

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Кафедра программного обеспечения информационных технологий

**В.В.Бахтизин, Л.А.Глухова**

## **МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО ПРОЕКТИРОВАНИЯ IDEF0**

Учебное пособие  
по курсу  
ТЕХНОЛОГИЯ РАЗРАБОТКИ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

для студентов специальности 40 01 01  
“Программное обеспечение информационных технологий”  
дневной формы обучения

Минск 2003

УДК 681.3 (075.8)  
ББК 32.973.202-018.2 я73  
Б30

Рецензент:

зав. кафедрой информатики Минского государственного высшего радиотехнического колледжа, канд. техн. наук, доц. Ю.А.Скудняков

**Бахтизин В.В.**

Б 30      Методология функционального проектирования IDEF0: Учеб. пособие по курсу «Технология разработки программного обеспечения» для студ. спец. 40 01 01 “Программное обеспечение информационных технологий дневной формы обучения” В.В.Бахтизин, Л.А.Глухова. – Мн.: БГУИР, 2003. – 24 с.: ил.

ISBN 985-444-537-2.

В учебном пособии приведено описание методологии функционального проектирования IDEF0. Даны базовые понятия методологии. Описаны правила построения IDEF0-диаграмм и IDEF0-модели. Рассмотрены стратегии декомпозиции, используемые при IDEF0-моделировании. Описана организация процесса моделирования при использовании IDEF0-методологии. Приведен перечень заданий для лабораторных и индивидуальных работ.

УДК 681.3 (075.8)  
ББК 32.973.202-018.2 я73

ISBN 985-444-537-2

© Бахтизин В.В., Глухова Л.А., 2003  
© БГУИР, 2003

## СОДЕРЖАНИЕ

Введение .....	4
Общие сведения о методологии структурного анализа и проектирования SADT .....	5
Основные понятия IDEF0-модели .....	6
Синтаксис диаграмм .....	8
Синтаксис моделей .....	12
Декомпозиция и ее стратегии при IDEF0-моделировании .....	17
Процесс моделирования в IDEF0 .....	18
Задания для лабораторных и индивидуальных работ .....	21
Контрольные вопросы .....	22
Литература .....	23

Библиотека БГУИР

## ВВЕДЕНИЕ

Как показывают исследования, большинство ошибок вносится в программы на ранних этапах их разработки (на этапах анализа и проектирования) и гораздо меньше их возникает на этапах кодирования и тестирования-отладки.

Как правило, ошибки, возникающие на ранних этапах создания системы, являются следствием неполноты функциональных спецификаций или несогласованности между спецификациями и проектом, выполненным по ним. Очевидно, что основная причина этого кроется в неадекватности используемых методов создания систем поставленным задачам.

В 80-е годы разработан ряд методов структурного проектирования программ, специально предназначенных для использования на ранних этапах процесса разработки сложных систем широкого профиля и позволяющих существенно сократить возможности внесения ошибок в разрабатываемую систему. Наиболее известными и широко используемыми из данных методов являются:

- метод структурного анализа и проектирования SADT Росса;
- методы, ориентированные на потоки данных (методы Йодана, ДеМарко, Гейна, Сарсона);
- методы структурирования данных (методы Джексона-Уорнера, Орра, Чена).

Появление новых методов проектирования поставило задачу создания программного обеспечения (ПО), позволяющего автоматизировать их использование при проектировании больших систем. К середине 80-х годов сформировался рынок программных средств, названных CASE-системами.

Первоначально термин CASE трактовался как Computer Aided Software Engineering (компьютерная поддержка проектирования ПО). В настоящее время данному термину придается более широкий смысл и он расшифровывается как Computer Aided System Engineering (компьютерная поддержка проектирования систем), а современные CASE-средства ориентируются на создание спецификаций, проектирование и моделирование сложных систем широкого назначения.

Наиболее перспективные CASE-продукты базируются на предположении, что программная система – это частный случай системы вообще. Процесс создания ПО хотя и имеет свои особенности, но включает в себя практически те же стадии, что и системы общего назначения.

С середины 70-х годов в США финансировался ряд проектов, ориентированных на разработку методов описания и моделирования сложных систем [1]. Один из них – проект ICAM (Integrated Computer-Aided Manufacturing). Его целью являлась разработка подходов, обеспечивающих

повышение эффективности производства благодаря систематическому внедрению компьютерных технологий. В соответствии с проектом ICAM было разработано семейство методологий IDEF (ICAM DEFinition), которое состоит из трех самостоятельных методологий моделирования различных аспектов функционирования производственной среды или системы:

- IDEF0 – методология создания функциональной модели производственной среды или системы (основана на методе SADT Росса);
- IDEF1 – методология создания информационной модели производственной среды или системы (основана на реляционной теории Кодда и использовании ER-диаграмм Чена);
- IDEF2 – методология создания динамической модели производственной среды или системы.

Позднее в рамках проекта ICAM были начаты работы по созданию технологии объединения в сеть неоднородных вычислительных систем. Одним из практических результатов данных работ стало создание методологии семантического моделирования данных IDEF1X – расширения методологии IDEF1.

Методологии IDEF0 и IDEF1X являются стандартизированными и, будучи независимыми, дают адекватное и достаточно полное представление о сложной системе.

## **ОБЩИЕ СВЕДЕНИЯ О МЕТОДОЛОГИИ СТРУКТУРНОГО АНАЛИЗА И ПРОЕКТИРОВАНИЯ SADT**

Методология SADT (Structured Analysis And Design Technique) [3] сформулирована в общих чертах Дугласом Т.Россом (компания SofTech) около 30 лет назад. На рынке SADT появилась в 1975 г. К 1981 г. SADT уже использовали более чем в 50 компаниях.

Существует два основных направления в SA-моделировании (Structured Analysis–моделировании): *функциональные модели* выделяют события в системе, *модели данных* выделяют объекты (данные) системы, связывающие функции между собой и с их окружением. В обоих случаях используется один и тот же графический язык блоков и дуг (но блоки и дуги меняются ролями).

При наиболее полном моделировании используются взаимодополняющие модели обоих типов (например, методология SADT компании SofTech). Зачастую используется только функциональный вариант данной методологии в правительственной стандартизированной версии, получившей название *IDEF0*.

В коммерческом мире методология SADT широко используется для определения требований к проектируемой системе. Методология SADT, как правило, применяется на ранних этапах процесса создания системы ("жизненного цикла системы"), часто еще до разработки технического задания (ТЗ) или спецификации и специально с этой целью. На более поздних этапах

процесса создания системы следует применение других методов проектирования.

#### **Достоинства методологии SADT:**

- 1) универсальность – SADT может использоваться для проектирования сложных систем любого назначения (например, управление и контроль, аэрокосмическое производство, телефонные сети, учет материально-технических ресурсов и др.), а не только программного обеспечения (ПО);
- 2) SADT – единственная методология, легко отражающая такие системные характеристики, как управление, обратная связь и исполнители;
- 3) SADT имеет развитые процедуры поддержки коллективной работы;
- 4) в отличие от подавляющего большинства других технологий, SADT может быть использована на ранних этапах создания системы (предпроектная стадия);
- 5) SADT может сочетаться с другими структурными методами проектирования.

В данном учебном пособии приведено подмножество полной методологии SADT, ориентированное на функции системы и называемое "функциональным моделированием" (методология IDEF0). Более детально данная методология рассмотрена в [3].

## **ОСНОВНЫЕ ПОНЯТИЯ IDEF0-МОДЕЛИ**

Под *Системой* подразумевается совокупность взаимодействующих компонентов и взаимосвязей между ними.

*Моделированием* называется процесс создания точного описания системы. IDEF0-методология предназначена для создания описания систем и основана на концепциях системного моделирования.

Описание системы с помощью методологии IDEF0 называют *моделью*. В IDEF0-моделях используются как естественный, так и графический языки.

IDEF0-модель дает полное и точное описание, адекватное системе и имеющее конкретное назначение. Назначение описания называют *целью модели*. *Формальное определение модели* в IDEF0 имеет следующий вид [3]:

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

Таким образом, *целью модели* является получение ответов на некоторую совокупность вопросов. Обычно вопросы для IDEF0-модели формируются на самом раннем этапе проектирования (еще нет ТЗ, спецификации и т. п.). Затем основная суть этих вопросов должна быть выражена в одной-двух фразах.

С определением модели тесно связан выбор позиции, с которой наблюдается система и создается ее модель. Методология IDEF0 требует, чтобы модель рассматривалась все время с одной и той же позиции. Эта позиция называется "*точкой зрения*" данной модели. Точку зрения лучше всего представлять как место (позицию) человека или объекта, на которое надо встать, чтобы увидеть систему в действии.

Например, при разработке автоматизированной обучающей системы (АОС) точкой зрения может быть позиция неквалифицированного пользователя, квалифицированного пользователя, программиста и т.п.

### **Пример**

Для облегчения усвоения материала выберем предметную область, близкую и понятную студентам и учащимся. Итак, пусть необходимо разработать IDEF0-модель процесса выполнения лабораторных работ.

На первом этапе проектирования формулируются вопросы к IDEF0-модели, формируется цель модели, определяются претенденты на точку зрения, выбирается точка зрения.

Например, в *перечень вопросов* к IDEF0-модели могут входить такие вопросы:

- Какие этапы необходимо выполнить студенту в ходе лабораторной работы?
- Какие сотрудники участвуют в процессе выполнения студентом лабораторной работы?
- Какие виды работ должен осуществлять преподаватель во время выполнения студентом лабораторной работы?
- Какие виды работ должен осуществлять лаборант во время выполнения студентом лабораторной работы?
- Какое оборудование необходимо для выполнения лабораторных работ?
- Как влияют результаты отдельных этапов на итоги выполнения лабораторной работы?
- В чем заключается защита лабораторной работы?

На основании перечня вопросов формулируется *цель модели*:

Определить основные этапы процесса выполнения лабораторной работы, их влияние друг на друга и на результаты защиты работы с целью обучения студентов методологии IDEF0.

*Претенденты* на точку зрения: преподаватель, студент, лаборант. С учетом цели модели предпочтение следует отдать точке зрения преподавателя, так как она наиболее полно охватывает все этапы лабораторной работы, и только с этой точки зрения можно показать взаимосвязи между отдельными этапами и обязанности участников лабораторной работы.

*Субъектом* моделирования является сама система. Но система не существует изолированно, она связана с окружающей средой. Иногда трудно сказать, где кончается система и начинается среда. Поэтому в методологии IDEF0 подчеркивается необходимость точного определения границ системы,

чтобы избежать включения в модель посторонних субъектов. IDEF0-модель должна иметь единственный субъект.

Таким образом, субъект определяет, что включить в модель, а что исключить из нее. Точка зрения диктует автору модели выбор нужной информации о субъекте и форму ее подачи. Цель становится критерием окончания моделирования.

Конечным результатом моделирования является набор тщательно взаимоувязанных описаний, начиная с описания самого верхнего уровня всей системы и кончая подробным описанием деталей или операций системы. Каждое из таких описаний называется *диаграммой*. IDEF0-модель – это древовидная структура диаграмм, где верхняя диаграмма является наиболее общей, а нижние наиболее детализированы. Каждая из диаграмм какого-либо уровня представляет собой декомпозицию некоторого компонента диаграммы предыдущего уровня.

### **Выводы**

Методология IDEF0 создана специально для представления сложных систем путем построения моделей. IDEF0-модель – это описание системы, в котором 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 IDEF0. Описание модели IDEF0 организовано в виде иерархии взаимосвязанных диаграмм. Вершина этой древовидной структуры представляет самое общее описание системы, а ее основание состоит из наиболее детализированных описаний.

## **СИНТАКСИС ДИАГРАММ**

Диаграмма является основным рабочим элементом при создании модели. Каждая IDEF0-диаграмма содержит блоки (работы) и дуги (стрелки). Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Функциональные блоки на диаграмме изображаются прямоугольниками (рис.1). Блок представляет функцию или активную часть системы (в последнем случае блок обозначается с помощью буквы А в его номере).

Каждая сторона блока имеет определенное назначение. Левая сторона предназначена для входов, верхняя - для управления, правая – для выходов, нижняя – для механизмов.

В основе методологии IDEF0 лежат следующие *правила*:

- 1) функциональный блок преобразует *входы* в *выходы*;
- 2) *управление* ограничивает или предписывает условия выполнения преобразований;



3) *механизмы* показывают, кто, что и как выполняет эти преобразования (т.е. механизмы непосредственно осуществляют эти преобразования).

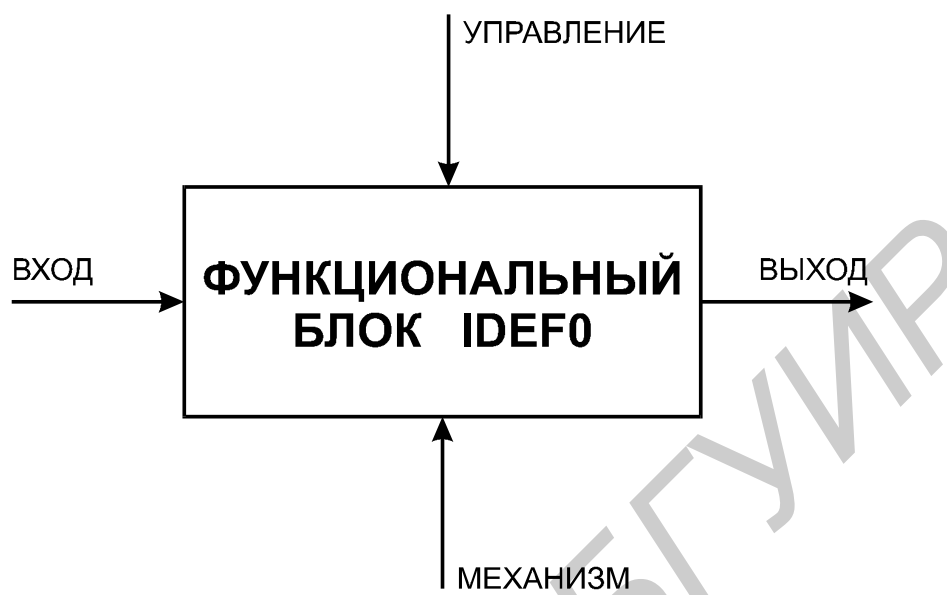


Рис.1. Основная конструкция IDEF0-модели

Рассмотрим синтаксис IDEF0-диаграмм на примере IDEF0-диаграммы, содержащей основные этапы процесса выполнения лабораторной работы. Данная диаграмма приведена на рис.2.

Название IDEF0-блока основано на использовании отглагольного существительного, обозначающего действие (вычисление того-то, определение того-то, обработка того-то и т.д.). На рис.2 блоки имеют названия «Изучение теории», «Ответы на контрольные вопросы», «Выполнение индивидуального задания», «Написание отчета», «Защита лабораторной работы».

Методология IDEF0 требует, чтобы в диаграмме было не менее *трех* и не более *шести* блоков. Это ограничение поддерживает сложность диаграмм на уровне, доступном для чтения, понимания и использования.

Блоки на IDEF0-диаграмме размещаются по степени важности. В IDEF0 этот относительный порядок называется *доминированием*. Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы.

В методологии IDEF0 принято располагать блоки по диагонали диаграммы. Наиболее доминирующий блок обычно размещается в левом верхнем углу диаграммы, наименее доминирующий – в правом нижнем углу. Таким образом, топология диаграмм показывает, какие функции оказывают большее влияние на остальные.

Блоки на IDEF0-диаграмме должны быть пронумерованы. Нумерация блоков выполняется в соответствии с порядком их доминирования (1 – наибольшее доминирование, 2 – следующее и т.д.). Порядок доминирования (номер блока) располагается в правом нижнем углу функционального блока.

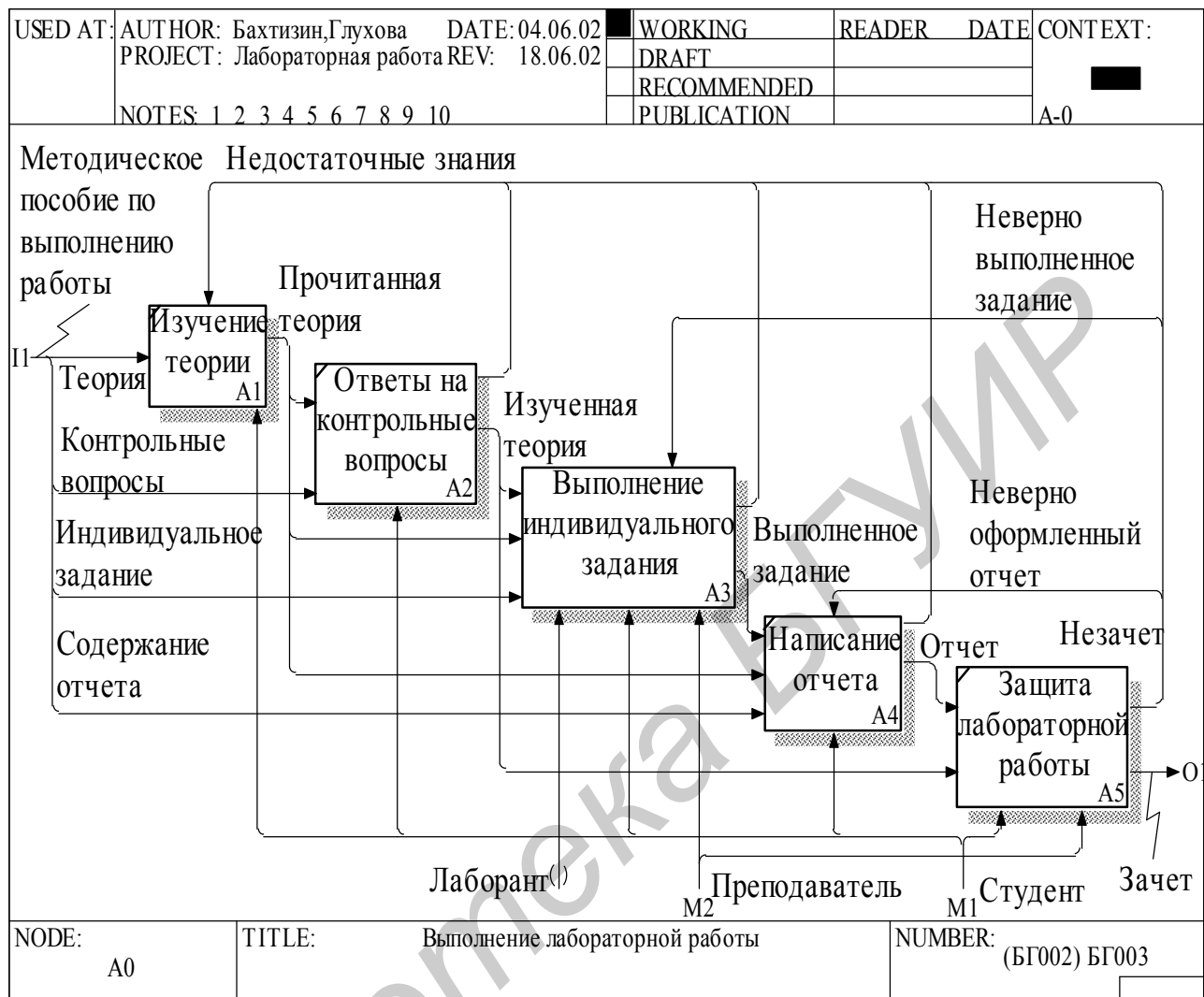


Рис.2. Стандартный IDEF0-бланк и IDEF0-диаграмма, содержащая основные этапы процесса выполнения лабораторной работы

Дуги на IDEF0-диаграмме изображаются линиями со стрелками. Для функциональных IDEF0-диаграмм дуга представляет множество объектов. Под *объектом* в общем случае понимаются некоторые данные (планы, машины, информация, данные в компьютерах). Основу названия дуги на IDEF0-диаграммах составляют существительные. Названия дуг называются *метками*.

Например, на диаграмме, изображенной на рис.2, дуги имеют названия "Индивидуальное задание", "Выполненное задание", "Отчет" и т.д.

Между объектами и дугами возможны *четыре вида отношений*: вход, управление, выход, механизм.

Таким образом, дуги 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 выполнить другая функция.

Следовательно, IDEF0-диаграммы являются предписывающими диаграммами, представляющими входные-выходные преобразования и указывающими правила этих преобразований.

В методологии IDEF0 используется *пять типов взаимосвязей* между блоками для описания их отношений: управление, вход, обратная связь по управлению, обратная связь по входу, "выход-механизм" [3].

*Отношение управления* возникает тогда, когда выход одного блока непосредственно влияет на работу блока с меньшим доминированием.

*Отношение входа* возникает тогда, когда выход одного блока становится входом для блока с меньшим доминированием.

Обратные связи по управлению и по входу представляют собой итерацию или рекурсию.

*Обратная связь по управлению* возникает тогда, когда выход некоторого блока влияет на работу блока с большим доминированием.

*Обратная связь по входу* имеет место тогда, когда выход одного блока становится входом другого блока с большим доминированием.

*Связь "выход-механизм"* встречается нечасто и отражает ситуацию, при которой выход одной функции становится средством достижения цели для другой функции. Данная связь характерна при распределении источников ресурсов (например, физическое пространство, оборудование, финансирование, материалы, инструменты, обученный персонал и т.п.).

Дуга в IDEF0 редко изображает один объект. Обычно она символизирует набор объектов. Поэтому дуги могут разъединяться и соединяться.

*Разветвления дуг* обозначают, что все содержимое дуг или его часть может появиться в каждом ответвлении дуги. Дуга всегда помечается до разветвления, чтобы дать название всему набору. 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;
- каждая метка ветви указывает, что именно содержит ветвь.

При *слиянии дуг* результирующая дуга всегда помечается для указания нового набора объектов, возникшего после объединения. Каждая ветвь перед слиянием помечается или нет в соответствии со следующими *правилами*:

- непомеченные ветви содержат все объекты, указанные в общей метке дуги после слияния;
- метка ветви указывает, что конкретно содержит ветвь.

Разветвления дуг и их соединения – это синтаксис, который позволяет описывать декомпозицию (разделение на структурные части) содержимого дуг. Разветвляющиеся и соединяющиеся дуги отражают иерархию объектов, представленных этими дугами. Из отдельной диаграммы редко можно понять

полную иерархию дуги. Для этого требуется чтение большей части модели. Поэтому методология IDEF0 предусматривает дополнительное описание полной иерархии объектов системы посредством формирования *гlossария* для каждой диаграммы модели и объединения этих гlossариев в *Словарь данных*. Таким образом, Словарь данных – это основное хранилище полной иерархии объектов системы.

Для систематизации информации о диаграммах и модели в целом используется стандартный IDEF0-бланк (см. рис.2). Каждое поле бланка имеет конкретное назначение и заполняется по определенным правилам. Основные из этих полей будут описаны ниже.

При создании IDEF0-модели одна и та же диаграмма может перечерчиваться несколько раз, что приводит к появлению различных ее вариантов. Чтобы различать их, в методологии SADT используется *схема контроля конфигурации диаграмм*, основанная на *хронологических номерах*, или *С-номерах*. С-номерные коды образуются из инициалов автора (авторов) и последовательных номеров. Эти коды записываются в нижнем правом углу IDEF0-бланка (БГ003 на рис.2).

Если диаграмма заменяет более старый вариант, предыдущий С-номер помещается в скобках (например, БГ002 на рис.2). Каждый автор проекта IDEF0 ведет реестр (список) всех созданных им диаграмм, нумеруя их последовательными целыми числами. Для этого используется специальный *бланк реестра С-номеров IDEF0*.

### **Выводы**

Основой IDEF0-диаграмм является блок. Каждая сторона блока имеет определенное назначение (вход, управление, выход, механизм). IDEF0-диаграмма содержит 3-6 блоков, связанных дугами, и может иметь несколько версий. Чтобы различить данные версии, используются С-номера. Блоки на диаграмме представляют функции моделируемой системы, дуги - множество различных объектов системы. Блоки изображаются на диаграмме в соответствии с порядком их доминирования. Дуги могут разветвляться и объединяться различными способами.

## **СИНТАКСИС МОДЕЛЕЙ**

Диаграммы, собранные и связанные вместе, представляют собой IDEF0-модель проектируемой или анализируемой системы. В методологии IDEF0 дополнительно к правилам синтаксиса диаграмм существуют правила синтаксиса моделей. Синтаксис IDEF0-моделей позволяет автору проекта определить границу модели, связать диаграммы в одно целое и обеспечить точное согласование между диаграммами.

IDEF0-модель – это иерархически организованная совокупность диаграмм. Диаграмма содержит 3-6 блоков. Каждый из блоков потенциально может быть детализирован на другой диаграмме. Разделение блока на его структурные части (блоки и дуги) называется *декомпозицией*.

Декомпозиция формирует границы, то есть блок и касающиеся его дуги определяют точную границу диаграммы, представляющей декомпозицию этого блока. Эта диаграмма называется *диаграммой-потомком*. Декомпозируемый блок называется *родительским блоком*, а содержащая его диаграмма – *родительской диаграммой*. Название диаграммы-потомка совпадает с функцией родительского блока.

Таким образом, IDEF0-диаграмма является декомпозицией некоторой ограниченной функции (субъекта). Принцип ограничения субъекта встречается на каждом уровне.

Один блок и несколько дуг на самом верхнем уровне используются для определения границы всей системы. Этот блок описывает общую функцию, выполняемую системой. Дуги, касающиеся этого блока, описывают главные управления, входы, выходы и механизмы этой системы. Диаграмма, состоящая из одного блока и его дуг, определяющая границу системы, называется *контекстной диаграммой модели*. Все, что лежит внутри блока, является частью описываемой системы, а все, лежащее вне его, образует *среду системы*.

На рис.3 приведена контекстная диаграмма процесса выполнения лабораторной работы.

Общая функция модели записывается на контекстной диаграмме в виде названия блока (для рассматриваемого процесса – выполнение лабораторной работы). Блок самого верхнего уровня модели всегда нумеруется нулем.

С контекстной диаграммой связывается цель модели и точка зрения.

Декомпозицией контекстной диаграммы (ее диаграммой-потомком) является диаграмма, приведенная на рис.2.

Название (поле TITLE IDEF0-бланка) диаграммы декомпозиции совпадает с названием декомпозируемого блока родительской диаграммы. Для диаграмм, изображенных на рис.2 и 3, таким названием является «Выполнение лабораторной работы».

Название контекстной диаграммы определяется общей функцией моделируемой системы, то есть совпадает с названием блока контекстной диаграммы (см. рис.3).

Таким образом, две диаграммы IDEF0-модели имеют одно и то же название. Это контекстная диаграмма и ее диаграмма-потомок. Названия всех остальных диаграмм модели уникальны.

Каждая диаграмма модели идентифицируется *"номером узла"* (NODE), расположенным на IDEF0-бланке в левом нижнем углу. Номер узла для контекстной диаграммы имеет следующий вид: заглавная буква А (Activity в функциональных диаграммах), дефис и ноль (см. рис.3).

Номер узла диаграммы, декомпозирующей контекстную диаграмму, – тот же номер узла, но без дефиса (A0, см. рис.2).

Все другие номера узлов образуются посредством добавления к номеру узла родительской диаграммы номера декомпозируемого блока. Например, номер узла родительской диаграммы – А0. Тогда номер узла диаграммы, декомпозирующей первый блок родительской диаграммы, – А01. Первый ноль при образовании номера узла принято опускать, то есть номер узла запишется в виде А1. При декомпозиции третьего блока родительской диаграммы А1 номер узла диаграммы-потомка будет соответствовать значению А13.

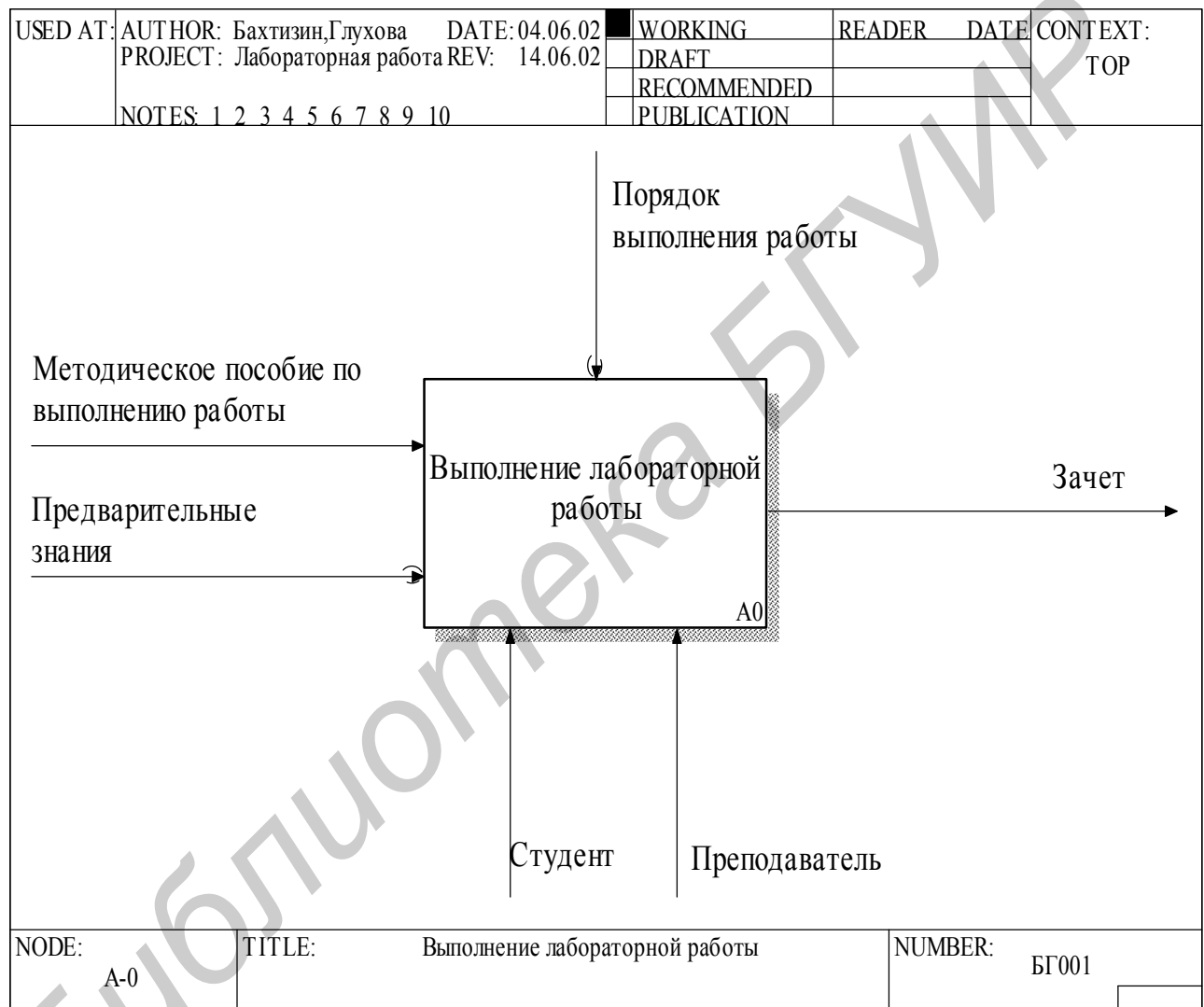


Рис.3. Контекстная диаграмма процесса выполнения лабораторной работы

Для связи диаграмм при движении вверх по иерархии модели могут применяться С-номера или номера узлов. После декомпозиции родительского блока на диаграмме-потомке формируется ссылка на родительскую диаграмму. Для этого используется поле "контекст" (CONTEXT), расположенное в правом верхнем углу IDEF0-бланка. В данном поле маленькими квадратиками изображается каждый блок родительской диаграммы (с сохранением их

относительного положения), заштриховывается квадратик декомпозированного блока и размещается С-номер или номер узла родительской диаграммы.

Например, на диаграмме-потомке, декомпозирующей третий блок А3 родительской диаграммы А0, заполнение поля "контекст" будет соответствовать приведенному на рис.4.

Связь между диаграммами посредством С-номеров или номеров узлов позволяет осуществлять тщательный контроль за введением новых диаграмм в иерархию модели.

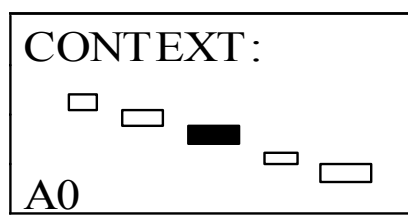


Рис.4. Заполнение поля "КОНТЕКСТ" диаграммы-потомка

IDEF0-диаграммы имеют *внешние дуги* – дуги, выходящие наружу и ведущие к краю страницы. Эти дуги являются интерфейсом между диаграммой и остальной частью модели. Диаграмма должна быть состыкована со своей родительской диаграммой, то есть внешние дуги должны быть согласованы по числу и наименованию с дугами, касающимися декомпозированного блока родительской диаграммы (граничными дугами).

В IDEF0 принята система обозначений, позволяющая аналитику точно идентифицировать и проверять связи по дугам между диаграммами. Эта схема кодирования дуг называется *ICOM* (Input-Control-Output-Mechanism).

*Правила стыковки и обозначения внешних дуг* диаграммы-потомка с граничными дугами родительского блока могут быть сформулированы следующим образом [3]:

- зрительно соединяется каждая внешняя дуга диаграммы-потомка с соответствующей граничной дугой родительского блока;
- каждой зрительной связи присваивается код (I – для входных дуг, C – для связей между дугами управления, O – для связей между выходными дугами, M – для связей между дугами механизма, см. рис.2);
- после каждой буквы добавляется цифра, соответствующая положению данной дуги среди других дуг того же типа, касающихся родительского блока. Входные и выходные дуги пересчитываются сверху вниз, а дуги управлений и механизмов пересчитываются слева направо (в том порядке, как они расположены на родительской диаграмме по отношению к родительскому блоку). Например, на рис.2 внешние дуги M1, M2 пронумерованы в соответствии с расположением граничных дуг на родительском блоке (см. рис.3).

Особые ситуации возникают, когда дуги "входят в тоннель" между диаграммами.

Дуга *"входит в тоннель"*, если:

1) она является внешней дугой, которая отсутствует на родительской диаграмме (дуга имеет скрытый источник);

2) она касается блока, но не появляется на диаграмме, которая его декомпозирует (имеет скрытый приемник).

Тоннельные дуги от скрытого источника начинаются скобками, чтобы указать, что эти дуги идут из какой-то другой части модели, прямо извне модели или они не важны для родительской диаграммы и поэтому на ней не изображаются. Например, на рис.2 дуга механизма «Лаборант» является тоннельной дугой со скрытым источником. Ее нет на родительской диаграмме (см. рис.3), поскольку для родительской диаграммы данная дуга является маловажной.

Тоннельные дуги со скрытым приемником заканчиваются скобками, чтобы отразить тот факт, что такая дуга идет к какой-то другой части модели, выходит из нее или не будет более в этой модели рассматриваться. Тоннельные дуги со скрытым приемником часто используются в том случае, если данные дуги должны связываться с каждым блоком диаграммы-потомка. Изображение таких дуг может существенно загромождать данную диаграмму. Например, дуги «Предварительные знания» и «Порядок выполнения работы» (см. рис.3) должны связываться с каждым блоком диаграммы декомпозиции. Их изображение на диаграмме-потомке является малоинформативным. Поэтому данные дуги реализованы как тоннельные дуги со скрытым приемником и на диаграмме декомпозиции, изображенной на рис.2, не показаны.

Таким образом, "Вхождение дуг в тоннель" используется, как правило, для упрощения описания системы – тогда, когда диаграммы в модели становятся слишком трудными для чтения и понимания.

### **Выводы**

IDEF0-диаграммы являются декомпозициями ограниченных субъектов. Субъект ограничивается блоком и касающимися его дугами. Диаграмма, содержащая границу, называется родительской диаграммой. Диаграмма, декомпозирующая блок родительской диаграммы, называется диаграммой-потомком. Для связывания родительской диаграммы и диаграммы-потомка используются С-номера, номера узлов и коды ICOM. Номер узла идентифицирует уровень данной диаграммы в иерархии модели. Для упрощения описания системы используется специальный технический прием "вхождение дуг в тоннель".



## ДЕКОМПОЗИЦИЯ И ЕЕ СТРАТЕГИИ ПРИ IDEF0-МОДЕЛИРОВАНИИ

Декомпозиция – это процесс создания диаграммы, детализирующей определенный блок и связанные с ним дуги. IDEF0-декомпозиция включает в себя анализ (начальное разделение элемента на более мелкие части) и синтез (последующее их соединение для более детального описания элемента) [3].

Следуя правилам IDEF0, автор производит *вначале* анализ и синтез системных объектов (напомним, что под объектом в общем случае понимаются некоторые данные, изображаемые на модели дугами). Так, список данных начинается со всех граничных дуг и их ICOM- кодов, а заканчивается их составляющими. После этого некоторые составляющие могут быть объединены для выделения объектов, которые будут выступать в качестве управляющих.

*Затем* автор выполняет подобный анализ и синтез функций системы, делая это в соответствии со списком данных. В процессе объединения и введения новых управляющих дуг создается список функций для дальнейшей детализации. Эти функциональные части объединяются в наборы из трех-шести блоков.

После определения функциональных частей список данных и список функций используются для чернового варианта диаграммы. Снова выполняются анализ и синтез, в результате чего формируются наборы объектов, которые представляются дугами, соединяющими блоки.

Такая последовательность выполнения декомпозиции имеет большое значение, поскольку в методологии IDEF0 анализ объектов системы оказывает важнейшее влияние на анализ функций.

В процессе создания диаграмм опытный разработчик постоянно следит за стратегией декомпозиции и ее влиянием на качество модели. При создании IDEF0-модели наиболее часто используются следующие *стратегии декомпозиции*:

1) функциональная стратегия. Базируется на функциональных взаимоотношениях действий системы. Рекомендуется следовать этой стратегии всегда, когда это возможно;

2) декомпозиция в соответствии с функциями, выполняемыми людьми или организациями. Рекомендуется использовать эту стратегию только в начале работы (так как позже взаимосвязи между исполнителями могут быть очень сложны) над моделью системы, относящейся к разряду P3 (people – люди, paper – бумага, procedures – процедуры). Это помогает собрать исходную информацию о системе. Затем можно создавать более обоснованную функциональную декомпозицию в соответствии с первой стратегией;

3) декомпозиция в соответствии с уже известными стабильными подсистемами. Это приводит к созданию набора моделей - по одной модели на каждую подсистему или важную компоненту. Затем для описания всей системы строится составная модель, объединяющая все отдельные модели. Стратегия

эффективна для систем команд и управления, когда разделение на основные части системы не меняется. Например, создается модель отдельно для торпеды, отдельно для защиты от торпед и отдельно для движения подводной лодки. Затем эти модели объединяются вместе для описания способов защиты подводной лодки;

4) стратегия декомпозиции, основанная на отслеживании "жизненного цикла" для ключевых входов системы. Эффективна для моделирования систем, непрерывно преобразующих свои входы в конечный продукт (например, система очистки нефти). Декомпозиция осуществляется в соответствии с этапами преобразования входа (этапами жизненного цикла);

5) декомпозиция по физическому процессу, основанная на выделении функциональных стадий, этапов завершения, шагов выполнения и т.п. Результатом стратегии часто является последовательное описание системы, не учитывающее ограничения, накладываемые функциями друг на друга. Поэтому эту стратегию рекомендуется использовать, только если целью модели является описание физического процесса или в крайнем случае, когда разработчик IDEF0-модели не понимает, как действовать.

### ***Выводы***

Для построения модели используется процесс декомпозиции. Декомпозиция включает этапы анализа и синтеза. Существуют различные стратегии декомпозиции. Их выбор и получение диаграммы A0 высокого качества часто требуют нескольких итераций и многих изменений. Декомпозиция прекращается, когда модель достаточно точна, чтобы отвечать на вопросы, составляющие ее цель.

## **ПРОЦЕСС МОДЕЛИРОВАНИЯ В IDEF0**

Процесс моделирования в IDEF0 включает сбор информации об исследуемой области, документирование полученной информации с представлением ее в виде модели и уточнение модели посредством итеративного рецензирования [3].

На рис.5 изображен процесс моделирования в IDEF0, описанный с помощью IDEF0-диаграммы. Процесс моделирования в IDEF0 является итерационным, что приводит к точному описанию системы. В основе высокой эффективности процесса IDEF0-моделирования лежит разделение функций, выполняемых участниками IDEF0-проектов (см. входы механизмов на рис.5). Это один из примеров поддержки коллективной работы в IDEF0: эксперты являются источниками информации, авторы создают диаграммы и модели, библиотекарь координирует обмен письменной информацией, читатели рецензируют модели, комитет технического контроля принимает и утверждает модель.

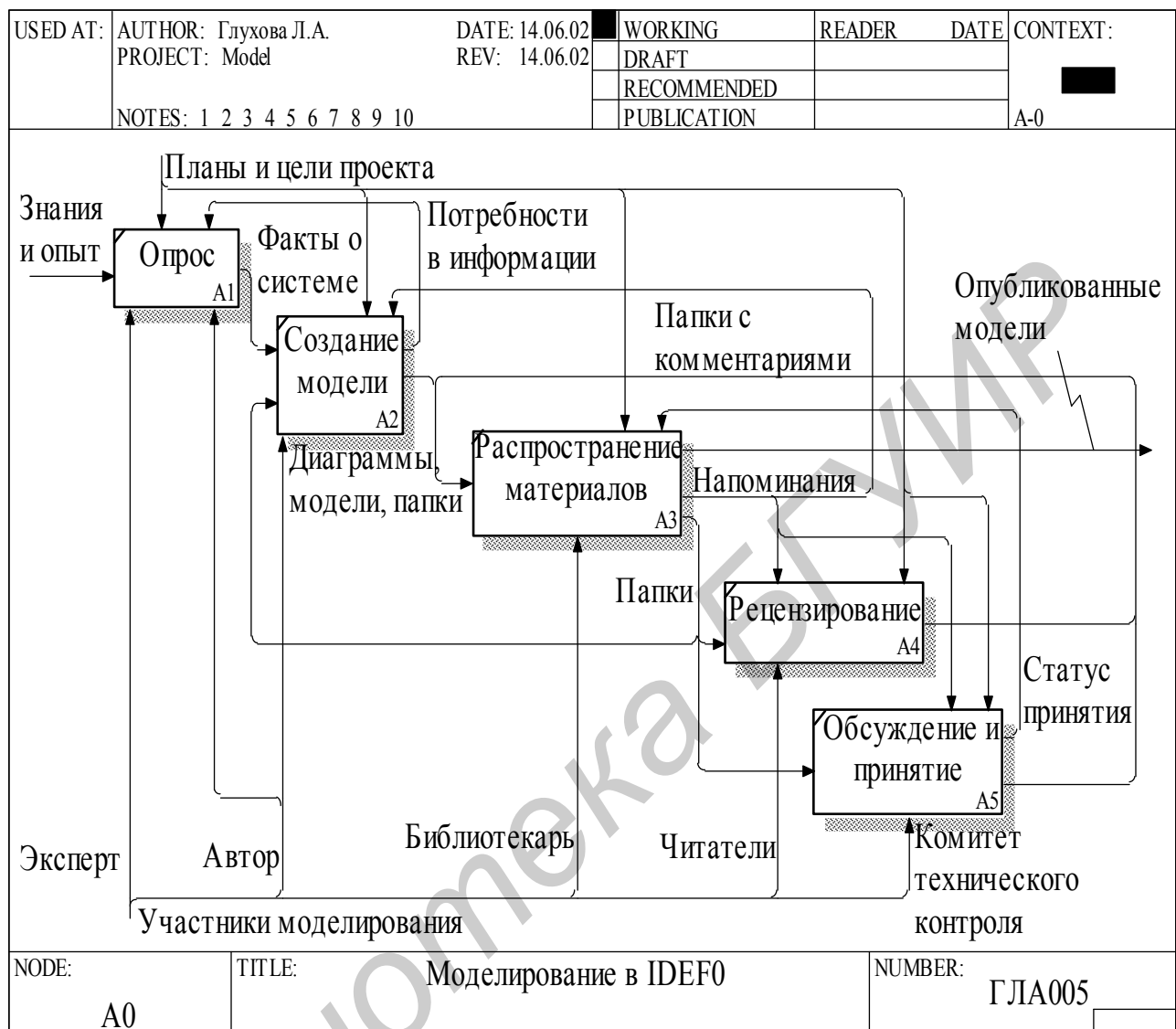


Рис.5. Процесс моделирования в IDEF0

Целью первого этапа IDEF0-моделирования (блок A1 "Опрос") является получение знаний о моделируемой системе (о предметной области). Для этого могут быть использованы различные источники информации: чтение документов, опрос экспертов, наблюдение за работой системы и т. п. Во время опроса графический язык IDEF0 используется как средство для заметок, которые служат основой для построения диаграмм.

Вторым этапом моделирования является создание модели (блок A2). На данном этапе аналитик документирует полученные им знания о данной проблемной области, представляя их в виде одной или нескольких IDEF0-диаграмм. Процесс создания модели осуществляется с помощью специального метода детализации ограниченного субъекта.

При его использовании автор (аналитик) *вначале* анализирует объекты

(информация, данные, механизмы и т.п.), входящие в систему, а *затем* использует полученные знания для анализа *функций* системы. На основе этого анализа создается диаграмма, в которой объединяются сходные объекты и функции. Этот путь проведения анализа системы и документирования его результатов является уникальной особенностью методологии IDEF0.

Создающиеся IDEF0-модели проходят через серию последовательных улучшений до тех пор, пока они в точности не будут представлять реальную предметную область.

Одним из основных компонентов методологии IDEF0 является *итеративное рецензирование*. В процессе него автор и эксперт многократно совещаются относительно достоверности создаваемой модели. Итеративное рецензирование называется *циклом автор/читатель*. Данный цикл начинается, когда автор распространяет информацию о какой-либо части своей работы с целью получения отзыва о ней. Материалы для распространения оформляются в виде "папок" – небольших пакетов с результатами работы. Данные результаты критически обсуждаются другими специалистами (в IDEF0 они называются читателями) в течение определенного времени. Сделанные замечания помещаются в папку в виде нумерованных комментариев. К определенному сроку замечания поступают к автору. Автор отвечает на каждое замечание и обобщает критику, содержащуюся в замечаниях.

Таким образом, методология IDEF0 поддерживает как асинхронный, так и параллельный просмотр модели. Это является наиболее эффективным способом распределения работы в коллективе. На практике над различными частями модели работает совместно несколько авторов, так как каждый функциональный блок модели представляет отдельный компонент, который может быть независимо проанализирован и декомпозирован.

На рис.5 цикл автор/читатель представлен блоками А3 «Распространение материалов» и А4 «Рецензирование».

Для эффективного моделирования важнейшее значение имеет организация своевременной обратной связи между участниками IDEF0-проекта, так как устаревшая информация способна свести на нет все усилия по разработке системы. Поэтому IDEF0 выделяет специальную роль *наблюдателя за процессом рецензирования*. Эту роль выполняет так называемый библиотекарь, который является главным координатором процесса моделирования в IDEF0. Он обеспечивает своевременное и согласованное распространение рабочих материалов, контролирует их движение.

При IDEF0-моделировании выделяется специальная группа людей, которые отвечают за то, что создаваемая в процессе анализа модель будет точна и используется в дальнейшем, за контроль качества модели, за соответствие выполняемой работы конечным целям всего проекта. Эта группа называется *Комитетом технического контроля*. Если модель признана Комитетом применимой, она публикуется. В противном случае авторам направляются замечания для необходимой доработки. Данная функция процесса

моделирования на рис.5 представлена блоком А5 «Обсуждение и принятие».

В настоящее время существует ряд современных CASE-средств, поддерживающих технологию IDEF0. Среди наиболее дешевых и доступных на нашем рынке программных средств следует отметить CASE-средство VPwin. Процесс функционального моделирования в среде VPwin подробно описывается в [2].

### **Выводы**

IDEF0 – это методология, объединяющая процесс моделирования, управление конфигурацией проекта, использование дополнительных языковых средств и руководство проектом со своим графическим языком. Процесс IDEF0-моделирования может быть разделен на несколько этапов: опрос экспертов, создание диаграмм и моделей, распространение документации, оценка адекватности моделей, принятие их для дальнейшего использования. Каждый из исполнителей проекта выполняет конкретные обязанности. Ряд современных CASE-средств поддерживают технологию IDEF0-моделирования.

## **ЗАДАНИЯ ДЛЯ ЛАБОРАТОРНЫХ И ИНДИВИДУАЛЬНЫХ РАБОТ**

С помощью CASE-средства VPwin разработать иерархические функциональные модели следующих предметных областей:

- 1) библиотечный каталог;
- 2) отдел кадров университета;
- 3) студенты университета;
- 4) деканат (сотрудники и студенты факультета);
- 5) расписание занятий преподавателей кафедры;
- 6) расписание занятий студентов университета;
- 7) результаты экзаменационных сессий студентов за весь период обучения;
- 8) управление университетом;
- 9) общежитие;
- 10) школа;
- 11) делопроизводство предприятия;
- 12) делопроизводство профкома организации;
- 13) справочная служба предприятия (завода);
- 14) учет материальных ценностей (склад);
- 15) магазины города;
- 16) общественный транспорт города;
- 17) клинические больницы города;
- 18) предприятия города;
- 19) работа предприятия;

- 20) справочная служба аптек;
- 21) справочная служба поликлиники;
- 22) справочная бытуслуг;
- 23) бронирование места на ж/д транспорте;
- 24) домоуправление;
- 25) бухгалтерия предприятия;
- 26) учет налогоплательщиков;
- 27) оплата услуг телефонной сети;
- 28) банковские услуги.

Модель должна включать в себя не менее трех уровней иерархии.

## **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Понятие Case-средств и их назначение.
2. Семейство методологий IDEF. Их назначение.
3. Направления SADT-моделирования.
4. Этапы жизненного цикла программных средств, для которых наиболее эффективно использование методологии SADT.
5. Достоинства методологии SADT.
6. Понятие системы и модели в IDEF0.
7. Формальное определение модели в IDEF0.
8. Цель модели в IDEF0.
9. "Точка зрения" модели в IDEF0.
10. Субъект моделирования в IDEF0. Принцип ограничения субъекта.
11. Объекты и функции в IDEF0.
12. Правила представления функциональных блоков на IDEF0-диаграмме.
13. Назначения сторон функциональных блоков на IDEF0-диаграмме.
14. Принцип доминирования и его представление на IDEF0-диаграмме.
15. Назначение дуг на IDEF0-диаграмме.
16. Описание дуг на IDEF0-диаграмме.
17. Виды отношений между объектами и дугами на IDEF0-диаграмме.
18. Типы взаимосвязей между блоками на IDEF0-диаграмме.
19. Разветвления дуг и правила их обозначения на IDEF0-диаграмме.
20. Слияние дуг и правила их обозначения на IDEF0-диаграмме.
21. Глоссарии и Словарь данных.
22. С-номера. Назначение и правила записи.
23. Бланк реестра С-номеров IDEF0.
24. Понятие диаграммы-потомка, родительского блока, родительской диаграммы в IDEF0-модели.
25. Контекстная диаграмма модели.
26. Номер узла IDEF0-диаграммы. Назначение и правила записи.

27. Организация связей между IDEF0-диаграммами.
28. Правила заполнения поля "КОНТЕКСТ" IDEF0-диаграммы.
29. Внешние дуги IDEF0-диаграммы и система их обозначений.
30. Правила стыковки и обозначения внешних дуг диаграммы-потомка с граничными дугами родительского блока.
31. «Вхождение дуги в тоннель». Назначение и правила обозначения.
32. IDEF0-декомпозиция. Определение, составные части, последовательность применения в IDEF0.
33. Стратегии декомпозиции в IDEF0.
34. Основные этапы процесса моделирования в IDEF0.
35. Назначение цикла автор/читатель в процессе моделирования в IDEF0.
36. Роль «библиотекаря» в процессе моделирования в IDEF0.
37. Роль «Комитета технического контроля» в процессе моделирования в IDEF0.
38. CASE-средства, поддерживающие технологию IDEF0-моделирования.

## **ЛИТЕРАТУРА**

1. Каменнова М.С. Системный подход к проектированию сложных систем // Журнал доктора Добба. – 1993. - № 1. – С.9-14.
2. Маклаков С.В. VPwin и ERwin. CASE-средства разработки информационных систем. – М., 1999.
3. Марка Д.А., МакГоуэн К. Методология структурного анализа и проектирования SADT. – М., 1993.

Учебное издание

**Бахтизин Вячеслав Вениаминович,  
Глухова Лилия Александровна**

**МЕТОДОЛОГИЯ  
ФУНКЦИОНАЛЬНОГО ПРОЕКТИРОВАНИЯ  
IDEF0**

Учебное пособие

по курсу

**ТЕХНОЛОГИЯ РАЗРАБОТКИ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

для студентов специальности 40 01 01

“Программное обеспечение информационных технологий”  
дневной формы обучения

Редактор Т.Н.Крюкова  
Корректор Е.Н.Батурчик

---

Подписано в печать 06.10.2003.

Формат 60x84 1/16.

Бумага офсетная.

Печать ризографическая.

Гарнитура «Таймс».

Усл.печ.л. 1,51.

Уч.-изд.л. 1,5.

Тираж 100 экз.

Заказ 227.

---

Издатель и полиграфическое исполнение:

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Лицензия ЛП N 156 от 30.12.2002.

Лицензия ЛВ N 509 от 03.08.2001.

220013, Минск, П. Бровка, 6.