

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЁВА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)» (СГАУ)

ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ НА БАЗЕ СТАНДАРТА IDEF0

Рекомендовано редакционно-издательским советом федерального государственного автономного образовательного учреждения высшего образования «Самарский государственный аэрокосмический университет имени академика С.П. Королёва (национальный исследовательский университет)» в качестве методических указаний

САМАРА
Издательство СГАУ
2014

УДК СГАУ: 6(075)

Рецензент канд. техн. наук, доцент. И. В. П о т а п о в

Функциональное моделирование на базе стандарта IDEF0: метод. указания / сост. Д.Ю. Киселев, Ю.В. Киселев, А.В. Вавилин. – Самара: Изд-во СГАУ, 2014. – 20 с.

Включают в себя основные сведения о методологии графического структурного анализа, описывающей внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ, а также описание существующих структур организации потоков данных и информацию, необходимую для работы с IDEF0.

Предназначены для студентов специальностей и направлений подготовки «Техническая эксплуатация летательных аппаратов и двигателей» и «Техническая эксплуатация авиационных электросистем и пилотажно-навигационных комплексов».

Разработаны на кафедре ЭАТ.

УДК СГАУ: 6(075)

ВВЕДЕНИЕ

В настоящее время происходит постоянное усложнение производственно-технических и организационно-экономических систем (фирм, предприятий, производств и других субъектов хозяйственной деятельности). Совершенствование функционирования и повышение эффективности этих систем возможно только в результате некоей деятельности, при которой производится анализ этих систем. Такая деятельность называется ре-инжинирингом деловых процессов.

Ре-инжиниринг деловых процессов – это системный подход, ориентированный на достижение существенного измеряемого увеличения продуктивности и эффективности деятельности предприятия посредством кардинального пересмотра, переосмысления и ре-проектирования его ключевых деловых процессов. Естественно, что проведение анализа требует наличие специальных средств описания ре-проектируемых деловых процессов.

В конце 50-х годов прошлого столетия Дуглас Т. Росс впервые использовал понятие «SA-блок», которое легло в основу того, что позже было названо структурным анализом (*Structured Analysis – SA*) в отчете по созданию алгоритмического языка АРТ в Массачусетском технологическом институте (МТИ). SA-блок был использован Дуглас Т. Россом в 1967 году в отчете «AED-подход к системам автоматизированного проектирования», отмеченном премией.

1960 г. – начало разработки того, что теперь называется «иерархическая декомпозиция сверху вниз».

1969 г. – основание компании SofTech.

1972 г. – в ходе коммерческого проекта идеи и опыт были обобщены в методологию проектирования.

1973 г. – создание первого руководства автора для обучения аналитиков методу (*Architecture Method*) и примененному в работах по проекту в ВВС США при разработке систем автоматизированного производства.

1974 г. – методология была названа SADT (*Structural Analysis and Design Technique*) «Структурный анализ и проектирование».

Программа *Integrated Computer-Aided Manufacturing* (ICAM), предназначенная для интегрированной компьютеризации производства (США, конец 70-х гг. XX века) выявила потребность в более совершенных способах обмена информацией и методах анализа производственных систем для всех специалистов, занимающихся проблемой. В рамках программы ICAM для удовлетворения этих потребностей была разработана методология IDEF (**ICAM Definition**). Наиболее популярными стали следующие части методологии.

IDEF0 – используется для создания **функциональных** моделей (Integration Definition For Function Modeling), отражающих структурированное изображение функций производственной системы или среды, а также информации и объектов, связывающих эти функции.

IDEF1 (IDEF1X) – используется для создания **информационных** моделей, представляющих структуру информации, необходимую для поддержки функций систем. Многие CASE-инструменты, используемые при разработке современных систем управления базами данных (СУБД), позволяют автоматически генерировать программный код из IDEF1X диаграмм.

IDEF2 – используется для построения **динамических** моделей изменения во времени функций, информации и ресурсов систем.

Основу подхода и, соответственно, методологии IDEF0 составляет графический язык. Графический язык – это полное и выразительное средство, способное представить весь спектр процессов предприятия на любом уровне детализации:

- обеспечивает точное и лаконичное описание моделируемых объектов;
- облегчает взаимодействие и взаимопонимание специалистов, занятых анализом и проектированием процессов

К достоинствам графического языка можно отнести следующие факторы:

- язык прошел многолетнюю проверку и подтвердил свою работоспособность;
- легок и прост в изучении;
- поддерживается рядом программных продуктов.

История возникновения стандарта IDEF0. Методологию IDEF0 можно считать следующим этапом развития хорошо известного графического языка описания функциональных систем SADT (Structured Analysis and Design Technique). В процессе практической реализации, участники программы ICAM столкнулись с необходимостью разработки новых методов анализа процессов взаимодействия в промышленных системах. При этом, кроме усовершенствованного набора функций для описания бизнес-процессов, одним из требований к новому стандарту было наличие эффективной методологии взаимодействия в рамках «аналитик-специалист». Другими словами, новый метод должен был обеспечить групповую работу над созданием модели, с непосредственным участием всех аналитиков и специалистов, занятых в рамках проекта. В результате поиска соответствующих решений родилась методология функционального моделирования IDEF0. С 1981 года стандарт IDEF0

претерпел несколько незначительных изменений, в основном ограничивающего характера, и последняя его редакция была выпущена в декабре 1993 года Национальным Институтом По Стандартам и Технологиям США (NIST).

Основные элементы и понятия IDEF0. Графический язык IDEF0 удивительно прост и гармоничен. В основе методологии лежат четыре основных понятия.

Первым из них является понятие *функционального блока «Работа» (Activity Box)*, который графически изображается в виде прямоугольника и олицетворяет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении (например, «производить услуги», а не «производство услуг»).

Каждая из четырех сторон функционального блока имеет своё определенное значение (роль). Стрелка, входящая в соответствующую сторону имеет то же название:

- 1) верхняя сторона имеет значение «Управление» (*Control*);
- 2) левая сторона имеет значение «Вход» (*Input*);
- 3) правая сторона имеет значение «Выход» (*Output*);
- 4) нижняя сторона имеет значение «Механизм» (*Mechanism*).

Каждый функциональный блок в рамках единой рассматриваемой системы должен иметь свой уникальный идентификационный номер.

Вторым «китом» методологии IDEF0 является понятие *интерфейсной дуги (Arrow)*. Также интерфейсные дуги часто называют потоками или стрелками. Интерфейсная дуга отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, отображенную данным функциональным блоком.

Графическим отображением интерфейсной дуги является однонаправленная стрелка. Каждая интерфейсная дуга должна иметь свое уникальное наименование (*Arrow Label*). По требованию стандарта, наименование должно быть оборотом существительного.

С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Такими объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т.д.) или потоки данных и информации (документы, данные, инструкции и т.д.).

В зависимости от того, к какой из сторон подходит данная интерфейсная дуга, она носит название «входящей», «исходящей» или

«управляющей». Кроме того, «источником» (началом) и «приемником» (концом) каждой функциональной дуги могут быть только функциональные блоки. Необходимо отметить, что любой функциональный блок по требованиям стандарта должен иметь по крайней мере одну управляющую интерфейсную дугу и одну исходящую.

Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (*Data Flow Diagram*) и WFD (*Work Flow Diagram*).

Третьим основным понятием стандарта IDEF0 является **декомпозиция** (*Decomposition*). Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели.

Декомпозиция позволяет постепенно и структурированно представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

Последним из понятий IDEF0 является **глоссарий** (*Glossary*). Для каждого из элементов IDEF0, – диаграмм, функциональных блоков, интерфейсных дуг, – существующий стандарт подразумевает создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые характеризуют объект, отображенный данным элементом. Этот набор называется глоссарием и является описанием сущности данного элемента. Например, для управляющей интерфейсной дуги «распоряжение об оплате» глоссарий может содержать перечень полей соответствующего дуге документа, необходимый набор виз и т.д. Глоссарий гармонично дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

Построение модели IDEF0 всегда начинается с представления системы как единого целого – одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области. Такая диаграмма с одним функциональным блоком называется контекстной диаграммой, и обозначается идентификатором «А-0».

В пояснительном тексте к контекстной диаграмме должна быть указана цель (*Purpose*) построения диаграммы в виде краткого описания и зафиксирована точка зрения (*Viewpoint*).

Определение и формализация цели разработки модели IDEF0 – крайне важный момент. Фактически цель определяет соответствующие области в исследуемой системе, на которых необходимо

фокусироваться в первую очередь. Например, если мы моделируем деятельность предприятия с целью построения в дальнейшем на базе этой модели информационной системы, то эта модель будет существенно отличаться от той, которую бы мы разрабатывали для того же самого предприятия, но уже с целью оптимизации логистических цепочек.

Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Четкое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему. Правильный выбор точки зрения существенно сокращает временные затраты на построение конечной модели.

В процессе декомпозиции функциональный блок, который в контекстной диаграмме отображает систему как единое целое, подвергается детализации на другой диаграмме. Получившаяся диаграмма второго уровня содержит функциональные блоки, отображающие главные подфункции функционального блока контекстной диаграммы и называется дочерней (*Child diagram*) по отношению к нему (каждый из функциональных блоков, принадлежащих дочерней диаграмме соответственно называется дочерним блоком – *Child Box*). В свою очередь, функциональный блок-предок называется родительским блоком по отношению к дочерней диаграмме (*Parent Box*), а диаграмма, к которой он принадлежит – родительской диаграммой (*Parent Diagram*). Каждая из подфункций дочерней диаграммы может быть далее детализирована путем аналогичной декомпозиции соответствующего ей функционального блока. Важно отметить, что в каждом случае декомпозиции функционального блока все интерфейсные дуги, входящие в данный блок, или исходящие из него фиксируются на дочерней диаграмме. Этим достигается структурная целостность IDEF0-модели.

Часто бывают случаи, когда отдельные интерфейсные дуги не имеет смысла продолжать рассматривать в дочерних диаграммах ниже какого-то определенного уровня в иерархии, или, наоборот, – отдельные дуги не имеют практического смысла выше какого-то уровня. Например, интерфейсную дугу, изображающую «деталь» на входе в функциональный блок «Обработать на токарном станке» не имеет смысла отражать на диаграммах более высоких уровней – это будет только перегружать диаграммы и делать их сложными для восприятия. С другой стороны, случается необходимость избавиться от отдельных «концептуальных» интерфейсных дуг и не детализировать их глубже некоторого уровня. Для решения подобных задач в стандарте

IDEF0 предусмотрено понятие туннелирования. Обозначение «туннеля» (*Arrow Tunnel*) в виде двух круглых скобок вокруг начала интерфейсной дуги обозначает, что эта дуга не была унаследована от функционального родительского блока и появилась (из «туннеля») только на этой диаграмме. В свою очередь, такое же обозначение вокруг конца (стрелки) интерфейсной дуги в непосредственной близости от блока-приёмника означает тот факт, что в дочерней по отношению к этому блоку диаграмме эта дуга отображаться и рассматриваться не будет.

Принципы ограничения сложности IDEF0-диаграмм. Обычно IDEF0-модели содержат сложную и концентрированную информацию, и для того, чтобы избежать их перегруженности и сделать удобочитаемыми, в соответствующем стандарте приняты соответствующие ограничения сложности.

Ограничение количества функциональных блоков на диаграмме (три-шесть). Верхний предел (шесть) заставляет разработчика использовать иерархии при описании сложных предметов, а нижний предел (три) гарантирует, что на соответствующей диаграмме достаточно деталей, чтобы оправдать ее создание.

Ограничение количества интерфейсных дуг (четыре), подходящих к одному функциональному блоку или выходящих из одного функционального блока.

Разумеется, строго следовать этим ограничениям вовсе необязательно, однако, как показывает опыт, они являются весьма практичными в реальной работе.

Дисциплина групповой работы над разработкой IDEF0-модели. Стандарт IDEF0 содержит набор процедур, позволяющих разрабатывать и согласовывать модель большим коллективом, включающим специалистов в разных областях деятельности моделируемой системы. Обычно процесс разработки является итеративным и состоит из следующих условных этапов.

Создание модели группой специалистов, относящихся к различным сферам деятельности предприятия. Эта группа в терминах IDEF0 называется авторами (*Authors*). Построение первоначальной модели – динамический процесс: авторы опрашивают компетентных лиц о структуре различных процессов. На основе имеющихся положений, документов и результатов опросов создается черновик (*Model Draft*) модели.

Распространение черновика для рассмотрения, согласований и комментариев. На этой стадии происходит обсуждение черновика

модели с широким спектром компетентных лиц (в терминах IDEF0 – читателей) на предприятии. При этом каждая из диаграмм черновой модели письменно критикуется и комментируется, а затем передается автору. Автор, в свою очередь, также письменно соглашается с критикой или отвергает её с изложением логики принятия решения и вновь возвращает откорректированный черновик для дальнейшего рассмотрения. Этот цикл продолжается до тех пор, пока авторы и читатели не придут к единому мнению.

Официальное утверждение модели. Утверждает согласованную модель руководитель рабочей группы в том случае, если у авторов модели и читателей отсутствуют разногласия по поводу ее адекватности. Окончательная модель представляет собой согласованное представление о предприятии (системе) с заданной точки зрения и для заданной цели.

Наглядность графического языка IDEF0 делает модель вполне читаемой, в том числе и для лиц, которые не принимали участия в процессе ее создания, а также эффективной для проведения показов и презентаций. В дальнейшем на базе построенной модели могут быть организованы новые проекты, нацеленные на производство изменений на предприятии (в системе).

Особенности национальной практики применения функционального моделирования средствами IDEF0. В последние годы внимание в России к методологиям семейства IDEF неуклонно растет. При этом интерес к таким стандартам, как IDEF3-5, Геннадий Верников называет теоретическим, а к IDEF0 – вполне практически обоснованным. Собственно говоря, первые Case-средства, позволяющие строить DFD и IDEF0 диаграммы, появились на российском рынке еще в 1996 году, одновременно с выходом популярной книги по принципам моделирования в стандартах SADT. Тем не менее, большинство руководителей до сих пор расценивают практическое применение моделирования в стандартах IDEF скорее как дань моде, нежели чем эффективный путь оптимизации существующей системы управления бизнесом.

Стандарты IDEF в понимании большинства стали условно неотделимы от внедрения информационных технологий, хотя с их помощью порой можно эффективно решать даже небольшие локальные задачи, буквально при помощи карандаша и бумаги.

Одним из программных продуктов, реализующих эту методологию является *CA AllFusion Process Modeler (BPwin)*.

Функциональное моделирование на базе стандарта IDEF0

Цель работы – ознакомление пользователя с методологией функционального моделирования **IDEF0**, получения практических навыков в построении функциональной модели и анализе результатов моделирования.

Теоретические сведения

Методология IDEF0 основывается на следующих концептуальных положениях.

1.1. Модель

Модель дает полное, точное и адекватное описание системы и имеет конкретное назначение. Это назначение называется **целью** модели:

М моделирует систему С, если М отвечает на вопросы относительно С с точностью Т.

Целью создания модели является получение ответов на некоторую совокупность вопросов. Именно эти вопросы руководят процессом созданием модели и направляют его.

Если модель отвечает не на все вопросы, или ее ответы не точны, считается, что модель не достигла поставленной цели.

Четкая постановка цели, когда ряд вопросов сводится к одному предложению, позволяет определить направления и требуемую степень точности модели. Только осознав, насколько хорошо необходимо ответить на поставленные вопросы, можно определить, когда процесс моделирования будет считаться завершенным. Качество модели оценивается степенью полноты ответов на поставленные вопросы.

1.2 Блочное моделирование и его графическое представление

Основной концептуальный принцип IDEF0 состоит в представлении любой анализируемой системы в виде набора взаимосвязанных и взаимодействующих блоков. Этот набор отображает процессы, происходящие в анализируемой системе.

В методологии IDEF0 все, что происходит в системе, принято называть **функциями**. Каждой функции ставится в соответствии **блок**. Связи и взаимодействия между блоками на диаграмме IDEF0 описываются входящими в блок и выходящими из него стрелками. Входящие стрелки показывают, какие условия должны быть выполнены для осуществления функции.

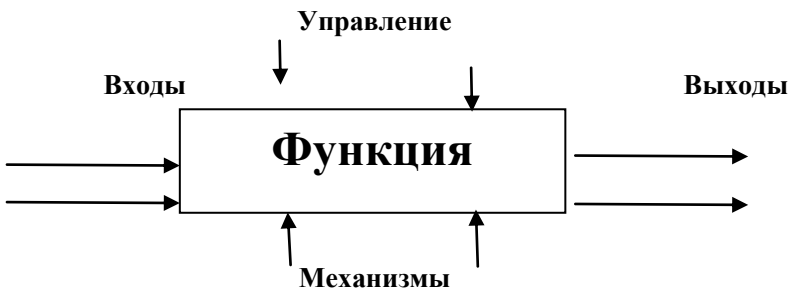


Рис. 1. Структура блока IDEF0

1.3. Лаконичность и точность

Описание модели системы должно быть лаконичным и точным. С этой точки зрения, многословное описание на естественном языке не подходит. Графический язык позволяет максимально точно и лаконично показать все элементы системы.

1.4. Передача информации

Средства IDEF0 позволяют легко передавать информацию от одного участника построения модели к другому. Это обеспечивается следующими факторами:

- простотой графического языка, позволяющего легко читать и понимать схему модели;
- для уточнения смысла диаграммы используются метки и пояснения;
- иерархической структурой диаграмм, что приводит к детализации и уточнению каждой функции;
- древовидной структурой диаграмм модели, что обеспечивает обзорность, как модели, так и входящих в нее элементов.

1.5. Строгость и формализм

Разработка модели IDEF0 ведется с использованием строгих формальных правил, определяемых как самим стандартом, так и синтаксисом графического языка.

1.6. Итеративное моделирование

Разработка модели IDEF0 ведется пошагово, с обсуждением каждой части модели и ее утверждением.

1.7. Отделение организации от функций

Разработка модели ведется без привязки к существующей организационной структуре, что помогает избежать внесения в модель субъективного мнения руководителей организации. Организационная структура появляется как результат использования модели. Несоответствия между организационной структурой, полученной в результате моделирования, и уже имеющейся организационной структурой позволяют выработать решения по совершенствованию организационной структуры.

Методология функционального моделирования IDEF0 – это технология описания системы в целом как множества взаимозависимых действий или функций. Важно отметить функциональную направленность: IDEF0-функции системы исследуются независимо от объектов, которые обеспечивают их выполнение. «Функциональная» точка зрения позволяет четко отделить аспекты назначения системы от аспектов ее физической реализации. На рис. 2 приведен пример типовой диаграммы IDEF0.

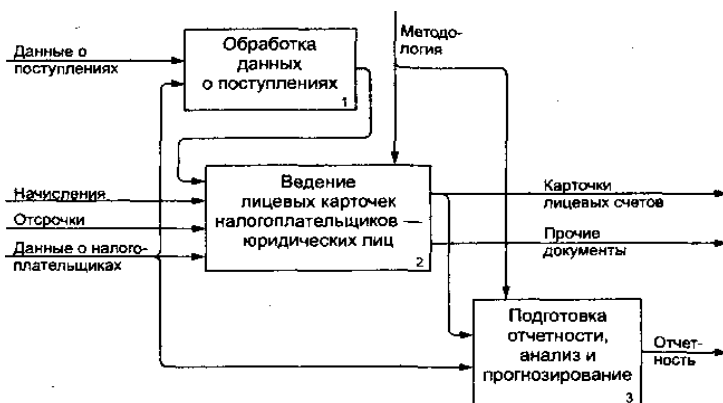


Рис. 2. Пример диаграммы IDEF0

Обслуживание клиента системы. Построение модели системы должно начинаться с изучения всех документов, описывающих ее функциональные возможности. Одним из таких документов является техническое задание, а именно его разделы «Назначение разработки», «Цели и задачи системы» и «Функциональные характеристики системы».

После изучения исходных документов и опроса заказчиков и пользователей системы необходимо сформулировать цель моделирования и определить точку зрения на модель.

Сформулируем цель моделирования: описать функционирование системы, которое было бы понятно ее пользователю, не вдаваясь в подробности, связанные с реализацией. Модель будем строить с точки зрения пользователей (студент, преподаватель, администратор, деканат, фирма).

Начнем с построения контекстной IDEF0-диаграммы. Согласно описанию системы, ее основной функцией является обслуживание клиентов посредством обработки запросов, от них поступающих. Таким образом, определим единственную работу контекстной диаграммы как «Обслужить клиента системы». Далее определим входные и выходные данные, а также механизмы и управление.

Для того чтобы обслужить клиента, необходимо зарегистрировать его в системе, открыть доступ к базам данных (БД) и обработать его запрос. В качестве входных данных будут использоваться понятия «имя клиента», «пароль клиента», «исходная БД», «запрос клиента». Выполнение запроса ведет либо к получению информации от системы, либо к изменению содержимого БД (например, при составлении экспертных оценок), поэтому выходными данными будут являться «отчеты» и «измененная БД». Процесс обработки запросов будет выполняться монитором системы под контролем администратора.

Таким образом, определим контекстную диаграмму системы (рис. 3). Проведем декомпозицию контекстной диаграммы, описав последовательность обслуживания клиента:

- Определение уровня доступа в систему.
- Выбор подсистемы.
- Обращение к подсистеме.
- Изменение БД (при необходимости).



Рис. 3. Контекстная диаграмма системы

Получим диаграмму, изображенную на рис. 4.

Закончив декомпозицию контекстной диаграммы, переходят к декомпозиции диаграммы следующего уровня. Обычно при рассмотрении третьего и более нижних уровней модели возвращаются к родительским диаграммам и корректируют их.

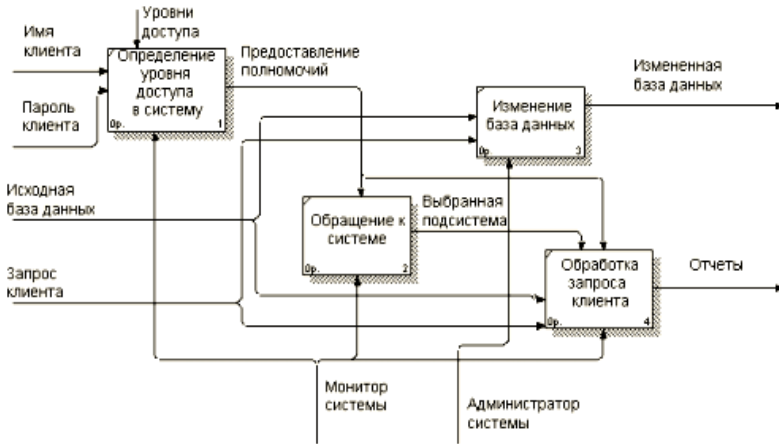


Рис. 4. Декомпозиция работы «Обслуживание, клиента системы»

Декомпозируем последовательно все блоки полученной диаграммы. Первым этапом при определении уровня доступа в систему является определение категории пользователя. По имени клиента осуществляется поиск в базе пользователей, определяя его категорию. Согласно определенной категории выясняются полномочия, предоставляемые пользователю системы. Далее проводится процедура доступа в систему, проверяя имя и пароль доступа. Объединяя информацию о полномочиях и уровне доступа в систему, для пользователя формируется набор разрешенных действий. Таким образом, определение уровня доступа в систему будет выглядеть как показано на рис. 5.

После прохождения процедуры доступа в систему монитор анализирует запрос клиента, выбирая подсистему, которая будет обрабатывать запрос. Декомпозиция работы «Обращение к подсистеме» не отвечает цели и точке зрения модели. Пользователя системы не интересуют внутренние алгоритмы ее работы. В данном случае ему важно, что выбор подсистемы будет произведен автоматически, без его вмешательства, поэтому декомпозиция обращения к подсистеме только усложнит модель.

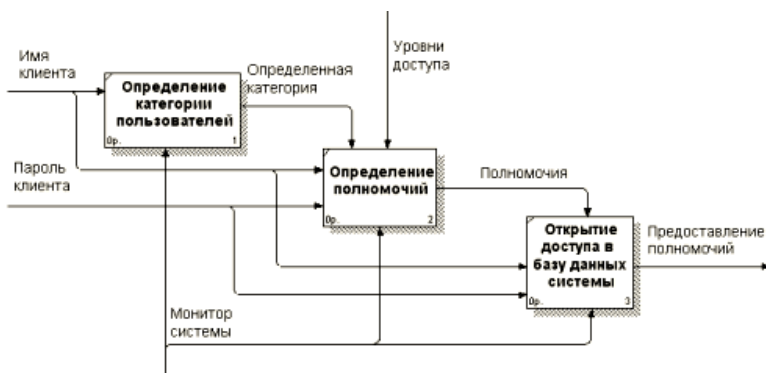


Рис. 5. Декомпозиция работы «Определение уровня доступа в систему»

Декомпозируем работу «Обработка запроса клиента», выполняемую подсистемой обработки запросов, определения категорий и полномочий пользователей. Перед осуществлением поиска ответа на запрос необходимо открыть БД (подключиться к ней). В общем случае БД может находиться на удаленном сервере, поэтому может потребоваться установление соединения с ней. Определим последовательность работ.

- Открытие БД.
- Выполнение запроса.
- Генерация отчетов.

После открытия БД необходимо сообщить системе об установлении соединения с БД, после чего следует выполнить запрос и сгенерировать отчеты для пользователя (рис. 6).

Необходимо отметить, что в «Выполнение запроса» включается работа различных подсистем. Например, если запрос включает в себя тестирование, то его будет исполнять подсистема профессиональных и психологических тестов. На этапе выполнения запроса может потребоваться изменение содержимого БД, например при составлении экспертных оценок. Поэтому, на диаграмме необходимо предусмотреть такую возможность.

При анализе полученной диаграммы возникает вопрос: по каким правилам происходит генерация отчетов? Необходимо наличие заранее сформированных шаблонов, по которым будет производиться выборка из БД, причем эти шаблоны должны соответствовать запросам и должны быть заранее определены. Кроме того, клиенту должна быть предоставлена возможность выбора формы отчета.

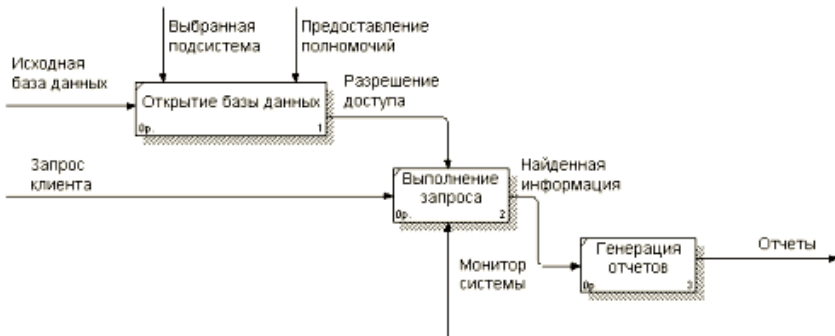


Рис. 6. Декомпозиция работы «Обработка запроса клиента»

Скорректируем диаграмму, добавив в нее стрелки «Шаблоны отчетов» и «Запросы на изменение БД» и туннельную стрелку «Клиент системы». Туннелирование «Клиента системы» применено для того, чтобы не выносить стрелку на диаграмму верхнего, так как функция выбора формы отчета не является достаточно важной для отображения ее на родительской диаграмме.

Техническая эксплуатация ВС.

Для выполнения работ необходимы средства и персонал, а также управляющие воздействия, определяющие порядок работ, которые также отражены на модели IDEF0.

Функциональная модель системы технической эксплуатации и входящей в нее системы ТО представлена на рис. 7 – 10.



Рис. 7. Верхний уровень функциональной модели технической эксплуатации воздушных судов

Последним элементом, до которого производится детализация функциональной системы, является простейший элемент, разложение которого на составные части не представляется возможным.

Процесс поддержания ВС в исправном состоянии осуществляется в рамках системы ТО. В качестве исходной информации для осуществления данного процесса выступают заявки (содержат перечень работ и могут содержать описание отказа или дефекта), требования (на оказания услуг), запросы (на доработку технической документации или для продления ресурса ВС), документы об условиях эксплуатации. В результате обработки заявок и требований решается, какие виды работ должны быть проведены на ВС. При выполнении заявок и требований определяются ответственные за их исполнение, а также документация, используемая при выполнении задач.

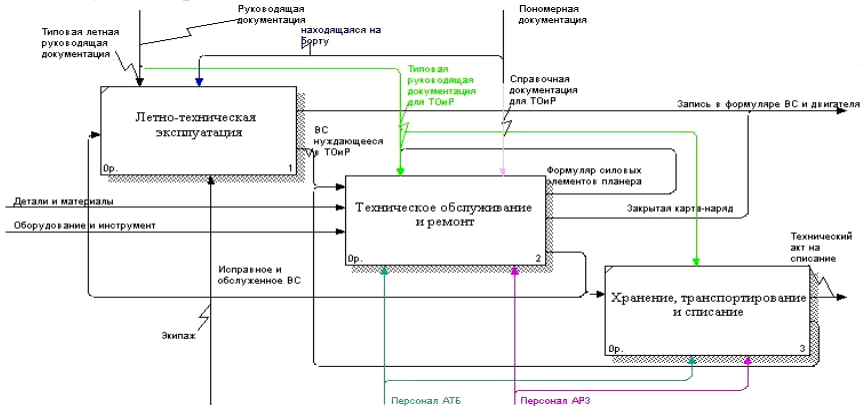


Рис. 8. Составляющие компоненты системы технической эксплуатации

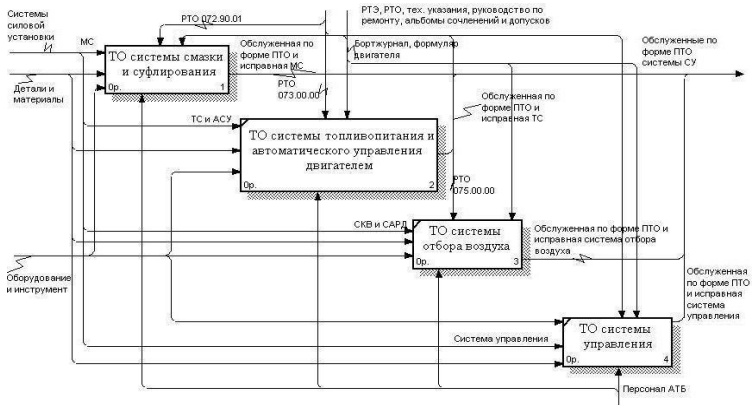


Рис. 9. Представление процесса технического обслуживания систем силовой установки

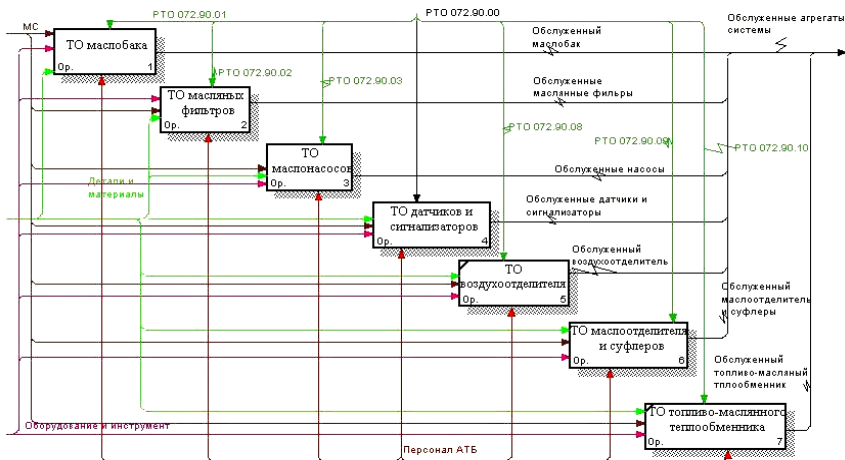


Рис. 10. Пример технического обслуживания агрегатов системы смазки двигателя

В случае определения необходимости выполнения одного из видов работ, они так же могут быть представлены в виде функциональной модели на низшем уровне иерархической структуры.

Задание. Смоделировать работу системы технического обслуживания воздушного судна по указанию преподавателя.

Контрольные вопросы

1. Что представляет собой модель в нотации IDEF0?
2. Что обозначают работы в IDEF0?
3. Назовите порядок наименования работ?
4. Какое количество работ должно присутствовать на одной диаграмме?
5. Что называется порядком доминирования?
6. Как располагаются работы по принципу доминирования?
7. Каково назначение сторон прямоугольников работ на диаграммах?
8. Перечислите типы стрелок.
9. Объясните принцип именования разветвляющихся и сливающихся стрелок.

Список используемых источников

1. Черемных, С.В. Структурный анализ систем: IDEF-технологии / С.В. Черемных, И.О. Семенов, В.С. Ручкин. – М.: Финансы и статистика, 2003. – 208 с.
2. Ретин, В.В. Процессный подход к управлению. Моделирование бизнес-процессов / В.В. Ретин, В.Г. Елиферов – М.: РИА «Стандарты и качество», 2004. – 398 с.
3. Верников Г. Описание стандарта IDEF0 [электронный ресурс] / Геннадий Верников. – Режим доступа www.edefinfo.ru.
4. Дубейковский, В.И. Эффективное моделирование AllFusion Process Mockler (Врwin) / В.И. Дубейковский. – М.: Диалог-МиФи, 2009. – 384 с.
5. Дубейковский, В.И. Практика функционального моделирования / В. И. Дубейковский– М.: Диалог-МиФи, 2004. – 464 с.

Учебное издание

**ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ
НА БАЗЕ СТАНДАРТА IDEFO**

Методические указания

Составители:

***Киселев Денис Юрьевич,
Киселев Юрий Витальевич,
Вавилин Артем Валерьевич***

Редактор И.И. Спиридонова
Доверстка И.И. Спиридонова

Подписано в печать 8.12.2014. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная.
Печ. л. 1,25. Тираж 50 экз. Заказ . Арт. 40 /2014.

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

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