Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

ОТЧЕТ

по лабораторной работе

на тему:

Функциональное моделирование. Разработка функциональной модели предметной области в среде CASE-средства BPwin с использованием методологий IDEF0 и IDEF3

Выполнил:

Студент гр. 851005

Ермолович И.В.

Проверил:

Асс. Фадеева Е.Е.

Минск, 2020

**Этапы ЖЦ ПС**

* В соответствии со стандартом *СТБ ИСО/МЭК* *12207–2003* под ***жизненным циклом*** программного средства или системы подразумевается совокупность процессов, работ и задач, включающая в себя разработку, эксплуатацию и сопровождение программного средства или системы, и охватывающая их жизнь от формулирования концепции до прекращения использования. ЖЦ ПС состоит из *процессов*. Каждый процесс ЖЦ разделен на набор *работ*. Каждая работа разделена на набор *задач*.

***Процессы ЖЦ ПС*** делятся на три*группы*:

* основные;
* вспомогательные;
* организационные.

**Основные процессы жизненного цикла** –это процессы,которые реализуются под управлением основных сторон, участвующих в жизненном цикле программных средств. Основными сторон ми являются заказчик, поставщик, разработчик, оператор и персонал сопровождения программных продуктов. К *основным процессам* относится пять процессов:

* Заказ
* Поставка
* Разработка
* Эксплуатация
* Сопровождение

Процесс **заказа** определяет работы и задачи заказчика и состоит из определения потребностей заказчика в системе или программном продукте, подготовки и выпуска заявки на подряд, выбора поставщика и управления процессом заказа до завершения приемки системы или программного продукта.

***Процесс поставки*** определяет работы и задачи поставщика.Данныйпроцесс начинается с решения о подготовке предложения в ответ на заявку на подряд, присланную заказчиком, или с подписания договора с заказчиком на поставку системы или программного продукта. Затем определяются процедуры и ресурсы, необходимые для управления и обеспечения проекта, включая разработку проектных планов и их выполнение.

***Процесс разработки*** состоит из работ и задач,выполняемых разработчиком. Данный процесс содержит *тринадцать работ*:

1. подготовка процесса разработки;
2. анализ требований к системе;
3. проектирование системной архитектуры;
4. анализ требований к программным средствам;
5. проектирование программной архитектуры;
6. техническое проектирование программных средств;
7. программирование и тестирование программных средств;
8. сборка программных средств;
9. квалификационные испытания программных средств;
10. сборка системы;
11. квалификационные испытания системы;
12. ввод в действие программных средств;
13. обеспечение приемки программных средств.

***Процесс эксплуатации*** определяет работы и задачи оператора.Данныйпроцесс включает эксплуатацию программного продукта и поддержку пользователей в процессе эксплуатации.

***Процесс сопровождения*** определяет работы и задачи персонала сопровождения и реализуется при модификациях программного продукта. Назначением процесса является изменение существующего программного продукта при сохранении его целостности. Процесс охватывает вопросы переносимости и снятия программного продукта с эксплуатации.

**CASE -средства**

***CASE-технология*** –это совокупность методологий разработки и сопровождения сложных систем (в том числе ПС ), поддерживаемая комплексом взаимосвязанных средств автоматизации.

*Основные цели использования CASE -технологий* при разработке ПС–отделить анализ и проектирование от программирования и последующих работ процесса разработки, предоставив разработчику соответствующие методологии визуального анализа и проектирования.

В настоящее время при выполнении ранних работ процесса разработки (работ, связанных с анализом предметной области и проектированием систем и ПС) широко используются CASE-технологии. Термин CASE расшифровывается как Computer Aided System Engineering (компьютерная поддержка проектирования систем). При структурном анализе и проектировании широко применяется семейство методологий IDEF.

***IDEF0 –*** методология функционального моделирования производственной среды или системы; отображает структуру и функции системы, а также потоки информации и материальных объектов, связывающие эти функции; основана на методе SADT Росса;

***IDEF3*** –методология моделирования сценариев процессов,происходящих в производственной среде или системе; отображает состояния объектов и потоков данных, связи между ситуациями и событиями; используется для документирования процессов, происходящих в производственной среде или системе;

Под Системой подразумевается совокупность взаимодействующих компонентов и взаимосвязей̆ между ними.

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

Под ***моделью*** ***IDEF0*** подразумевается графическое и текстовое представ-ление результатов анализа предметной области, разработанное с определенной целью и с выбранной точки зрения и идентифицирующее функции системы [1]. 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:

M 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. 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 точки зрения, с которой наблюдается система и создается модель. Точка зрения - это позиция человека или объекта, в которую надо встать, чтобы увидеть систему в действии. Методология IDEF0 требует, чтобы модель рассматривалась все время с одной и той же позиции.  
Например, при разработке автоматизированной обучающей системы (АОС) точкой зрения может быть позиция неквалифицированного пользователя, квалифицированного пользователя, программиста и т.п.

***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***.

Методология IDEF0 создана специально для представления сложных систем путем построения моделей. IDEF0-модель – это описание системы, разработанное для единственного субъекта с определенной целью и с выбранной точки 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 со-стоит из наиболее детализированных описаний.

**СИНТАКСИС IDEF0-ДИАГРАММ**

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

Функциональные блоки на диаграмме изображаются прямоугольниками. 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).

**Назначение дуг**

* IDEF0 различают ***пять типов дуг****:* вход (input), управление (control), выход (output), механизм (mechanism), вызов (call)
* основе методологии IDEF0 лежат следующие *правила*:

1. *вход* представляет собой входные данные,используемые или преобразуемые функциональным блоком для получения результата (выхода); блок может не иметь ни одной входной дуги (например блок, выполняющий генерацию случайных чисел);
2. *выход* представляет собой результат работы блока;наличие выходнойдуги для каждого блока является обязательным;
3. *управление* ограничивает или определяет условия выполнения преобразований в блоке; в качестве дуг управления могут использоваться некоторые условия, правила, стратегии, стандарты, которые влияют на выполнение функционального блока; наличие управляющей дуги для каждого блока является обязательным;
4. *Механизмы* показывают,то,что и как выполняет преобразования вблоке; механизмы определяют иные ресурсы, непосредственно осуществляющие эти преобразования (например, денежные средства, персонал, оборудование и т.п.); механизмы представляются стрелками, подключенными к нижней стороне блока и направленными вверх к блоку; наличие дуг механизмов для блока не является обязательным;
5. *Вызовы* представляют собой специальный вид дуги и обозначают обращение из данной модели или из данной части модели к блоку, входящему в состав другой или другой части модели, обеспечивая их связь; с помощью дуги вызова разные модели или разные части одной модели могут совместно использовать один и тот же блок; вызовы не являются компонентом собственно методологии SADT, они является расширением IDEF0-методологии и предназначены для организации коллективной работы над моделью, разделения модели на независимые модели и объединения различных моделей предметной области в одну модель; вызовы представляются стрелка-ми, подключенными к нижней стороне блока и направленными вниз от блока; наличие дуги вызова для блока не является обязательным.

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.  
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.

**Типы взаимосвязей между блоками**

Дуги определяют, как блоки влияют друг на друга. Это влияние может выражаться:

* в передаче выходной информации к другому блоку для дальнейшего преобразования;
* в выработке управляющей информации, предписывающей, что именно должен выполнить другой блок;
* в передаче информации, определяющей средство достижения цели для другого блока.
* учетом этого в методологии IDEF0 используется ***пять типов взаимо-связей между блоками*** для описания их отношений:управление,вход,обрат-ная связь по управлению, обратная связь по входу, выход-механизм

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

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

Обратные связи по управлению и по входу предназначены для преставления итерации или рекурсии.

*Обратная связь по управлению* возникает тогда,когда выход некоторогоблока влияет на работу блока с большим доминированием (рис. 5.5).

*Обратная связь по входу* имеет место тогда,когда выход одного блокастановится входом другого блока с большим доминированием (рис. 5.6).

*Связь «выход-механизм»* встречается нечасто и отражает ситуацию,прикоторой выход одного блока становится средством достижения цели для другого блока (рис. 5.7). Данная связь характерна при распределении источников ресурсов (например, физическое пространство, оборудование, финансирование, материалы, инструменты, обученный персонал и т. п.).

* 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)  
  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.

**Тоннельные дуги**

Особые ситуации возникают, когда дуги «входят в тоннель» между диаграммами. Дуга «входит в тоннель» в следующих случаях:

1. Она является внешней дугой, которая отсутствует на родительской диаграмме.(дуга имеет скрытый источник)
2. Она касается родительского блока, но не появляется на диаграмме, которая его декомпозирует(дуга имеет скрытый приемник)

Тоннельные дуги от скрытого источника начинаются круглыми 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.

Тоннельные дуги со скрытым приемником заканчиваются круглыми скобками, чтобы отразить тот факт, что такая дуга идет к какой-то другой части модели, выходит из нее или не будет более в этом модели рассматриваться.

Таким образом, «вхождение в тоннель» для дуг используется чаще всего для упрощения описания системы – тогда, когда диаграммы в модели становятся слишком сложными для чтения и понимания.

**Вывод**  
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 C-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.

**IDEF3**

Для описания логики взаимодействия информационных потоков модель  
системы дополняют диаграммами методологии IDEF3. Диаграммы данного  
вида называются диаграммами потоков работ (WorkFlow Diagram).  
Методология моделирования IDEF3 позволяет графически описать течение  
процессов во времени и отношения процессов и объектов, являющихся частями  
этих процессов.

В методологии IDEF3 существует два типа моделей:  
· модель, отражающая процессы в их логической последовательности и  
позволяющая увидеть функционирование системы;  
· модель, отражающая “сеть переходных состояний объекта” и  
позволяющая увидеть последовательность состояний, в которых может  
оказаться объект при прохождении через определенный процесс.  
С помощью диаграмм IDEF3 можно анализировать динамику событий из  
реальной жизни, например, какие действия должны выполнять различные  
сотрудники университета во время вступительных экзаменов или во время  
учебной воздушной тревоги и т.п.

**Единицы работы (Unit of Work - UOW)** - основной компонент  
диаграммы IDEF3, близкий по смыслу к функциональному блоку IDEF0. Единицы работы называются также работами (Activity). Имя работы  
записывается в центральном поле прямоугольника.  
В левом нижнем углу прямоугольника записывается идентификатор  
работы. Он присваивается при создании модели и не изменяется в процессе  
моделирования. Идентификатор работы обозначается иерархическим номером.  
В его состав в наиболее полном варианте входит символ А (Activity), номер  
родительской работы, номер декомпозиции, номер работы в пределах данной  
декомпозиции.

**Связи (Arrow Links) –** изображаются линиями со стрелками и  
показывают взаимоотношения работ. Все связи являются однонаправленными и могут быть направлены в любую сторону. Предпочтительнее направление  
связей слева направо или сверху вниз.  
В IDEF3 различают три типа связей:  
· *связь предшествования (Precedence)* – связывает единицы работ и  
обозначает, что, прежде чем начнется работа-приемник, должна  
завершиться работа-источник; изображается сплошной линией со  
стрелкой;  
· *связь отношения (Relational Link)* - показывает связь между двумя  
единицами работ (UOW) или между единицей работы и объектом  
ссылки; обозначается пунктирной линией;  
· *поток объектов (Object Flow)* – показывает участие некоторого  
объекта в двух или более единицах работ (например, объект  
производится в ходе выполнения одной работы и используется в  
другой работе); обозначается сплошной линией с двумя стрелками.

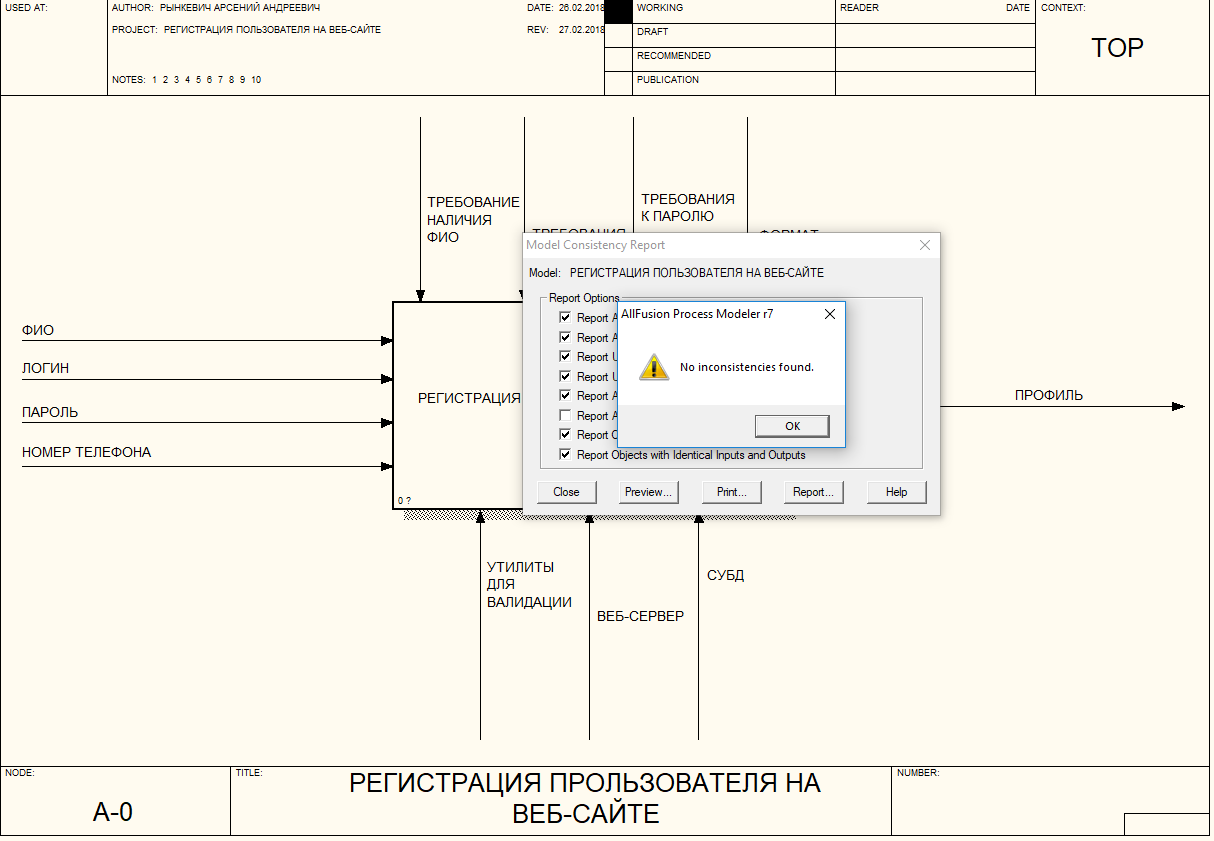
**Перекрестки (Junctions)** – используются, чтобы показать разветвления и  
альтернативные пути развития процесса, которые могут возникнуть во время  
его выполнения. На диаграммах IDEF3 связи могут сливаться и разветвляться  
только через перекрёстки.  
Различают два типа перекрестков:  
· *перекресток слияния (Fan-in Junction)* – узел, собирающий несколько  
связей в одну; указывает на необходимость условия завершенности  
работ (источников связей) для продолжения процесса.  
· *перекресток разветвления (Fan-out Junction)* – узел, в котором  
единственная входящая в него связь разветвляется; показывает, что  
работы, следующие за перекрестком, выполняются параллельно или  
альтернативно.

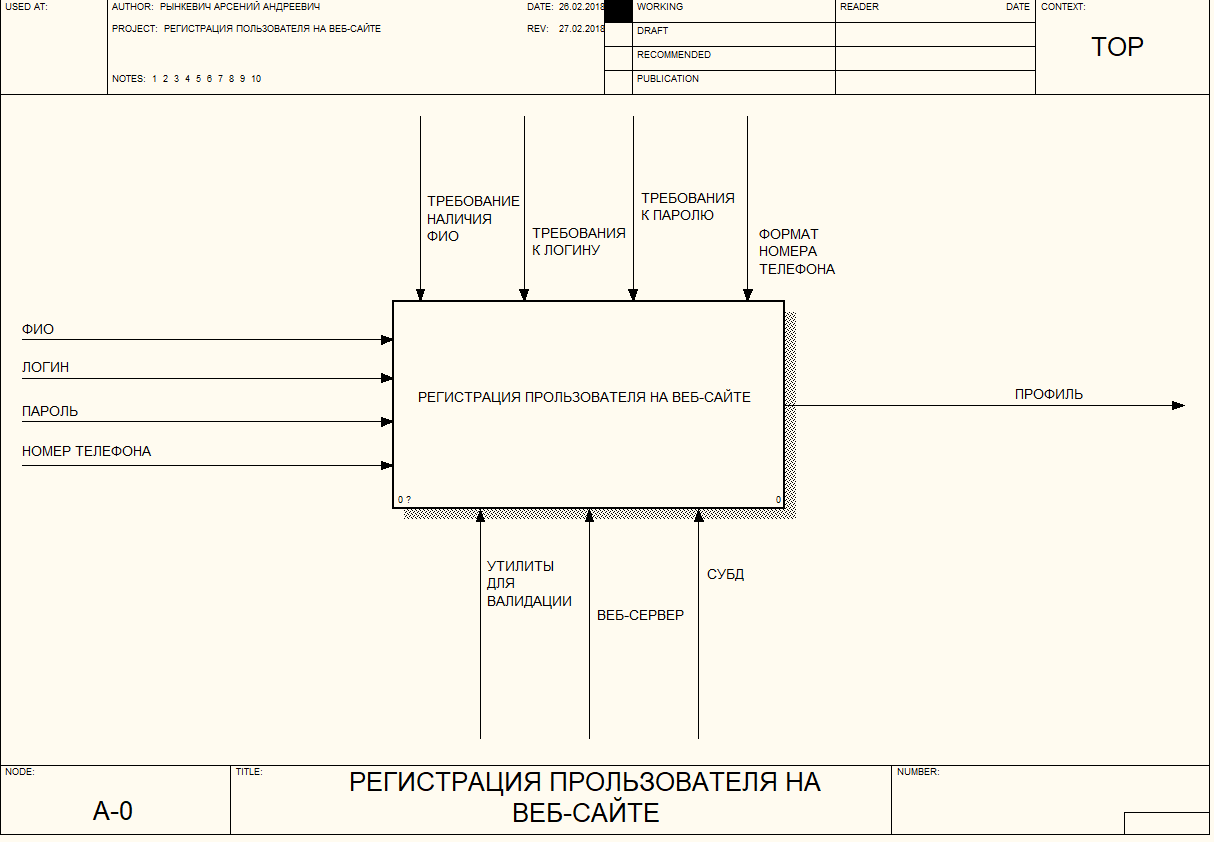
**Объекты ссылок (Referents) -** служат для отображения некоторых идей  
или концепций без использования специальных графических элементов, таких как стрелки, перекрестки или работы.

**Вывод**

Для описания логики взаимодействия информационных потоков модель  
системы дополняют диаграммами методологии IDEF3. Диаграммы данного  
вида называются диаграммами потоков работ (WorkFlow Diagram).  
Методология моделирования IDEF3 позволяет графически описать течение  
процессов во времени и отношения процессов и объектов, являющихся частями  
этих процессов. Таким образом, данная методология подходит для работ, в которых важна синхронизация данных.

**Тестирование**

В процессе проверки ошибок найдено не было.

**Скриншоты**

