004.4 (075)

ББК 32.973

Б 398



**Инновационная образовательная программа
"Развитие центра компетенции и подготовка
специалистов мирового уровня в области аэро-
космических и геоинформационных технологий"**

Рецензенты: д-р техн. наук, проф. Н. Д. П р о н и ч е в

Ген. директор ЗАО «Консалтинг Аудит» О.Н. Б а р с у к о в а

Безменова Н.В.

Б 398 **Методология концептуального моделирования IDEF0 в инструментальной среде BPWin (на примере жизненного цикла двигателя летательного аппарата): учеб. пособие / Н.В. Безменова, С.А. Шустов. – Самара: Изд-во Самар. гос. аэрокосм. ун-та, 2006. – 64 с. : ил.**

ISBN 5-7883-0474-1

Рассмотрены особенности системного моделирования сложных объектов. Дано описание различных методик функционального моделирования из серии IDEF. Описана одна их методик, реализованная в виде Государственного стандарта IDEF0. Изложено построение модели на примере проектирования двигателей летательных аппаратов в программной среде BPWin, описан интерфейс программы и основные приемы работы.

Учебное пособие предназначено для студентов дневного отделения, обучающихся по специальностям 130206 (авиационные двигатели и энергетические установки, специализация «Организация производства»), 130409 (ракетные двигатели, специализация «Промышленная экология»), а также для студентов дневного и очно-заочного отделений, обучающихся по специальности 060800 (экономика и управление на предприятии, машиностроительная отрасль)

УДК 004.4 (075)

ББК 32.973

ISBN 5-7883-0474-1

© Безменова Н.В., Шустов С.А., 2006

© Самарский государственный
аэрокосмический университет, 2006

ОГЛАВЛЕНИЕ

Введение	5
1 Основные понятия о технологии системного моделирования	7
1.1 Содержание этапов системного моделирования	7
1.2 Понятие о концептуальном моделировании	8
1.3 Обзор методик концептуального моделирования	10
1.4 Основные понятия о CASE-средствах системного моделирования	12
Вопросы для самоконтроля	13
2 Основные понятия о методологии IDEF0	14
2.1 Предпосылки создания SADT и IDEF0	14
2.2 Основные принципы SADT-моделирования	18
2.2.1 Цель в SADT-моделировании	19
2.2.2 Объект моделирования и точка зрения автора модели	20
2.2.3 SADT- модель - набор взаимосвязанных диаграмм	22
2.3 Синтаксис SADT-диаграмм	27
2.3.1 Блоки и дуги SADT – диаграмм	28
2.3.2 Размещение блоков на SADT-диаграммах	29
2.3.3 Дуги, их назначение и классификация	30
2.3.4 Правила разветвления и слияния дуг	34
2.4 Синтаксис SADT- модели	36
2.4.1 Контекстная диаграмма и диаграммы декомпозиции SADT-модели	37
2.4.2 Декомпозиция контекстной диаграммы и правило нумерации узлов	38

Вопросы для самоконтроля	39
3. CASE-средство BPWin для поддержки методологии	
IDEF0	41
3.1 Общие сведения о CASE-средстве BPWin	41
3.2 Запуск программы BPWin	42
3.3 Формирование контекстной диаграммы	43
3.4 Описание модели средствами BPWin	46
3.5 Построение диаграмм декомпозиции	50
3.6 Сохранение результатов и выход из BPWin	54
Вопросы для самоконтроля	55
Заключение	56
Список литературы	59
ПРИЛОЖЕНИЕ А Словарь терминов	61

ВВЕДЕНИЕ

Эффективность управления предприятием аэрокосмической отрасли и его конкурентоспособность в современных условиях определяются качеством бизнес-процессов [1].

В мировой практике требования к качеству бизнес-процессов предприятий сформулированы в виде стандарта качества ISO 9002. В последних версиях стандарта особое внимание уделяется анализу, контролю и постоянному улучшению бизнес-процессов в соответствии с изменяющимися внешними условиями и целями предприятия.

Основным инструментом анализа и улучшения бизнес-процессов в стандарте ISO серии 9000 является функциональное моделирование.

Кроме того, методы функционального моделирования являются неотъемлемой частью разработки современного сложного программного обеспечения, находясь, таким образом, на стыке управленческих и информационных технологий.

Основные методы моделирования данных, функций и процессов объединены в группу стандартов IDEF, один из которых (IDEF0) в настоящее время является Государственным стандартом Российской Федерации.

Данный метод IDEF0 – метод функционального моделирования или метод моделирования бизнес-процессов.

IDEF0 является основным инструментом по внедрению стандарта качества. Для этого на предприятии строится модель бизнес-процессов «как есть», проводится ее анализ, выявляются узкие места и проблемы, разрабатывается модель идеальных бизнес-процессов «как должно быть». Затем проводятся организа-

ционные изменения в бизнес-процессах предприятия. Согласно современным требованиям к системе менеджмента такая работа должна проводиться на предприятиях постоянно.

Стандарт IDEF0 представляет собой методологию структурного анализа процессов в любых сложных системах: экономических, информационных, интеллектуальных, социальных и др. В данном учебном пособии рассматриваются история создания методики IDEF0, ее содержание на примере моделирования процесса создания конкурентоспособного ДЛА, а также CASE-средства программной поддержки процесса моделирования.

Одним из производителей CASE-средств является компания Platinum technologies. Этой компанией разработаны такие программные продукты как BPWin (для функционального моделирования в стандартах IDEF0, DFD, IDEF3 Workflow), ERWin (для проектирования баз данных в стандарте ERD).

В пособии подробно рассмотрены принципы работы с программой BPWin: создание модели, описание свойств модели, создание и редактирование контекстной диаграммы и диаграмм декомпозиции, рисование блоков и стрелок.

В конце каждого раздела приводятся вопросы для самоконтроля.

Методика IDEF0 прошла апробацию в учебном процессе СГАУ при решении практических вопросов повышения эффективности управления предприятием [2,3].

1 ОСНОВНЫЕ ПОНЯТИЯ О ТЕХНОЛОГИИ СИСТЕМНОГО МОДЕЛИРОВАНИЯ

Технология системного моделирования включает в себя принципы рациональной организации процесса моделирования и инструментарий, обеспечивающий их практическую реализацию.

Рациональная организация процесса системного моделирования основана на принципе моделирования «сверху вниз» и состоит из взаимосвязанных этапов – анализа, проектирования, реализации и эксплуатации.

Инструментарий технологии системного моделирования включает в себя методики системного моделирования и средства их компьютерной поддержки, которые получили название CASE-средства (Computer Aided System Engeneering – компьютерная поддержка проектирования систем) [4].

1.1 Содержание этапов системного моделирования

Рассмотрим содержание этапов системного моделирования на примере разработки модели жизненного цикла вновь создаваемого ДЛА. Цель создания этой модели – определение экономической целесообразности создания нового ДЛА на возможно более ранней стадии, чтобы избежать нерациональных расходов.

На этапе анализа определяются требования к создаваемой модели. Для этого, используя данные об объекте моделирования и внешней среде, определяются границы предметной области, строятся модели внешней среды и предметной области (в рассматриваемом примере это жизненный цикл создания ДЛА). В процессе формирования модели предметной области выделяются объекты моделирования, их свойства и связи, которые будут существенны для будущих пользователей модели. Изучаются информационные потребности будущих пользователей создаваемой модели, которые также должны учитываться.

На этапе проектирования разрабатываются детальные

модели всех объектов, включенных в предметную область на этапе анализа. Информационная модель на этапе проектирования обычно имеет модульную форму, а процесс ее построения включает в себя:

- уточнение базы данных для последующей генерации SQL-предложений на базе определенной СУБД;
- уточнение структуры пользовательского интерфейса;
- построение структурных схем, отражающих логику работы пользовательского интерфейса и модель бизнес-логики.

На этапе программной реализации информационной модели осуществляется генерация программного кода, отладка и тестирование компьютерной формы разработанной модели.

На этапе эксплуатации разработанная информационная модель используется для решения тех проблем, ради которых она создавалась.

1.2 Понятие о концептуальном моделировании

Этап анализа в разных прикладных дисциплинах (САПР, проектирование информационных систем, программирование и др.) называется также внешним проектированием, макро-моделированием и т.д. В последнее время для этого этапа широкое распространение получил термин «концептуальное моделирование», который и будет использоваться в дальнейшем, а модели предметной области и внешней среды, полученные на этом этапе, будем называть концептуальными моделями.

Концептуальная модель должна давать представление о всех объектах предметной области и их связях с внешней средой, но без подробной детализации. Поэтому эта модель называется также архитектурной моделью.

Этап концептуального моделирования является весьма ответственным, поскольку именно на этом этапе определяется состав тех свойств объекта, которые будут описываться будущей моделью [4].

От того, насколько удачным будет отбор моделируемых свойств, будет зависеть эффективность создаваемой модели. Например, практика создания таких сложных объектов, как современный авиационный или ракетный двигатель, показывает, что если были допущены ошибки на стадии концептуального моделирования, их устранение на стадии проектирования обходится дороже в 10 раз, на стадии реализации – в 100 раз, а на стадии эксплуатации – в 1000 раз. Поэтому для современного этапа создания ДЛА характерной чертой является рост требований к качеству моделей, используемых на этапе анализа.

В соответствии с методологией системного подхода содержание концептуальной модели будет определяться целями моделирования. В рассматриваемом примере целью моделирования является надежная оценка экономической эффективности проекта, поэтому объектами моделирования должны быть все стадии жизненного цикла, связанные как с затратами, так и с получением прибыли. Применительно к авиационному двигателю прибыль получается на этапе его эксплуатации, а затраты возникают на всех этапах: проектирования, производства, эксплуатации, модернизации и утилизации.

Поэтому объектами моделирования в данном случае должны быть:

- во-первых, этапы жизненного цикла: проектирование, производство, эксплуатация, модернизация, утилизация;
- во-вторых, связи между этими этапами жизненного цикла;
- в-третьих, модель внешней среды, которая в условиях рыночной экономики должна включать в себя потребителей продукции, источники ресурсов, конкурентов, субъектов власти на государственном и муниципальном уровнях.

Вследствие очевидной сложности рассматриваемой модели, конечная ее форма, ориентированная на пользователей, может быть

только информационной. Представление модели системы в информационной форме является характерной чертой технологии системного моделирования.

Применительно к концептуальному моделированию это означает необходимость представления всех моделируемых объектов предметной области и связей между ними в информационной форме, т.е. разработки так называемой логической информационной модели предметной области. Эта логическая информационная модель представляет собой описание на логическом уровне структуры данных, потоков данных и процессов их обработки.

Формированием логической информационной модели завершается этап концептуального моделирования.

1.3 Обзор методик концептуального моделирования

Назначение и содержание методик концептуального моделирования рассмотрим на примере IDEF-методологии, получившей широкое распространение на практике [6,7].

IDEF-методология создавалась в рамках предложенной ВВС США программы компьютеризации промышленности ICAM. Во время реализации этой программы выявилась потребность в разработке методов анализа процессов взаимодействия в производственных (промышленных) системах. Принципиальным требованием была возможность эффективного обмена информацией между всеми специалистами – участниками программы. В результате в военном ведомстве США был разработан такой метод, поддержанный NIST (Национальный институт стандартов и технологий) и IEEE (Институт инженеров по электротехнике и электронике) и принятый в качестве федерального стандарта обработки информации. Этот метод получил название IDEF (Integrated Definition language) – интегрированный язык определений.

В настоящее время IDEF принят в НАТО как стандарт

моделирования процессов и данных, используется как наиболее эффективный международный язык передачи данных в 16 государствах – членах НАТО, а также в аэрокосмической и оборонной промышленности, включен в число CALS-стандартов.

С недавнего времени стандарты серии IDEF приняты и в Российской Федерации.

На основе IDEF разработан ряд методологий, перечень которых приведен ниже:

1) **IDEF0** (Function Modeling) – методология функционального моделирования (поддерживается CASE-средством BPWin);

2) **IDEF1** (Information Modeling) – методология информационного моделирования на основе ER-диаграмм (диаграмм сущность-связь);

3) **IDEF1X** (Data Modeling) – модификация методологии IDEF1 в целях автоматизации процесса моделирования (поддерживается CASE-средством ERWin);

4) **IDEF2** (Simulation Modeling) – методология имитационного моделирования (поддерживается CASE-средством Silverrun);

5) **IDEF3** (Process Description Capture) – методология процессного моделирования workflow (поток работ);

6) **IDEF4** (Object-oriented Design) – методология объектно-ориентированного моделирования (поддерживается CASE-средством Rational Rouse);

7) **IDEF5** (Ontology Description Design) – методология моделирования метасущностей (метамоделирование);

8) **IDEF6** (Design Rationale Capture) – методология логического моделирования;

9) **IDEF7** (Information System Audit Method) – методология аудита информационных систем;

10) **IDEF8** (User Interface Modeling) – методология моделирования интерфейса пользователей;

11) **IDEF9** (Scenario-Driven Info Sys Design Spec) – методология моделирования оптимального управления сценариями информационных систем;

12) **IDEF10** (Implementation Architecture Modeling) – методология архитектурного (структурного) моделирования;

13) **IDEF11** (Information Artifact Modeling) – методология информационного моделирование артефактов (структур сущностей);

14) **IDEF12** (Organization Modeling) – методология организационного моделирования;

15) **IDEF13** (Tree Schema Mapping Design) – методология 3D-моделирования схем (чертежей, планов, карт);

16) **IDEF14** (Network Design) – методология сетевого моделирования.

1.4 Основные понятия о CASE-средствах системного моделирования

Для практического использования методы моделирования нуждаются в программной поддержке. За последние несколько десятилетий сформировалось это новое направление развития прикладного программного обеспечения и получило название - CASE-средства (Computer Aided Systems/Software Engineering – компьютерные средства проектирования систем) [6,7,8].

CASE-средства представляют собой комплекс взаимосвязанных средств автоматизации процессов системного моделирования и служат инструментарием для поддержки разработчика информационных систем на этапах концептуального моделирования и проектирования.

Для CASE-средств характерны следующие черты:

- мощная графика для описания и документирования результатов концептуального моделирования и проектирования, а также для улучшения интерфейса с пользователем, развивающая творчес-

кие возможности специалистов и не отвлекающая их от процесса проектирования на решение второстепенных вопросов;

- интеграция, обеспечивающая легкость передачи данных между средствами и позволяющая управлять всем процессом проектирования и разработки непосредственно через процесс планирования проекта;
- использование компьютерного хранилища (репозитория) для всей информации о проекте, который может разделяться между разработчиками и исполнителями как основа для автоматического формирования программного обеспечения и повторного использования в будущих системах.

Детальное изложение технологии использования CASE-средств на примере программы для моделирования процессов VPWin приводится в главе 3 данного пособия.

Вопросы для самоконтроля

1. Какие основные этапы включает процесс системного моделирования сложных технических объектов?
2. Сформулируйте основное содержание этапов анализа и проектирования?
3. Что называют концептуальным моделированием сложных систем?
4. Почему этап концептуального моделирования имеет важное значение при создании сложных технических объектов?
5. Какие этапы жизненного цикла ДЛА влияют на экономическую эффективность процесса его создания?
6. Что называют стандартами IDEF и откуда они произошли?
7. Что называют CASE-средствами и для чего они нужны?

2 ОСНОВНЫЕ ПОНЯТИЯ О МЕТОДОЛОГИИ IDEF0

2.1 Предпосылки создания SADT и IDEF0

Методология SADT возникла в конце 60-х годов в ходе революции, вызванной структурным программированием [9]. Когда большинство специалистов билось над созданием программного обеспечения, немногие старались разрешить более сложную задачу создания крупномасштабных систем, включающих и людей, и машины: в телефонной связи, промышленности, управлении и контроле за вооружением. Эти первоначально немногочисленные специалисты, традиционно занимавшиеся созданием крупномасштабных систем, первыми осознали необходимость большей упорядоченности всего процесса создания сложных систем. Таким образом, разработчики начали формализовать процесс создания системы, разбивая его на следующие фазы:

- анализ – определение того, что система должна делать;
- проектирование – определение подсистем и характеристик их взаимодействия;
- реализация – разработка подсистем по отдельности, затем объединение подсистем в единое целое;
- тестирование – проверка работы разработанной системы;
- эксплуатация – использование разработанной системы для решения прикладных проблем.

Эта последовательность всегда выполнялась итерационно, потому что система полностью никогда не удовлетворяла требованиям пользователей, которые часто менялись в процессе разработки. И, тем не менее, постоянно возникали сложности с устранением ошибок и замечаний к системе. Эксплуатационные расходы, возникавшие после сдачи системы, стали существенно превышать расходы на ее создание и продолжали расти с огромной

скоростью из-за низкого качества исходно созданных программ. Некоторые считали, что рост эксплуатационных расходов обусловлен характером ошибок, допущенных в процессе создания системы. Исследования показали, что большой процент ошибок в системе возник в процессе анализа и проектирования, гораздо меньше их было допущено при реализации и тестировании, а цена (временная и денежная) обнаружения и исправления ошибок становилась выше на более поздних стадиях проекта (см. раздел 1.2). Традиционные подходы к созданию систем по мере роста их сложности приводили к возникновению многих проблем. Не было единого подхода. Привлечение пользователя к процессу разработки не контролировалось. Проверка на согласованность проводилась нерегулярно или вообще отсутствовала. Результаты одного этапа не согласовывались с результатами других. Процесс с трудом поддавался оценкам как качественным, так и количественным. Утверждалось, что когда создатели систем пользуются методологиями типа структурного программирования и проектирования сверху вниз, они решают либо не поставленные задачи, либо плохо поставленные, либо хорошо поставленные, но неправильно понятые задачи.

Кроме того, ошибки в создании систем становились все менее доступны выявлению с помощью аппаратных средств или программного обеспечения, а наиболее катастрофические ошибки допускались на ранних этапах создания системы. Часто эти ошибки были следствием неполноты функциональных спецификаций или несогласованности между спецификациями и результатами проектирования. Проектировщики знали, что сложность систем возрастает и что определены они часто весьма слабо. Рост объема и сложности систем является жизненной реальией. Эту предпосылку нужно было принять как неизбежную. Но ошибочное определение системы не является неизбежным: оно - результат неадекватности методов создания систем.

Вскоре был выдвинут тезис: *совершенствование методов анализа есть ключ к созданию систем, эффективных по стоимости, производительности и надежности*. Для решения ключевых проблем традиционного создания систем широкого профиля требовались новые методы, специально предназначенные для использования на ранних стадиях процесса.

Создание SADT-методологии проистекало из этого убеждения. SADT – аббревиатура слов Structured Analysis and Design Technique (Технология структурного анализа и проектирования). Методы, подобные SADT, на начальных этапах создания системы позволяли гораздо лучше понять рассматриваемую проблему, что сокращало затраты как на создание, так и на эксплуатацию системы, а кроме того, повышало ее надежность. SADT - это способ уменьшить количество дорогостоящих ошибок за счет структуризации на ранних этапах создания системы, улучшения контактов между пользователями и разработчиками и сглаживания перехода от анализа к проектированию.

Создание SADT-методологии и сам термин SADT связаны с именем Дугласа Т. Росса, который в конце шестидесятых годов занимался разработкой программного обеспечения первых САПР в МТИ (Массачусетский технологический институт, США). Дуглас Т. Росс часть своих PLEX-теорий, относящихся к методологии и языку описания систем, назвал «Методология структурного анализа и проектирования» (SADT). Непосредственная работа над SADT- методологией началась в 1969 году, когда Дуглас Т. Росс основал компанию SoftTech Inc для продвижения SADT-технологии на рынок.

Первое крупное практическое приложение SADT-технологии было реализовано в 1973 г. при разработке большого аэрокосмического проекта, когда она была несколько пересмотрена сотрудниками SofTech Inc. В 1974 г. SADT была еще улучшена и передана одной из крупнейших европейских телефонных

компаний. Появление SADT на рынке произошло в 1975 г. после годовичного оформления в виде коммерческого продукта. К 1981 г. SADT уже использовали более чем в 50 компаниях при работе более чем над 200 проектами, включавшими более 2000 людей и охватывавшими дюжину проблемных областей, в том числе телефонные сети, аэрокосмическое производство, управление и контроль, учет материально-технических ресурсов и обработку данных.

Широкий спектр областей указывает на универсальность и мощь методологии SADT. В программе интегрированной компьютеризации производства (ICAM) Министерства обороны США была признана полезность SADT, что привело к стандартизации и публикации ее части, называемой IDEF0. Сегодня методология SADT получила широкое распространение в европейской, южно-азиатской и американской аэрокосмической промышленности (под названием IDEF0). Под этим названием SADT применялась тысячами специалистов в военных и гражданских промышленных организациях.

В коммерческом мире *SADT-методология используется для определения требований к создаваемой системе на этапе концептуального моделирования.* В этом качестве она конкурирует с методами, ориентированными на потоки данных, - структурного проектирования Е.Иордана, структурного анализа Т.ДеМарко, структурного системного анализа С. Гейна и Т. Сарсона, а также с методами структуризации данных М.Джексона, Лж.Д. Варнира и К. Орра. В отличие от этих методов структурного анализа, истоки которых нужно искать в проектировании программного обеспечения, SADT создана для описания произвольной системы и ее среды до определения требований к программному обеспечению или к чему-либо другому. Иными словами, *поставив своей целью описание системы в целом, создатели SADT-методологии*

изобрели графический язык и набор процедур анализа для понимания системы прежде, чем можно представить себе ее воплощение.

В настоящее время SADT-методология описания сложных систем признана международным стандартом и включена в число CALS-стандартов.

Почему SADT имеет такое широкое применение? Во-первых, SADT является единственной методологией, легко отражающей такие системные характеристики, как управление, обратная связь и исполнители. Это объясняется тем, что SADT изначально возникла на базе проектирования систем более общего вида в отличие от других структурных методов, «выросших» из проектирования программного обеспечения. Во-вторых, SADT в дополнение к существовавшим в то время концепциям и стандартам для создания систем имела развитые процедуры поддержки коллективной работы и обладала преимуществом, связанным с ее применением на ранних стадиях создания системы. Кроме того, широкое использование SADT показало, что ее можно сочетать с другими структурными методами. Это достигается использованием графических SADT-описаний в качестве схем, связывающих воедино различные методы, примененные для описания определенных частей системы с различным уровнем детализации. Таким образом, неадекватные спецификации систем того времени вызвали создание графического языка SADT, а его усиленное использование преобразовало SADT в законченную методологию, способную повысить качество продуктов, создаваемых на ранних стадиях развития проекта.

2.2 Основные принципы SADT-моделирования

Описание системы с помощью SADT называется *SADT-моделью*. В SADT-моделях используются как естественный, так и графический языки. Для передачи информации о конкретной

системе источником естественного языка служат люди, описывающие систему, а источником графического языка – сама методология SADT. Графический язык SADT обеспечивает структуру и точную семантику формализованному языку модели. Графический язык SADT организует естественный язык вполне определенным и однозначным образом, за счет чего SADT и позволяет описывать системы, которые до недавнего времени не поддавались адекватному представлению.

С точки зрения разработчика, SADT-модель может быть сосредоточена либо на функциях системы, либо на ее объектах. SADT-модели, ориентированные на функции, принято называть *функциональными моделями*, а модели, ориентированные на объекты системы, – *моделями данных*.

Функциональная модель представляет с требуемой степенью детализации систему функций, которые, в свою очередь, отражают свои взаимоотношения через объекты системы. Версия SADT-модели, реализованная в стандарте IDEF0, поддерживает только функциональное моделирование.

2.2.1 Цель в SADT-моделировании

SADT-модель дает полное, точное и адекватное описание системы, имеющее конкретное назначение. Это назначение, называемое целью модели, вытекает из формального определения модели в SADT:

М есть модель системы *S*, если *M* может быть использована для получения ответов на вопросы относительно *S* с точностью *A*.

Таким образом, целью модели является получение ответов на некоторую совокупность вопросов. Эти вопросы неявно присутствуют (подразумеваются) в процессе анализа и, следовательно, они руководят созданием модели и направляют его. Это означает, что сама модель должна будет дать ответы на эти вопросы с заданной степенью точности. Если модель отвечает не на все

вопросы или ее ответы недостаточно точны, то мы говорим, что модель не достигла своей цели. Определяя модель таким образом, SADT закладывает основы практического моделирования.

Смысл и трактовка этого определения оказали существенное влияние на практическое применение SADT. Обычно вопросы для SADT-модели формулируются на самом раннем этапе проектирования, при этом основная суть этих вопросов должна быть выражена в одной-двух фразах. Далее этот список вопросов сводится в одно предложение. Это предложение становится целью модели, а список вопросов сохраняется как детализация этого предложения.

После завершения работы над моделью информация, содержащаяся в модели, должна обеспечивать ответы на поставленные вопросы. Именно способность разрабатываемой модели ответить на перечень вопросов, связанных с формулировкой цели моделирования, является критерием завершенности модели. Поэтому только по степени полноты ответов на поставленные вопросы можно определить, когда процесс моделирования можно считать завершенным (т.е. когда модель будет соответствовать поставленной цели).

2.2.2 Объект моделирования и точка зрения автора модели

Модель всегда является определенным толкованием (представлением) системы как образа некоторого реального объекта. Однако моделируемая система никогда не существует изолированно, она всегда связана с окружающей средой. Причем зачастую трудно определить, где кончается система и начинается внешняя среда. По этой причине в методологии SADT подчеркивается необходимость точного определения границ системы. SADT-модель всегда ограничивает свой объект, т.е. модель должна точно устанавливать, что является и что не является объектом моделирования. Это достигается за счет однозначного указания границ моделируемого объекта, описывая то, что входит в систему,

а что лежит за ее пределами. Этим самым SADT-модель помогает сконцентрировать внимание именно на описываемой системе и позволяет избежать включения посторонних объектов. Вот почему SADT-модель должна иметь единственный объект.

С определением модели тесно связана позиция, с которой наблюдается система и создается ее модель. Поскольку качество описания системы резко снижается, если оно не сфокусировано на чем-то определенном, SADT требует, чтобы модель рассматривалась все время с одной и той же позиции. Эта позиция называется «точкой зрения» данной модели и должна быть четко зафиксирована автором модели. «Точку зрения» лучше всего представлять себе как позицию автора модели, с которой лучше всего увидеть систему в действии. С этой фиксированной точки зрения можно создать согласованное описание системы так, чтобы модель не дрейфовала вокруг да около, и в ней не смешивались бы несвязанные описания. Точка зрения определяется целью моделирования.

Например, при создании модели жизненного цикла авиационного двигателя можно встать на точку зрения главного конструктора, главного технолога, финансового директора, директора завода и т.д. В зависимости от точки зрения модель жизненного цикла будет меняться из-за различия в акцентах. Очевидно, что в условиях рыночной экономики наиболее полно модель жизненного цикла будет выражаться с позиции человека, ответственного за финансовый результат, связанный с реализацией проекта создания ДЛА [5]. В зависимости от организационной структуры таким человеком является президент, вице-президент, директор или менеджер, ответственный за данный проект.

В дальнейшем лицо, ответственное за реализацию проекта, в том числе и за его финансовый результат, будем называть главным менеджером проекта.

Именно с точки зрения главного менеджера проекта и должна рассматриваться модель жизненного цикла ДЛА в соответствии с целью моделирования – оценить степень экономической целесообразности проекта.

С этой точки зрения очевидными становятся и вопросы, на которые должна ответить модель – какова структура доходов и расходов проекта на различных стадиях жизненного цикла, каковы должны быть рациональные сроки той или иной стадии жизненного цикла, какой должна быть рациональная организация потоков ресурсов по времени и т. д.

2.2.3 SADT- модель - набор взаимосвязанных диаграмм

После того как определены объект, цель и точка зрения модели, начинается первая итерация процесса моделирования по методологии SADT. Автор модели определяет, что включить в модель, а что исключить из нее. Точка зрения диктует автору модели выбор нужной информации об объекте и форму ее подачи. Цель становится критерием окончания моделирования. Конечным результатом этого процесса является набор тщательно взаимосвязанных описаний, начиная с описания самого верхнего уровня всей системы и кончая подробным описанием деталей или операций системы.

Каждое из таких тщательно согласованных описаний называется диаграммой.

SADT-модель объединяет и организует диаграммы в иерархические структуры, в которых диаграммы наверху модели менее детализированы, чем диаграммы нижних уровней. Другими словами, модель SADT можно представить в виде древовидной структуры диаграмм, где верхняя диаграмма является наиболее общей, а самые нижние наиболее детализированы.

Примером такой модели является SADT- модель проекта создания конкурентоспособного ДЛА, начальные фрагменты

которой представлены на рис. 2.1-2.5. Цель этого проекта – создание ДЛА, способного в конкурентной борьбе завоевать свою нишу на рынке и на этой основе обеспечить прибыль в процессе его эксплуатации, окупив все первоначальные инвестиции на его проектирование, создание опытного образца и развертывание серийного производства. Диаграмма на рис. 2.1 (на вершине модели) описывает проект как процесс, преобразующий исходные требования к создаваемому ДЛА (сформулированные в виде технического задания – ТЗ) и начальные инвестиции в экономический результат (получение прибыли в течение жизненного цикла). Для этого используются механизмы: оборудование и разработчики (персонал), которые руководствуются современными CALS-стандартами.

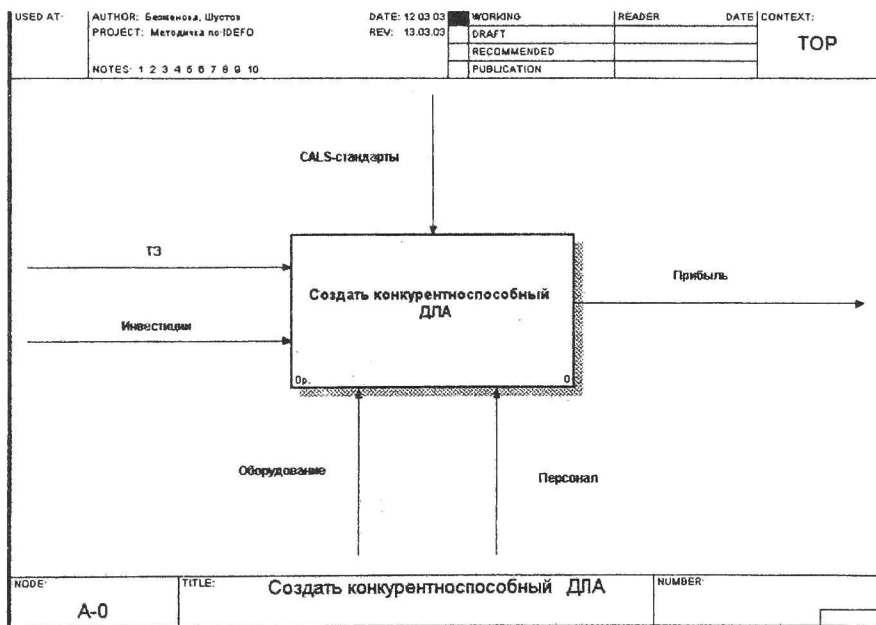


Рис. 2.1. Контекстная диаграмма процесса создания конкурентноспособного ДЛА

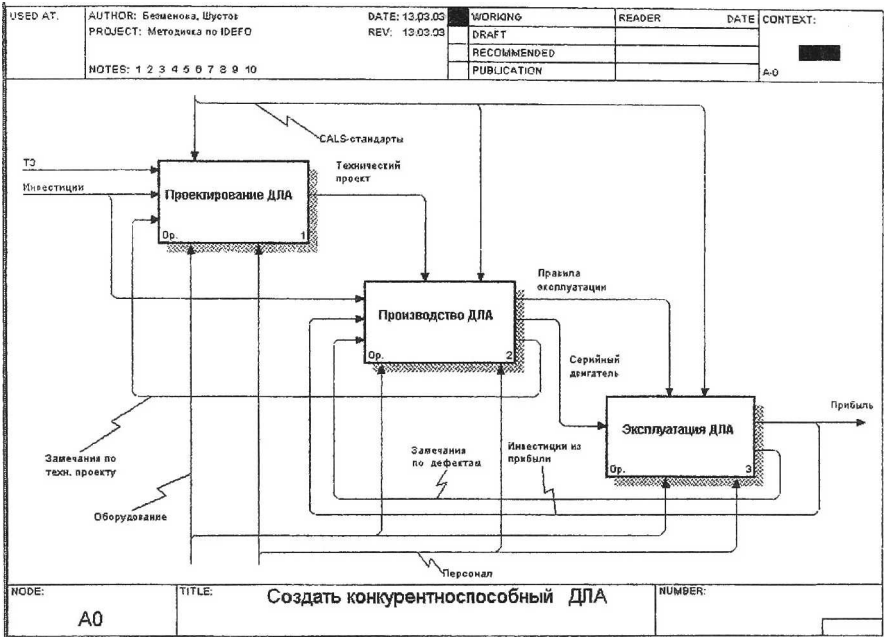


Рис. 2.2. Декомпозиция процесса создания ДЛА (первого уровня)

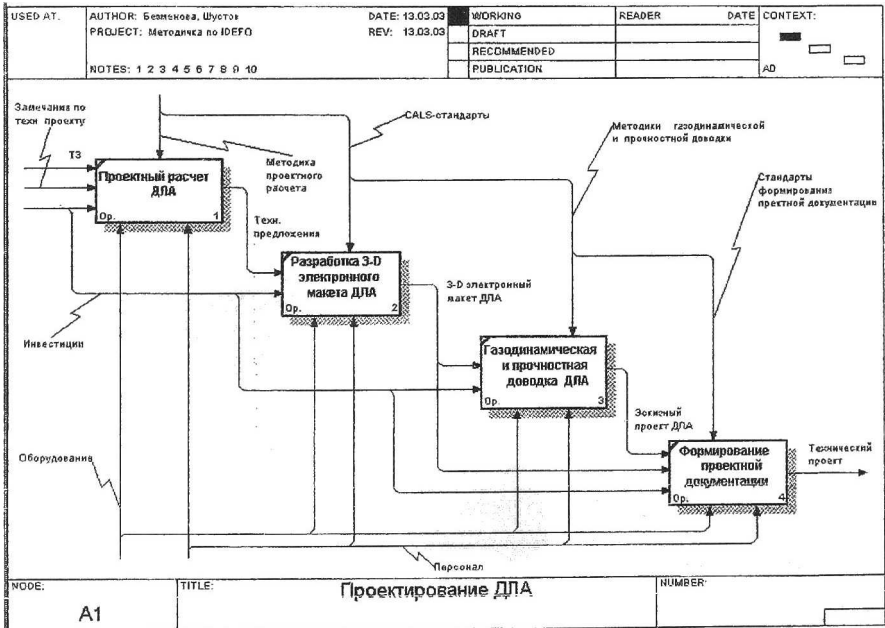


Рис. 2.3. Декомпозиция блока «Проектирование ДЛА»

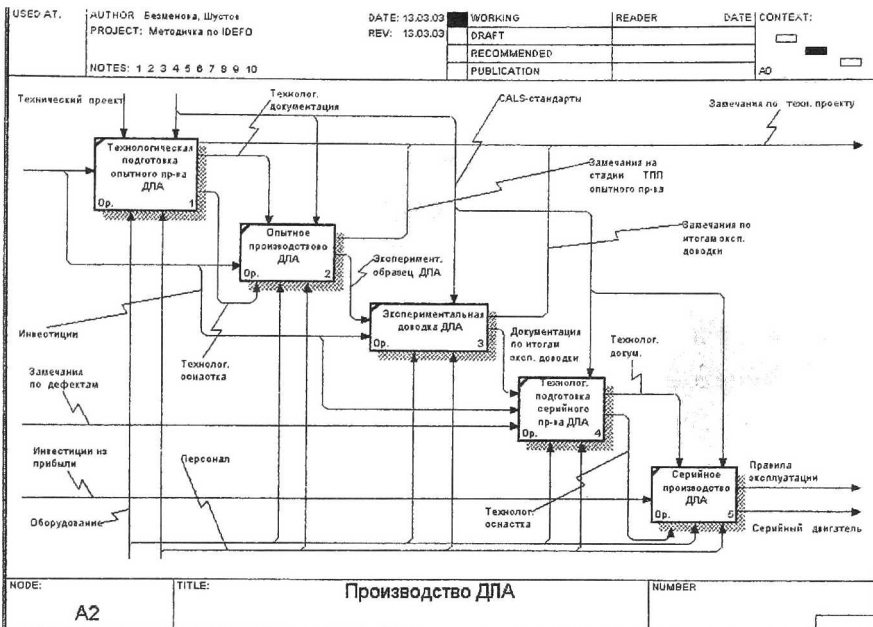


Рис. 2.4. Декомпозиция блока «Производство ДПА»

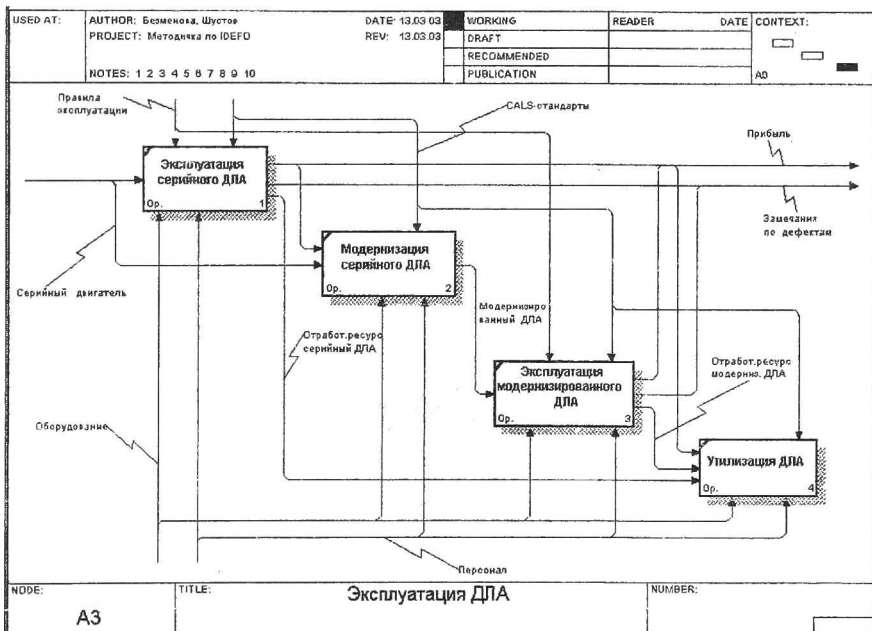


Рис. 2.5. Декомпозиция блока «Эксплуатация ДПА»

На рис. 2.1 и 2.2 представлены две диаграммы из модели жизненного цикла создания авиационного двигателя. Диаграмма на рис. 2.1 (на вершине модели) описывает жизненный цикл как процесс, преобразующий исходные требования к авиационному двигателю и ресурсы в экономический результат (получение прибыли в течение жизненного цикла).

Диаграмма на рис. 2.2 детализирует диаграмму на рис. 2.1, выделяя три главные стадии жизненного цикла создания авиационного двигателя – проектирование, производство и эксплуатацию. Таким образом, общая функция, указанная на верхней диаграмме (рис. 2.1), детализируется с помощью трех функций на нижней диаграмме. В свою очередь, каждая из диаграмм на рисунках 2.3, 2.4 и 2.5 раскрывает содержание SADT-блоков диаграммы, представленной на рис. 2.2.

Это пример того, как SADT организует описание системы, создавая иерархию добавляющихся на каждом уровне деталей.

Вернемся к диаграмме «Создать конкурентоспособный ДЛА» на рис. 2.1. Обратим внимание на то, что на этой диаграмме показана взаимная связь трех SADT-блоков с помощью стрелок, которые символизируют объекты жизненного цикла ДЛА. Если вы внимательно посмотрите на диаграмму, то заметите, что некоторые дуги доходят до ее границы. Посмотрите еще внимательнее, и вы увидите, что имена этих дуг совпадают с теми, что указаны на дугах самой верхней диаграммы (рис. 2.1).

Это пример того, как SADT соединяет диаграммы в модели через объекты системы. Такая схема соединения требует согласованного наименования и учета объектов системы с тем, чтобы две диаграммы можно было рассматривать как связанные между собой. Например, функциональный блок на рис. 2.1 имеет 6 дуг, и каждая из них может быть найдена среди дуг, идущих к границе или от границы диаграммы на следующем уровне (рис. 2.2).

Резюме по разделу 2.2

Сложности, связанные с описанием многих искусственных систем, объясняются тем, что эти системы слишком велики для того, чтобы можно было просто перечислить все их компоненты. С другой стороны, они могут быть упрощены за счет обобщающих предположений. Методология SADT создана специально для представления таких сложных систем путем построения моделей. SADT-модель – это описание системы, у которого есть единственный субъект, цель и одна точка зрения. Целью служит набор вопросов, на которые должна ответить модель. Точка зрения – позиция, с которой описывается система. Цель и точка зрения – это основополагающие понятия SADT. В этой главе мы дали о них лишь беглое представление, оставляя более подробное рассмотрение до следующей главы. Описание модели SADT организовано в виде иерархии взаимосвязанных диаграмм. Вершина этой древовидной структуры представляет собой самое общее описание системы, а ее основание состоит из наиболее детализированных описаний.

2.3 Синтаксис SADT-диаграмм

SADT-диаграмма является основным рабочим элементом при создании SADT- модели. Разработчик диаграмм и моделей обычно называется аналитиком, или, в терминологии SADT, автором. Диаграммы имеют собственные синтаксические правила, отличающиеся от синтаксических правил моделей. Важно их хорошо понимать, поскольку графические обозначения имеют особый смысл. Вы увидите, что графика SADT позволяет определить различные системные функции и показать, как функции влияют друг на друга. Это учебное пособие посвящено только функциональным SADT-диаграммам, поэтому в данной главе обсуждаются синтаксические правила только для функциональных SADT-диаграмм.

2.3.1 Блоки и дуги SADT - диаграмм

В предыдущем разделе мы видели, что каждая SADT-диаграмма содержит блоки и дуги. Блоки изображают функции моделируемой системы, а дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними (рисунки 2.1-2.5). Диаграмме дается название, которое располагается в центре нижней части ее бланка. Например, на рис. 2.2 диаграмма имеет название «Создать конкурентоспособный ДЛА», а диаграмма на рис. 2.3 называется «Проектирование ДЛА». На каждой диаграмме написана стандартная идентифицирующая ее информация: автор диаграммы, частью какого проекта является работа, дата создания или последнего пересмотра диаграммы, статус диаграммы. Вся идентифицирующая информация располагается в верхней части бланка диаграммы. Например, в верхней части диаграммы на рис. 2.2 указано, что авторами SADT-модели «Создать конкурентоспособный ДЛА» являются Безменова Н.В. и Шустов С.А., что эта модель создана в рамках проекта «Методичка по IDEF0», дата создания модели 13.03.2003, это рабочий (working), т.е. промежуточный, а не окончательный (publication) вариант модели.

Функциональные блоки на диаграммах изображаются прямоугольниками – SADT-блоками (рис. 2.6).

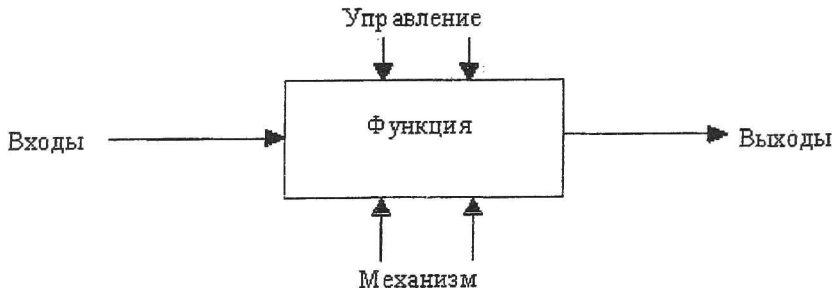


Рис. 2.6. Функциональный SADT-блок и интерфейсные дуги

SADT-блок представляет функцию или активную часть системы, поэтому названиями блоков служат глаголы или глагольные обороты. Например, названиями SADT-блоков диаграммы «Создать конкурентоспособный ДЛА» являются: «Проектирование ДЛА», «Производство ДЛА», «Эксплуатация ДЛА», как показано на рис. 2.2.

Кроме того, SADT требует, чтобы в диаграмме было не менее трех и не более шести SADT-блоков. Эти ограничения поддерживают сложность диаграмм и модели на уровне, удобном для чтения, понимания и использования.

В отличие от других графических методов структурного анализа в SADT каждая сторона блока имеет особое, вполне определенное назначение. Левая сторона блока предназначена для входов, верхняя - для управления, правая для выходов, нижняя для механизмов. Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, кто, что и как выполняет функция. Например, SADT-блок диаграммы на рис. 2.2 с названием «Проектирование ДЛА» может быть интерпретирован следующим образом: *техническое задание и инвестиции в процессе проектирования преобразуются в технический проект по правилам стандартов проектирования с помощью оборудования и персонала.*

2.3.2 Размещение блоков на SADT-диаграммах

SADT-блоки никогда не размещаются на диаграмме случайным образом. Они размещаются по степени важности, как ее понимает автор диаграммы. В SADT этот относительный порядок называется доминированием. Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы. Например, самым доминирующим блоком диаграммы может быть

либо первый из требуемой последовательности функций (как на рис. 2.2), либо планирующая или контролирующая функция, влияющая на все другие функции.

Наиболее доминирующий блок обычно размещается в верхнем левом углу диаграммы, а наименее доминирующий – в правом нижнем углу. В результате получается «ступенчатая» схема, подобная представленной на рис. 2.2 для блоков 1, 2, 3.

Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные. Чтобы подчеркнуть это, SADT-аналитик может перенумеровать блоки в соответствии с порядком их доминирования. Порядок доминирования может обозначаться цифрой, размещенной в правом нижнем углу каждого прямоугольника: 1 будет указывать на наибольшее доминирование, 2 - на следующее после наибольшего, и т.д. Так, на рис. 2.2 показано, что SADT-блок «*Проектирование ДЛА*» влияет на процесс производства двигателя, а процесс «*Эксплуатация ДЛА*», в свою очередь, может быть реализован только после его изготовления.

Блоки в SADT должны быть пронумерованы. Номера блоков служат однозначными идентификаторами для системных функций и автоматически организуют эти функции в иерархию модели. Используя номера блоков и оценивая влияние, которое один блок оказывает на другой, аналитик может организовать модель по принципу функционального доминирования. Это позволяет согласовать иерархический порядок функций в модели с уровнем влияния каждой функции на остальную часть системы. Поэтому мы настоятельно рекомендуем по мере возможности нумеровать блоки в соответствии с порядком их доминирования.

2.3.3 Дуги, их назначение и классификация

Дуги на SADT-диаграмме изображаются одинарными линиями со стрелками на концах. Для функциональных SADT-диаграмм

дуга представляет собой множество объектов. Мы вынуждены использовать здесь общее понятие «объекты», поскольку дуги в SADT могут представлять, например, планы, данные в компьютерах, машины и информацию. Дуги диаграммы «Проектирование ДЛА» на рис. 2.2 представляют: материалы, написанные на бумаге (например, *исходные данные для проекта*), технические устройства (например, *оборудование*), людей (например, *персонал*), рабочие чертежи (например, *технический проект*), управленческую информацию (например, *законы, стандарты*).

Так как в SADT дуги изображают объекты, они описываются (помечаются) существительными или существительными с определениями, располагающимися достаточно близко к линии дуги. Мы настоятельно рекомендуем размещать описания дуг, называемые метками, как можно ближе к линиям дуг, не нарушая, однако, читабельность диаграмм. Это устраняет неопределенность в том, к какой дуге относится метка, и исключается необходимость в дополнительных графических связях.

Между объектами и функциями возможны четыре отношения: вход, управление, выход, механизм. Каждое из этих отношений изображается дугой, связанной с определенной стороной SADT-блока. По соглашению левая сторона SADT-блока предназначена для входных дуг, верхняя сторона – для управленческих дуг, правая сторона для выходных дуг, нижняя сторона – для дуг механизмов. Таким образом, стороны блока чисто графически сортируют объекты, изображаемые касающимися блока дугами.

Входные дуги изображают объекты, используемые и преобразуемые функциями. Например, для диаграммы на рис. 2.2 в процессе «Проектирование ДЛА» *исходные данные* преобразуются в *технический проект*.

Управленческие дуги представляют информацию, управляющую действиями функций. Обычно управляющие дуги несут

информацию, которая указывает, что и как должна выполнять функция, а также вводят ограничения на эти функции. Например, в диаграмме на рис. 2.2 для SADT-блока «Проектирование ДЛА» дуга «законы и нормативные акты» через стандарты определяет правила выполнения чертежей, а через законы определяет требования безопасности, которым должен отвечать проектируемый двигатель.

Выходные дуги изображают объекты, в которые преобразуются входы. Например, *исходные данные и ресурсы в виде инвестиций* преобразуются процессом «Проектирование ДЛА» в *технический проект* (рис. 2.2).

Дуги механизмов отражают то, каким образом (т.е. с помощью каких механизмов) SADT-блок вход преобразуется в выход. Например, в SADT-блоке «Проектирование ДЛА» на рис. 2.2 вход преобразуется в выход с помощью персонала (*инженеров-конструкторов*) и оборудования (например, *компьютеров*).

Итак, SADT-диаграмма составлена из блоков, связанных дугами, которые определяют, как блоки влияют друг на друга. Это влияние может выражаться либо в передаче выходной информации к другой функции для дальнейшего преобразования, либо в выработке управляющей информации, предписывающей, что именно должна выполнять другая функция.

Поэтому SADT-диаграммы не являются ни просто блок-схемами, ни просто диаграммами потоков данных. Это предписывающие диаграммы, представляющие входные и выходные преобразования и указывающие правила этих преобразований. Дуги на них изображают интерфейсы между функциями системы, а также между системой и ее окружающей средой.

В методологии SADT требуется только пять типов взаимосвязей между блоками для описания их отношений: выход-управление, выход-вход, обратная связь по управлению, обратная

связь по входу, выход-механизм.

Связи по управлению и входу являются простейшими, поскольку они отражают прямые воздействия, которые интуитивно понятны и очень просты.

Связь выход-управление возникает тогда, когда выход одного блока непосредственно влияет на блок с меньшим доминированием. Например, на диаграмме «Создать конкурентоспособный ДЛА» (рис. 2.2) SADT - блок «Проектирование ДЛА» влияет на SADT-блок «Производство ДЛА» в соответствии с детальными указаниями, содержащимися в дуге «технический проект». Аналогично SADT-блок «Производство ДЛА» на этом же рис. 2.2 влияет на блок «Эксплуатация ДЛА» через дугу «правила эксплуатации», которая реализует связь по управлению.

Связь выход-вход возникает тогда, когда выход одного блока становится входом для блока с меньшим доминированием. Например, на рис. 2.2 выход SADT-блока «Производство ДЛА» «серийный двигатель» является входом в SADT-блок «Эксплуатация ДЛА».

Обратная связь по управлению и обратная связь по входу являются более сложными, поскольку они представляют итерацию или рекурсию. А именно: выходы из одной функции влияют на будущее выполнение других функций, что впоследствии влияет на исходную функцию.

Обратная связь по управлению возникает тогда, когда выход некоторого блока влияет на блок с большим доминированием. Примером такой связи является дуга «Правила формирования технических требований к изготовлению деталей» на рис. 2.2.

Обратная связь по входу имеет место тогда, когда выход одного блока становится входом другого блока с большим доминированием. Например, на рис. 2.2 выход SADT-блока «Производство ДЛА» «замечания по техническому проекту» становится входом SADT-блока «Проектирование ДЛА».

Связи «выход-механизм» встречаются более редко и представляют особый интерес. Они отражают ситуацию, при которой выход одной функции становится средством достижения цели для другой. Связи «выход-механизм» характерны при распределении ресурсов (например, требуемые инструменты и оборудование, техническая оснастка, обученный персонал, финансирование и т.д.). Примером такой связи являются дуги *«технологическая оснастка в опытном производстве»* и *«технологическая оснастка в серийном производстве»* на рис. 2.4.

2.3.4 Правила разветвления и слияния дуг

Дуга в SADT редко изображает один объект (напомним, что в SADT-моделировании под объектом понимается объект информационного описания произвольной природы). Часто дуга символизирует набор объектов. Например, дуга, именуемая *«технический проект»* (выход SADT- блока *«Проектирование ДЛА»* на рис. 2.2), отражает *рабочие чертежи, их спецификации, требования к изготовлению детали* и т.д.

Так как дуги представляют наборы объектов, они могут иметь множество начальных точек (источников) и конечных точек (назначений). Поэтому дуги могут разветвляться и соединяться различными сложными способами.

Вся дуга или ее часть может выходить из одного или нескольких блоков и заканчиваться в одном или нескольких блоках.

Например, дуга *«инвестиции в производство ДЛА»* на входе в SADT-блок *«Производство ДЛА»* (рис. 2.2) разделяется на дуги *«инвестиции в технологическую подготовку опытного производства ДЛА»*, *«инвестиции в опытное производство ДЛА»*, *«инвестиции в экспериментальную доводку ДЛА»*, *«инвестиции в технологическую подготовку серийного производства ДЛА»* (рис. 2.4).

С другой стороны, дуги *«замечания на стадии*

технологической подготовки опытного производства ДЛА», «замечания на стадии опытного производства ДЛА» и «замечания по итогам эксплуатации ДЛА» сливаются в дугу «замечания по техническому проекту ДЛА».

Для объяснения того, как дуги представляют разъединение и соединение наборов объектов, в SADT были разработаны специальные соглашения относительно представления и описания разветвлений и соединений дуг.

Разветвления дуг, изображаемые в виде расходящихся линий, означают, что все содержимое дуг или его часть может появиться в каждом ответвлении дуги. Дуга всегда помечается до разветвления, чтобы дать название всему набору. Кроме того, каждая ветвь дуги может быть помечена или не помечена в соответствии со следующими правилами:

- непомеченные ветви содержат все объекты, указанные в метке дуги перед разветвлением (т.е. все объекты принадлежат этим ветвям);
- ветви, помеченные после точки разветвления, содержат все объекты или их часть, указанные в метке дуги перед разветвлением (т.е. каждая метка ветви уточняет, что именно содержит ветвь).

Слияние дуг в SADT, изображаемое как сходящиеся вместе линии, указывает, что содержимое каждой ветви идет на формирование метки для дуги, являющейся результатом слияния исходных дуг. После слияния результирующая дуга всегда помечается для указания нового набора объектов, возникшего после объединения. Кроме того, каждая ветвь перед слиянием может помечаться или не помечаться в соответствии со следующими правилами:

- непомеченные ветви содержат все объекты, указанные в общей метке дуги после слияния (т.е. все объекты исходят из всех

ветвей);

- помеченные перед слиянием ветви содержат все или некоторые объекты из перечисленных в общей метке после слияния (т.е. метка ветви ясно указывает, что содержит ветвь).

Резюме к разделу 2.3

SADT-диаграмма содержит от трех до шести SADT-блоков, связанных дугами. SADT-блоки на диаграмме изображают системные функции, а дуги изображают множество различных объектов системы. SADT-блоки обычно располагаются на диаграмме в соответствии с порядком их доминирования, т.е. их важностью относительно друг друга. Дуги, связывающие SADT-блоки, изображают наборы объектов и могут разветвляться и соединяться различными способами. Однако, разветвляясь и соединяясь, дуги должны во всех случаях сохранять представляемые ими объекты.

2.4 Синтаксис SADT- моделей

Одна SADT-диаграмма сложна сама по себе, поскольку она содержит от двух до восьми блоков (по некоторым рекомендациям до шести), связанных множеством дуг. Для адекватного описания системы требуется несколько таких диаграмм (для реального предприятия их количество исчисляется сотнями). Диаграммы, собранные и связанные вместе, становятся SADT-моделью. В SADT дополнительно к правилам синтаксиса диаграмм существуют правила синтаксиса моделей. Синтаксис SADT-моделей позволяет аналитику определить границу модели, связать диаграммы в одно целое и обеспечить точное согласование между диаграммами.

Никакой другой метод структурного анализа не позволяет так точно, как SADT-методология, соединять диаграммы в тщательно организованные комплекты, называемые моделями.

2.4.1 Контекстная диаграмма и диаграммы декомпозиции SADT-модели

SADT-модель является иерархически организованной совокупностью диаграмм. Выше уже отмечалось, что SADT-диаграммы обычно состоят из двух-восьми SADT-блоков, каждый из которых потенциально может быть детализирован на другой диаграмме. Каждый SADT-блок может пониматься как отдельный тщательно определенный объект. Разделение такого объекта на его структурные части (SADT-блоки и дуги, составляющие диаграмму) называется *декомпозицией*.

Декомпозиция формирует границы, и каждый SADT-блок рассматривается как формальная граница некоторой части целой системы, которая описывается. Другими словами, SADT-блок и касающиеся его дуги определяют точную границу диаграммы, представляющей декомпозицию этого SADT-блока. Эта диаграмма, называемая диаграммой с потомком, описывает все, связанное с этим SADT-блоком и его дугами, и не описывает ничего вне этой границы. Декомпозируемый SADT-блок называется *родительским SADT-блоком*, а содержащая его SADT-диаграмма – соответственно *родительской SADT-диаграммой*. Таким образом, SADT-диаграмма является декомпозицией некоторого ограниченного объекта.

Принцип ограничения моделируемого объекта встречается на каждом уровне. Один SADT-блок и несколько дуг на самом верхнем уровне используются для определения границы всей системы. Этот SADT-блок описывает общую функцию, выполняемую системой. Дуги, касающиеся этого SADT-блока, описывают главные управления, входы, выходы и механизмы этой системы. Диаграмма, состоящая из одного SADT-блока и его дуг, определяет границу системы и называется *контекстной диаграммой модели*. Таким образом, этот SADT-блок изображает границу системы: все, лежащее внутри него, является частью

описываемой системы, а все, лежащее вне него, образует среду системы.

На рис. 2.1 показан пример контекстной диаграммы, представляющей собой верхний уровень модели жизненного цикла создания ДЛА. SADT-блок с названием «Создать конкурентоспособный ДЛА» описывает самую общую функцию и имеет нулевой номер (SADT-блок самого верхнего уровня модели всегда нумеруется нулем). Дуги контекстной диаграммы определяют ее интерфейс (сопряжение) с внешней средой. Фактически дуги контекстной диаграммы описывают перечень объектов, совокупность которых представляет собой модель внешней среды для моделируемого объекта.

2.4.2 Декомпозиция контекстной диаграммы и правило нумерации узлов

SADT-модели развиваются в процессе структурной декомпозиции сверху вниз. Сначала осуществляется декомпозиция SADT-блока на контекстной SADT-диаграмме, результатом которой является диаграмма, обычно содержащая от трех до шести SADT-блоков. При этом название новой SADT-диаграммы совпадает с названием декомпозируемого SADT-блока. Так, название диаграммы на рис. 2.2 такое же, как и название SADT-блока на контекстной диаграмме – «Создать конкурентоспособный ДЛА». При необходимости показать более детально структуру каждого из SADT-блоков диаграммы проводится их декомпозиция по правилам, указанным выше. Так, на рис. 2.3-2.5 показаны результаты декомпозиции каждого из SADT-блоков контекстной диаграммы на рис. 2.2.

Результатом этого процесса декомпозиции является SADT-модель, контекстная диаграмма верхнего уровня которой описывает систему в общих терминах «черного ящика», а диаграммы нижнего уровня описывают детализированные аспекты

и операции системы с той глубиной, которая необходима для достижения целей моделирования.

Таким образом, каждая SADT-диаграмма представляет собой некоторую законченную часть всей модели.

В методологии SADT идентифицируется каждая диаграмма данной модели посредством того, что называется «номер узла». Номер узла для контекстной диаграммы имеет следующий вид: название модели или аббревиатура, косая черта, заглавная буква A (Activity в функциональных диаграммах), дефис и ноль. Например, номером узла для контекстной SADT-диаграммы на рис. 2.1 является A-0. Номером узла SADT-диаграммы, декомпозирующей контекстную диаграмму, является тот же номер узла, но без дефиса (например, на рис. 2.2 номер узла A0). Все другие номера узлов образуются посредством добавления к номеру узла родительской SADT-диаграммы номера декомпозируемого SADT-блока. При этом первый ноль при образовании номера узла принято опускать. На рисунках 2.3-2.5 показаны три SADT-диаграммы, которые являются результатом декомпозиции SADT-блоков диаграммы на рис. 2.2. Эти диаграммы имеют номера A1, A2, A3 (но не A01, A02, A03, т.к. обозначение ноля опущено).

Вопросы для самоконтроля

1. В чем причина возникновения методологии SADT?
2. Что такое модель в методологии SADT?
3. Что понимают под целью моделирования и точкой зрения в методологии SADT?
4. Какие элементы существуют в SADT-модели?
5. Как изображаются функции на SADT-диаграмме?
6. Что понимается под доминированием функций и как это отображается на SADT-диаграмме?
7. Что представляют собой дуги на SADT-диаграмме?

8. Какие 4 вида дуг изображаются на SADT-диаграмме?
9. Какие виды связей допустимы в SADT-модели?
10. В чем содержание связи выход-управление?
11. В чем содержание обратной связи выход-вход?
12. Сколько блоков может быть на диаграмме декомпозиции в SADT-модели?
13. В чем назначение контекстной диаграммы в SADT-модели?

3 CASE-СРЕДСТВО BPWIN ДЛЯ ПОДДЕРЖКИ МЕТОДОЛОГИИ IDEF0

Для проведения анализа и реорганизации бизнес-процессов существует целый ряд программных продуктов различных производителей [7]. Компания Platinum technologies предлагает разработчикам информационных систем ряд взаимодействующих между собой CASE-средств: BPWin – для поддержки методологии IDEF0 (функциональное моделирование), IDEF3 (WorkFlow Diagram) и DFD (DataFlow Diagram); ERWin – для поддержки моделирования сущность-связь (ERD – Entity Relationship Diagram).

Нотация DFD включает такие понятия, как внешняя сущность, хранилище данных, потоки данных, что делает ее более удобной для моделирования документооборота. Методология IDEF3 включает элемент «перекресток», что позволяет описать логику взаимодействия элементов системы.

3.1 Общие сведения о CASE-средстве BPWin

В BPWin реализована технология функционального моделирования IDEF0. Описание процессов ведется в двух вариантах: существующие бизнес-процессы (модель AS-IS) и идеальное положение вещей – того, к чему надо стремиться (модель TO BE). Методология IDEF0 предписывает построение модели в виде совокупности взаимосвязанных диаграмм, отражающих различные уровни абстрагирования. При этом BPWin позволяет на любой ветви модели переключаться на нотацию IDEF3 или DFD и создавать смешанную модель.

На основе модели BPWin можно создать модель данных, для чего Platinum technologies предлагает мощный инструмент – ERWin. Процесс преобразования одной модели в другую полностью не автоматизирован, однако Platinum technologies предлагает удобный интерфейс для облегчения построения модели

данных на основе функциональной модели. ERWin имеет два уровня представления данных – логический и физический. На логическом уровне представлены сами данные как таковые (словарь данных) и взаимосвязи между ними. Физический уровень – это по существу реализация модели данных с использованием конкретной СУБД. Для этого в ERWin предусмотрена интеграция с такими популярными средствами разработки клиентской части, как Power Builder, Delphi, Visual Basic. Процесс разработки клиентской части для работы с базой данных завершает полный цикл разработки информационной модели объекта, а функциональная модель, разработанная с использованием программного продукта BPWin, является первым звеном этой цепи.

3.2 Запуск программы BPWin

Запуск программы осуществляется через меню Пуск→ Программы→PLATINUM BPWin→BPWin 5.0 или через панель быстрого старта в меню Пуск. Запустить программу можно также, дважды щелкнув мышью на значке уже созданного файла.

Если открывать программу из меню Пуск, то первое открывшееся окно задает параметры работы в группе. Для самостоятельной работы нажимайте кнопку Cancel (рис. 3.1).

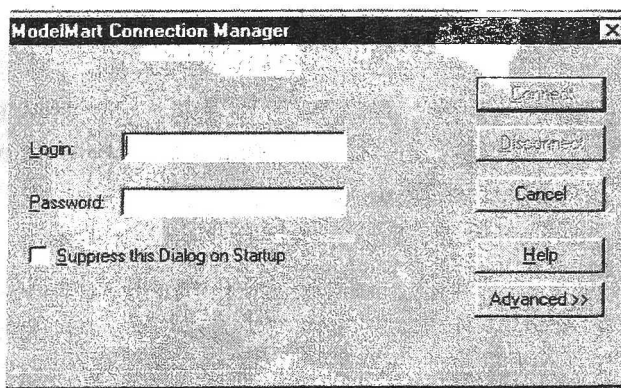


Рис. 3.1

Затем появляется диалоговое окно (рис. 3.2), в котором вам предлагается создать новую модель (для этого выбираете пункт Create model) или открыть уже существующую (выбирайте Open model).

Для создания новой модели в окошке Name введите название новой модели, затем выберите тип модели (Type), отметив необходимый вариант точкой:

- Business process (IDEF0) – модель бизнес-процессов;
- Workflow (IDEF3) – модель потоков работ;
- Dataflow (DFD) – модель потоков данных (рис. 3.2).

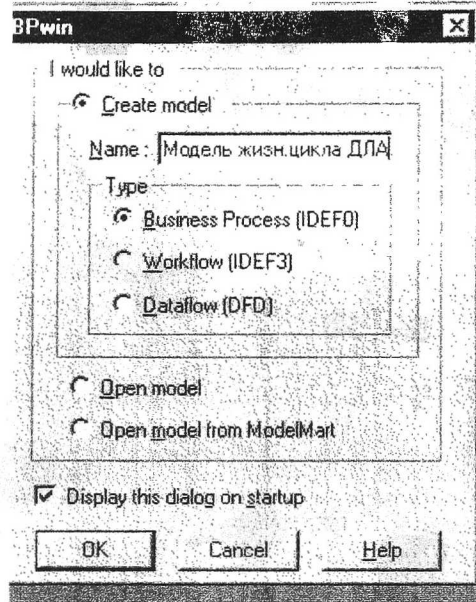



Рис. 3.2

3.3 Формирование контекстной диаграммы

Новая модель всегда начинается с контекстной диаграммы (рис. 3.3). В IDEF0-модели она состоит из одного блока, отражающего взаимосвязь моделируемой системы с внешней средой. Название этого блока должно наиболее полно отражать основную функцию системы.

Работать можно в двух основных режимах – режиме выделения и режиме рисования объектов. Режимы переключаются кнопками на панели инструментов. Создавать новые объекты нужно в режиме рисования, редактировать уже созданные – в режиме выделения объектов, перейти в который можно нажав кнопку .



USED AT:	AUTHOR:	DATE: 05.03.03	WORKING	READER	DATE	CONTEXT
	PROJECT: Модель жиконциста ДПА	REV: 05.03.03	DRAFT			TOP
	NOTES: 1 2 3 4 5 6 7 8 9 10		RECOMMENDED			
			PUBLICATION			
						
NOOE:	TITLE:	NUMBER:				
A-0						

Рис. 3.3

Для ввода названия перейдите в режим выделения объектов, затем дважды щелкните мышкой на блоке. После этого появится окно редактирования названия блока (рис. 3.4). Смело вводите название и нажимайте кнопку ОК. Для редактирования блока используйте контекстное меню, которое вызывается на блоке правой кнопкой мыши. Чтобы изменить шрифт, выбирайте пункт FontEditor, появится стандартное окно редактирования свойств шрифта, где вы можете выбрать шрифт, его размер, начертание, цвет, а также другие эффекты.

Стрелки рисуются с использованием одного и того же правила: щелчок на начале стрелки, при отпущенной кнопке мыши указатель подводится к нужному месту, щелчок на конце стрелки. Для рисования стрелок выберите соответствующий режим, нажав кнопку . Затем подведите указатель мыши к началу

рисования стрелки. Например, если это стрелка-вход, то она начинается у левой границы диаграммы (рис. 3.5). Подводите туда указатель мыши и щелкайте на черной полосе у границы диаграммы.

Затем подводите указатель мыши туда, где стрелка должна закончиться (если это стрелка-вход, то к левой границе блока) и опять щелкайте мышью на появившемся черном треугольнике.

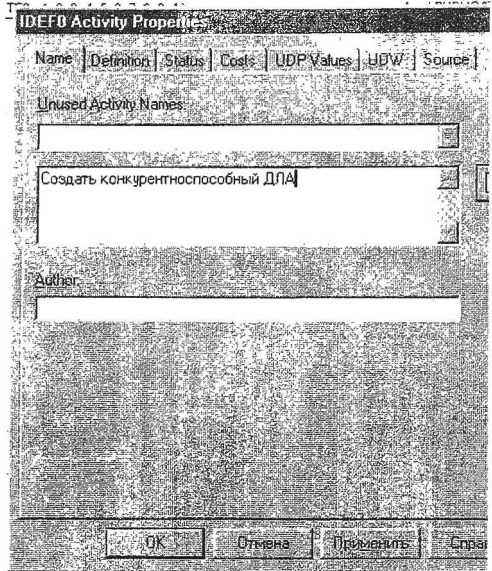


Рис. 3.4

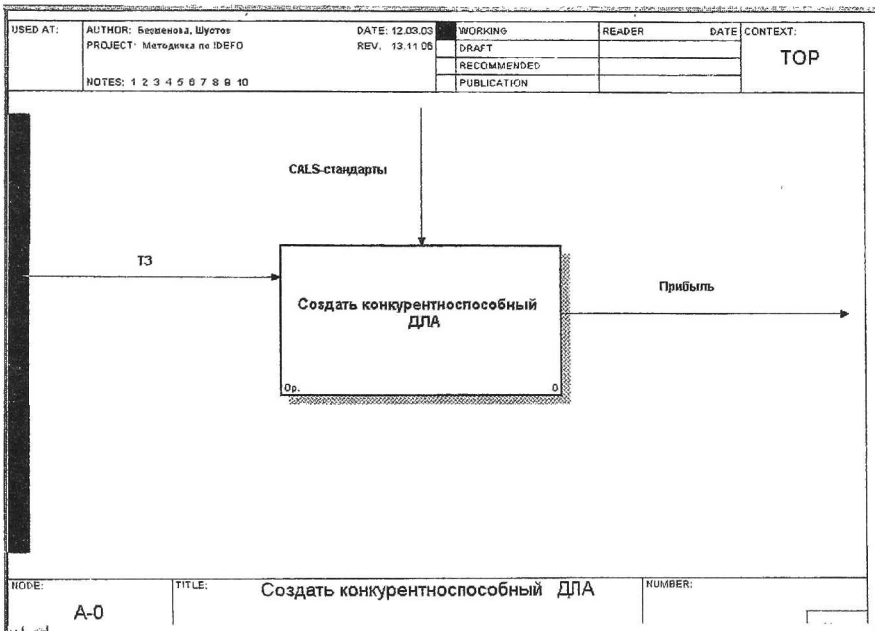



Рис. 3.5

Если все сделано правильно, то появится стрелка. Для того чтобы ввести имя стрелки, перейдите в режим выделения объектов, нажав кнопку , и дважды щелкните по стрелке. Появится диалоговое окно задания названий стрелок (рис. 3.6) . В этом окне представлены все названия стрелок, которые используются в данной модели. Вводите название стрелки в окошке Arrow Name или выбирайте из списка и затем редактируйте (помните при этом, что в одной модели не должно быть стрелок с одинаковыми названиями). После того, как название готово, жмите кнопку ОК.

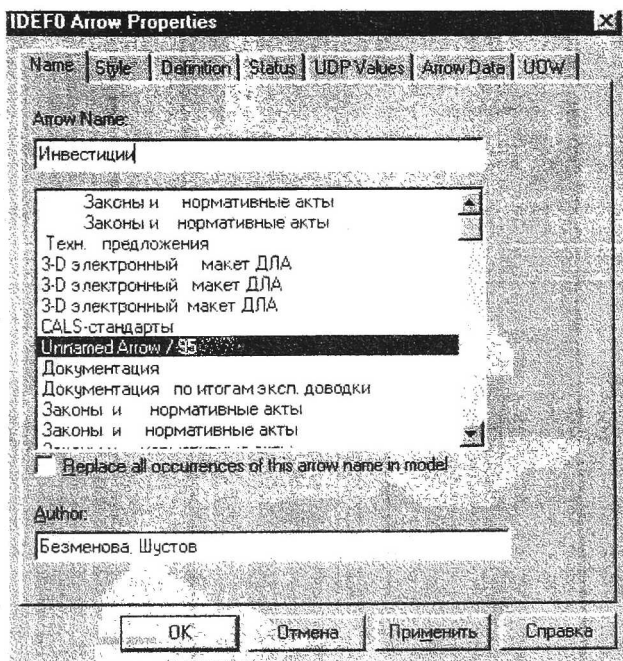


Рис. 3.6

3.4 Описание модели средствами VPWin

Для описания модели выбором пункта меню Edit→Model Properties вызывается диалог Model Properties (рис. 3.7-3.11). Закладка General (рис. 3.7) служит для внесения имени проекта и

модели, имени и инициалов автора и временных рамок модели AS-IS («как есть») и TO-BE («как надо»).

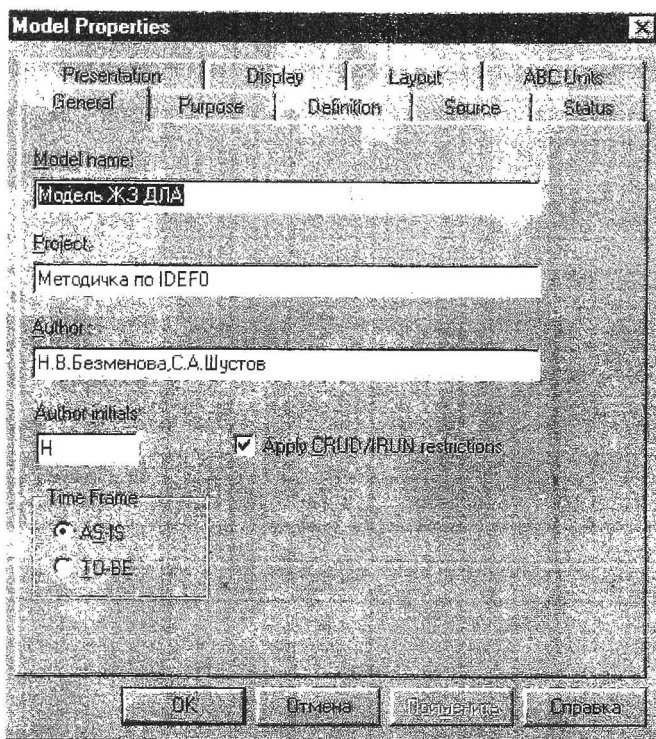


Рис. 3.7

Используя закладку **Purpose** (рис. 3.8), разработчик дает описание цели моделирования (Purpose) и точки зрения (Viewpoint), с которой ведется процесс моделирования.

С помощью закладки **Definition** дается описание области моделирования (рис. 3.9), а в закладке Status (рис. 3.10) - статуса модели (черновой вариант, рабочий, окончательный), время создания и последнего редактирования (отслеживается в дальнейшем автоматически по системной дате).

В закладке **Source** (рис. 3.11) описываются источники информации для построения модели.

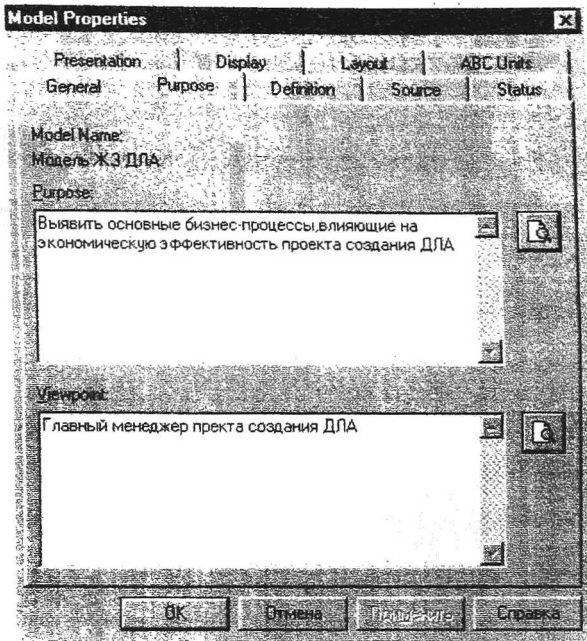


Рис. 3.8

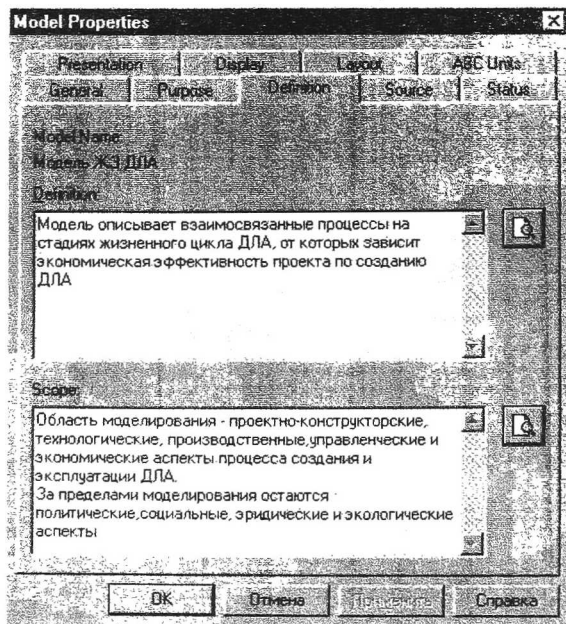


Рис. 3.9

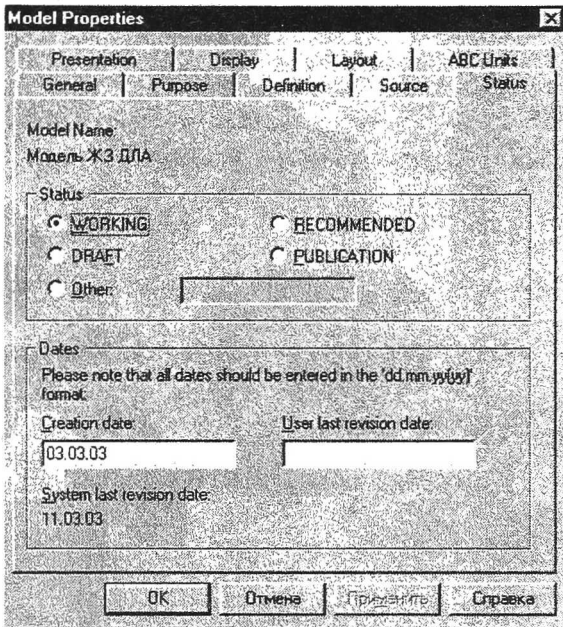


Рис. 3.10

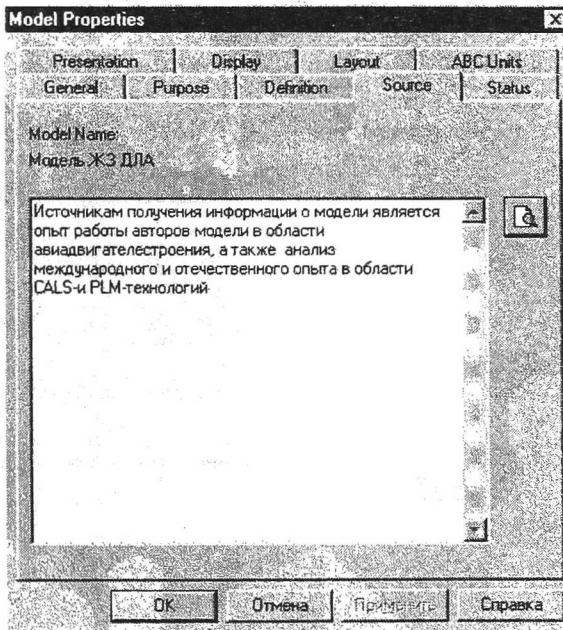




Рис. 3.11

3.5 Построение диаграмм декомпозиции

Теперь у вас уже есть готовая модель, содержащая пока только контекстную диаграмму. Чтобы дополнить ее диаграммой декомпозиции нужно вновь открыть уже существующий файл с моделью в BPWin. Для этого найдите в проводнике значок файла и дважды щелкните на нем мышью.

Если программа уже запущена на компьютере, то воспользуйтесь меню File→Open или кнопкой  на панели инструментов.

Чтобы создать диаграмму декомпозиции, выделите блок, декомпозицию которого будете проводить (пока у вас есть только один блок – выделяйте его, при этом блок и все сопряженные с ним дуги выделяются черным цветом), и нажимайте кнопку  на панели инструментов. При этом появится окно редактирования количества блоков на диаграмме декомпозиции (рис. 3.12). По умолчанию предполагается 4 блока, но возможны вариации от 2 до 8 блоков на одной диаграмме. Выбирайте необходимое количество и нажимайте кнопку ОК.

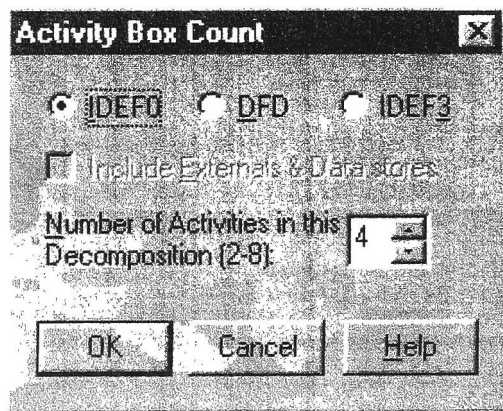


Рис. 3.12

На появившейся диаграмме появится заданное количество пустых блоков (в нашем случае 3, рис. 3.13), а также все стрелки, входящие и выходящие из блока родительской диаграммы, деком-

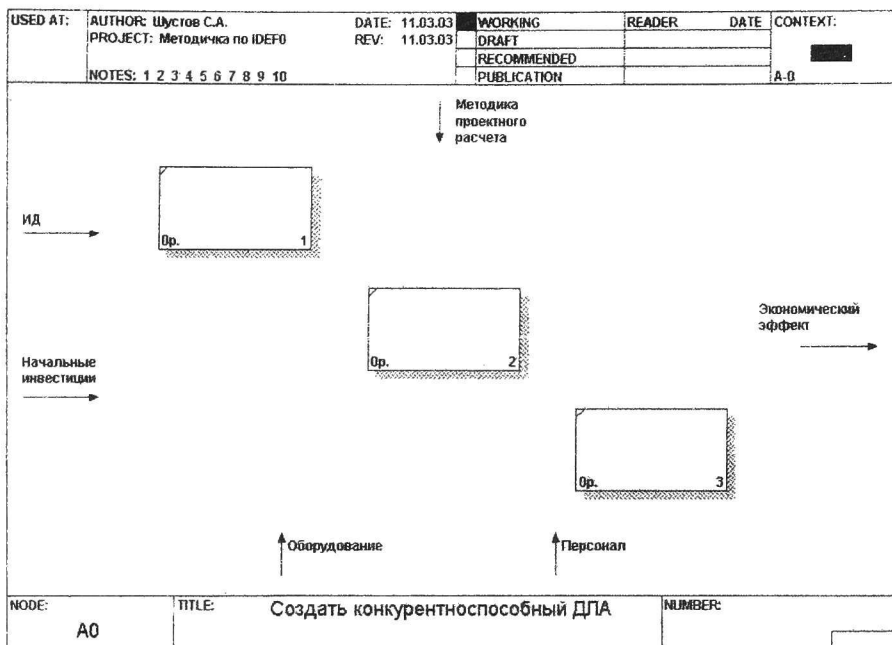



Рис. 3.13

позицию которого вы проводите.

Поскольку в нашем случае это декомпозиция одного блока на контекстной диаграмме, то на диаграмме декомпозиции появятся все стрелки из контекстной диаграммы. Новая диаграмма автоматически будет иметь название того блока, декомпозицию которого она собой представляет.

Теперь ваша задача – задать названия блокам на диаграмме декомпозиции, развести уже существующие стрелки, а также определить связи между блоками и изобразить их стрелками. В нашем случае процесс создания конкурентноспособного ДЛА представляет собой взаимосвязанные процессы, которые называются «Выполнить проектирование ДЛА», «Изготовить ДЛА», «Обеспечить экономическую эффективность...».

В рисовании стрелок действует уже описанное правило – щелкаем мышью в месте начала стрелки и в месте конца стрелки.

Не забудьте при этом перейти в режим рисования стрелок (кнопка ). Для того чтобы направить уже существующую стрелку на вход (управление или механизм) какому-либо блоку (рис. 3.14), щелкайте на самой стрелке, затем подведите указатель мыши к нужной стороне блока до появления черного треугольника, и щелкайте на нем мышью. При этом черные треугольники появятся только на тех сторонах блока, куда разрешено подвести выбранную стрелку (например, стрелка, отображающая управление, не может быть использована как вход или механизм).

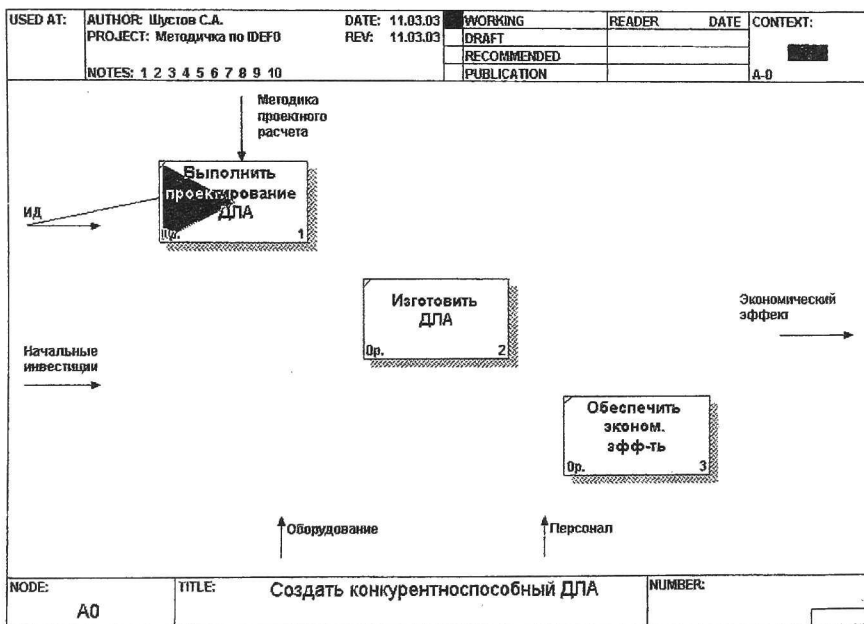



Рис. 3.14

Так же обстоят дела и со стрелками-выходами. Однако, чтобы развести стрелки-выходы, сначала щелкаете мышью на черном треугольнике на правой границе блока, из которого выходит стрелка, а затем уже на самой стрелке (рис. 3.15). В процессе моделирования может возникнуть ситуация, когда одна и та же

стрелка должна поступать на вход нескольким работам или когда несколько однотипных объектов объединены на родительской диаграмме в одну стрелку, а на диаграмме декомпозиции их нужно развести по разным блокам. Для декомпозиции стрелки перейдите в режим рисования стрелок (кнопка ) , затем щелкайте на том месте стрелки, где должно начаться ответвление, и на черном треугольнике на блоке, где должна закончиться стрелка. После этого ответвленной стрелке можно дать собственное название, уточняющее ее смысл, так же как и обычной стрелке.

Будем считать, что с первой задачей вы успешно справились. Попробуем теперь соединить между собой блоки, как показано на рис. 3.15. Для этого последовательно строятся все соединительные стрелки и им даются названия.

Чтобы уничтожить ненужную стрелку, выделите ее и нажмите Delete на клавиатуре.

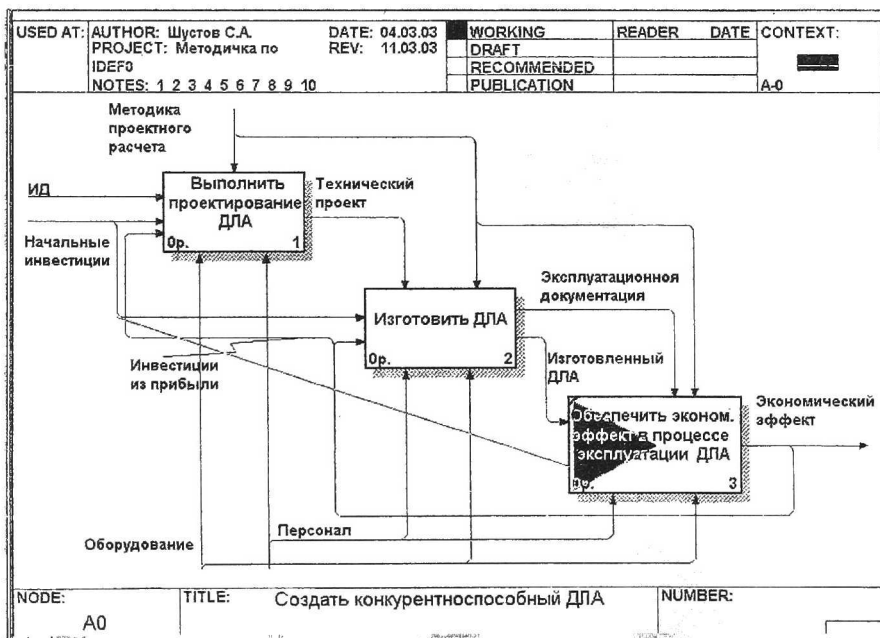




Рис. 3.15

3.6 Сохранение результатов и выход из BPWin

При работе с любыми программами необходимо время от времени сохранять свои результаты. Для этого воспользуйтесь пунктом меню File→Save или кнопкой .

Если вы сохраняете модель в первый раз, то потребуется задать имя файла, в котором она будет храниться на диске. Для этого используется стандартное окошко сохранения файла, в котором нужно выбрать папку, где будет храниться файл, набрать имя файла и нажать кнопку «Сохранить». Затем все изменения будут автоматически сохраняться в этом файле.

Чтобы закрыть программу, используйте меню File→Exit или кнопку  на строке заголовка окна программы. При выходе из программы появится диалоговое окно (рис. 3.16), предлагающее различные варианты выхода – с сохранением (Save as .BP1 file) или без сохранения (Close without saving) изменений, внесенных с момента последнего сохранения.

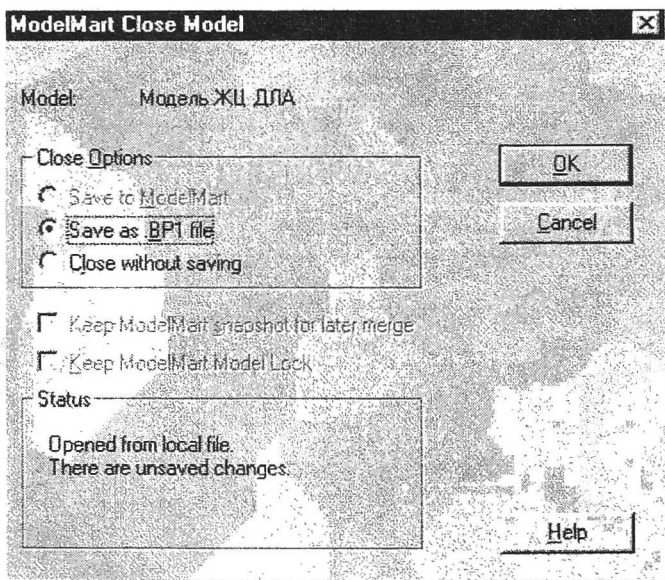


Рис. 3.16.

Вопросы для самоконтроля

1. Какие виды моделей реализованы в программной среде BPWin?
2. Какие два режима существуют для работы в программной среде BPWin?
3. Как в BPWin описать свойства модели?
4. Как в BPWin задавать или редактировать названия блоков в модели?
5. Какова технология рисования стрелок на диаграмме BPWin?
6. Могут ли в модели существовать два блока с одинаковыми именами?
7. Сколько блоков допускается на диаграмме декомпозиции?
8. Как в BPWin задавать или редактировать названия стрелок в модели?
9. Как в BPWin задавать или редактировать параметры шрифта на диаграмме?
10. Какое расширение имеет файл, созданный программой BPWin?

ЗАКЛЮЧЕНИЕ

В учебном пособии изложены основные подходы к моделированию процессов в сложных системах различной природы: экономических, технических, организационных, интеллектуальных, социальных и др.

Методологической основой этих методов является системный подход, основанный на принципе моделирования «сверху вниз».

Практические методики моделирования сформулированы в виде Международных стандартов серии IDEF, причем один из них (IDEF0) является Российским государственным стандартом и входит в стандарт качества ISO серии 9000.

Сегодня внедрение методов функционального моделирования является неотъемлемой частью повышения эффективности управления предприятий, конкурентоспособности, внедрения менеджмента качества, стандартизации по Международным стандартам серии ISO. Поэтому освоение и практическое использование этих методов является одной из важнейших задач управленческого персонала.

Метод функционального моделирования IDEF0 является несколько ограниченной версией методики SADT, которая была разработана в 70-е годы совместно с военным ведомством США и на сегодняшний день является одной из самых успешных практических методик.

В настоящее время SADT-методология описания сложных систем признана международным стандартом и включена в число CALS-стандартов.

SADT является единственной методологией, легко отражающей такие системные характеристики, как управление,

обратная связь и исполнители. Это объясняется тем, что SADT изначально возникла на базе проектирования систем более общего вида в отличие от других структурных методов, «выросших» из проектирования программного обеспечения.

Кроме того, SADT имеет развитые процедуры поддержки коллективной работы и готова к применению на ранних стадиях создания системы.

Широкое использование SADT показало, что ее можно сочетать с другими структурными методами. Это достигается использованием графических SADT-описаний в качестве схем, связывающих воедино различные методы, примененные для описания определенных частей системы с различным уровнем детализации.

В последние десятилетия активное развитие получило направление, связанное с разработкой программных средств для поддержки функционального моделирования – CASE-средств. Одно из них подробно рассмотрено в учебном пособии – программный продукт BPWin компании Platinum technologies.

В BPWin реализована технология функционального моделирования IDEF0. Описание процессов ведется в двух вариантах: существующие бизнес-процессы (модель AS-IS) и идеальное положение вещей – того, к чему надо стремиться (модель TO BE). Методология IDEF0 предписывает построение модели в виде совокупности взаимосвязанных диаграмм, отражающих различные уровни абстрагирования. При этом BPWin позволяет на любой ветви модели переключаться на нотацию IDEF3 или DFD и создавать смешанную модель.

На основе модели BPWin можно создать модель данных, для чего Platinum technologies предлагает мощный инструмент – ERWin.

В учебном пособии методика IDEF0 и работа в среде BPWin

рассмотрена на примере моделирования процесса создания конкурентоспособного ДЛА:

Изложенная технология моделирования бизнес-процессов используется в научной работе СГАУ и в учебном процессе для анализа системы управления реальных предприятий аэрокосмической отрасли. Преподавателями и студентами факультета двигателей летательных аппаратов СГАУ, начиная с 2000 года, проведена большая работа и разработан ряд предложений по повышению эффективности управления на ОАО «Моторостроитель» (г. Самара) и ОАО НПО «Сатурн» (г. Рыбинск).

По результатам этой работы защищены более 15 дипломных проектов [10,11]. Выпускники, принимавшие участие в этой работе, успешно работают на промышленных предприятиях различного профиля и занимают руководящие должности.

СПИСОК ЛИТЕРАТУРЫ

1. Добратулин, С.А. Повышение эффективности управления авиадвигательным предприятием на основе информационного моделирования производственных процессов [Текст] / С.А.Добратулин, И.Л.Шитарев, С.А. Шустов // Тезисы докладов Международной научно-технической конференции, посвященной памяти Генерального конструктора аэрокосмической техники академика Н.Д. Кузнецова. – Самара, 2001. - С. 14.
2. Воронин, П.В. SADT–модель реструктуризации крупного промышленного предприятия аэрокосмической отрасли [Текст] / П.В.Воронин, А.В.Коблов, Н.И.Туркин // Тезисы докладов 50-ой студенческой научно-технической конференции. – Самара, 2000. - С. 15.
3. Дегтярев, М.В. Анализ возможностей концептуального моделирования авиадвигательным предприятием на основе методологии IDEFO [Текст] / М.В.Дегтярев, Е.Ю.Минкина, К.В.Онопrienко // Сборник научных работ студентов факультета двигателей летательных аппаратов, выпуск 2. - Самара, 2002 - С. 47-42.
4. Безменова, Н.В. Технология концептуального моделирования бизнес-процессов крупного предприятия аэрокосмической отрасли с помощью CASE-средств [Текст] / Н.В.Безменова, С.А.Добратулин, С.А.Шустов // Тезисы докладов 3-й Международной конференции по неравновесным течениям в соплах и струях, 3...7 июля 2000. – М., 2000. - С. 352-353.
5. Добратулин, С.А. Информационное моделирование внутрипроизводственного планирования производства двигателей летательных аппаратов [Текст] / С.А.Добратулин, С.А.Шустов // Тезисы докладов XI международной конференции «Вычислительная механика и современные программные системы (ВМСППС'2001)», 2-6 июля 2001 г., Москва-Истра. – М., 2001. - С. 175-176.

6. Калянов, Г.Н. CASE-структурный системный анализ [Текст] / Г.Н.Калянов. - М.: Изд-во «ЛЮРИ», 1996 - 242 с.
7. Вендров, А.В. CASE-технологии (современные методы и средства проектирования информационных систем) [Текст] / А.В.Вендров. М.: Финансы и статистика, 1998 - 176 с.
8. Маклаков, С.В. BPWin и ERWin . CASE- средства разработки информационных систем С.В.Маклаков. - М.: ДИАЛОГ-МИФИ, 2000. – 256 с.
9. Марка, Дэвид А. SADT – методология структурного анализа и моделирования [Текст] / Дэвид А. Марка, Клемент Мак Гоуэн. М.: Метатехнология, 1993 - 241 с.
10. Туркин, Н.И. Разработка информационной модели бизнес-планирования применительно к проблеме антикризисного управления (на примере ОАО «Моторостроитель») [Текст]: пояснительная записка к дипломной работе: 15.02.2001 / Н.И.Туркин. - Самара, 2001. - 124 с.
11. Коблов, А.В. Разработка информационной модели логистики в производстве авиационных двигателей применительно к проблеме антикризисного управления ОАО «Моторостроитель» [Текст]: пояснительная записка к дипломной работе: 15.02.2001 / А.В.Коблов. - Самара, 2001. - 121 с.

ПРИЛОЖЕНИЕ А

Словарь терминов

Объект (лат. Objectum предмет) - все то, на что направлена человеческая деятельность. Если объектом является часть реального мира, такой объект называется **реальным объектом**.

ДЛА - двигатель летательного аппарата.

Жизненный цикл (ЖЦ) – взаимосвязанные процессы, имеющие место при создании нового объекта: предпроектные исследования, проектирование, производство, эксплуатация, утилизация.

Проектирование – процесс преобразования исходного описания объекта в виде требований к нему в конечное описание, достаточное для передачи объекта в производство.

Модель (лат. Modulus - мера) – это абстрактный объект-заместитель объекта-оригинала, отражающий наиболее существенные свойства оригинала.

Моделирование - процесс замещения объекта-оригинала его моделью (т.е. процесс получения модели); процесс использования модели для исследования свойств объекта-оригинала.

Системный подход - метод, основанный на представлении знаний об объекте любой природы в виде **системы**.

Система – это множество взаимосвязанных элементов, обладающих одним или несколькими **интегративными свойствами**.

Интегративное свойство - это такое свойство, которым обладает только система в целом, но не обладает ни один из элементов системы.

Структура - множество элементов и связей между ними.

Простая система – это такая система, ни один из элементов которой не имеет структуры.

Сложная система – система, в которой хотя бы один элемент сам является системой (такой элемент называется **подсистемой**).

Целенаправленная система - система, имеющая собственные цели.

Концептуальное моделирование – этап анализа в системном моделировании, на котором определяются требования к создаваемой модели.

IDEF (Integrated Definition language) – интегрированный язык определений.

IDEF0 (Function Modeling) – методология функционального моделирования (поддерживается CASE- средством BPWin).

CASE-средства (Computer Aided Systems/Software Engineering) – компьютерные средства проектирования систем.

SADT (Structured Analysis and Design Technique) - технология структурного анализа и проектирования.

SADT-диаграмма описание объекта с использованием графических элементов SADT-технологии.

Входные дуги - объекты, используемые и преобразуемые функциями.

Управленческие дуги - информация, управляющая действиями функций, указывающая, что и как должна выполнять функция, а также вводящая ограничения на эти функции.

Выходные дуги - объекты, в которые преобразуются входы.

Дуги механизмов – объекты, с помощью которых вход преобразуется в выход.

Контекстная диаграмма SADT-модели - диаграмма, состоящая из одного SADT-блока и его дуг, определяющая границу системы и ее взаимодействия с окружающей средой.

Диаграмма декомпозиции SADT-модели - диаграмма, детализирующая одну из функций системы путем разделения ее на ряд взаимосвязанных подфункций.

Бизнес-процессы – процессы, протекающие в организационно-технических системах.

CALS (Computer Aided and Life cycle Support) - компьютерная поддержка жизненного цикла изделия.

CALS-стандарты - набор методов и средств информационного обеспечения создания сложных технических изделий на основе интеграции данных на всех этапах жизненного цикла.

Учебное издание

*Безменова Наталья Витальевна
Шустов Станислав Алексеевич*

**МЕТОДОЛОГИЯ КОНЦЕПТУАЛЬНОГО
МОДЕЛИРОВАНИЯ IDEF0
В ИНСТРУМЕНТАЛЬНОЙ СРЕДЕ ВРWin
(на примере жизненного цикла двигателя
летательного аппарата)**

Учебное пособие

Редакторская обработка Н. С. Купринова
Корректорская обработка Т. К. Кретьнина
Доверстка А. А. Нечитайло

Подписано в печать 28.12.06. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная.
Усл. печ. л. 3,7. Усл. кр.-отг. 3,8. Уч.-изд.л. 4,0.
Тираж 50 экз. Заказ 185 . ИП-84(2)/2006

Самарский государственный
аэрокосмический университет.
443086 Самара, Московское шоссе, 34.

Изд-во Самарского государственного
аэрокосмического университета.
443086 Самара, Московское шоссе, 34.