

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

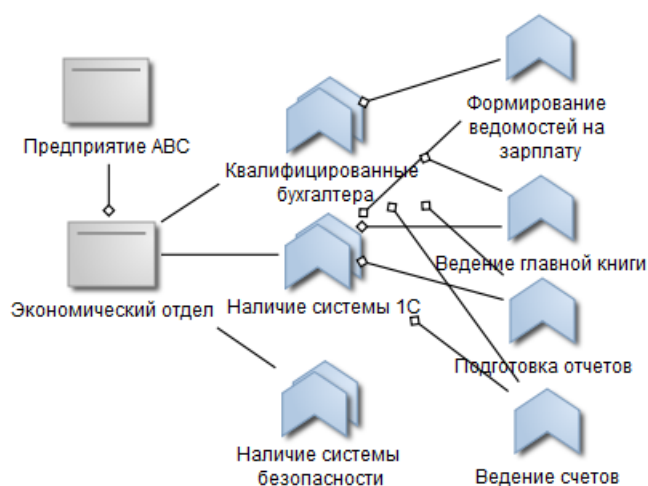
Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт финансов, экономики и управления

А.В. Цуканов

**МОДЕЛИРОВАНИЕ
АРХИТЕКТУРЫ ПРЕДПРИЯТИЙ:
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

Учебное пособие



Севастополь
СевГУ
2019

УДК 681.51(076)

ББК 65.23я73

Ц85

Р е ц е н з е н т ы:

В.Я. Копп – д-р техн. наук, профессор
кафедры приборных систем и автоматизации технологических процессов
Севастопольского государственного университета

Ю.Е. Обжерин – д-р техн. наук, профессор кафедры высшей математики
Севастопольского государственного университета

Цуканов А.В.

Ц85 Моделирование архитектуры предприятий: лабораторный практикум: учеб. пособие / А.В. Цуканов. – Севастополь: СевГУ, 2019. – 178 с.: ил.

Содержание лабораторного практикума соответствует утвержденной в СевГУ рабочей программе дисциплины «Моделирование архитектуры предприятий» по направлению 38.04.02 «Менеджмент».

Предназначен для студентов всех форм обучения и слушателей курсов повышения квалификации в рамках дополнительного профессионального образования.

УДК 681.51(076)

ББК 65.23я73

Рекомендовано Ученым советом института финансов, экономики и управления Севастопольского государственного университета, протокол № 1 от 26 сентября 2018 г.

© Цуканов А.В., 2019

© ФГАОУ ВО «Севастопольский
государственный университет», 2019

Оглавление

Предисловие	5
Введение	6
Глава 1. Начало работы с системой ADOit	8
1.1. Главное меню системы.....	8
1.2. Создание, переименование и удаление папки.....	12
Глава 2. Лабораторный практикум «Архитектурные принципы»...	21
2.1. Принципы в архитектуре предприятия.....	21
2.2. Создание папок с принципами бизнеса.....	24
2.3. Описание принципа.....	29
2.4. Задание по выполнению практикума.....	36
2.5. Содержание отчета.....	36
2.6. Варианты заданий.....	37
2.7. Контрольные вопросы.....	37
2.8. Примеры принципов бизнеса.....	38
Глава 3. Лабораторный практикум «Моделирование организацион- ной структуры предприятия»	39
3.1. Виды организационной структуры предприятия.....	39
3.2. Модель организационной структуры в системе ADOit.....	41
3.3. Пример модели организационной структуры.....	64
3.4. Задание по выполнению практикума	65
3.5. Содержание отчета.....	65
3.6. Варианты заданий	66
3.7. Контрольные вопросы	68
3.8. Примеры принципов создания и управления организацион- ной структурой предприятия.....	69
Глава 4. Лабораторный практикум «Бизнес-архитектура создания ценности»	70
4.1. Сегменты бизнес-архитектуры в системе ADOit.....	70
4.2. Модель клиентов.....	77
4.3. Модель продукта.....	78
4.4. Система взаимодействия с клиентом.....	79
4.5. Система дистрибуции.....	80
4.6. Возможности (“Capability”) бизнеса.....	81
4.7. Бизнес функции.....	88
4.8. Задание по выполнению практикума	96
4.9. Содержание отчета.....	97
4.10. Варианты заданий.....	97
4.11. Контрольные вопросы	99
Глава 5. Лабораторный практикум «Моделирование архитектуры бизнес-процессов»	100

5.1. Архитектура бизнес-процессов.....	100
5.2. Связь процессов с функциями.....	107
5.3. Модель данных.....	109
5.4. Связь процессов и данных.....	112
5.5. Матричное представление связей между объектами.....	113
5.6. Заинтересованные лица (Stakeholders).....	119
5.7. Ресурсы бизнес-процессов.....	120
5.8. Задание по выполнению практикума.....	120
5.9. Содержание отчета.....	120
5.10. Варианты заданий.....	121
5.11. Контрольные вопросы.....	121
5.12. Фрагмент процессной модели деятельности банка.....	122
Глава 6. Лабораторный практикум «Моделирование архитектуры информационной системы»	124
6.1. Состав архитектуры информационной системы.....	124
6.2. Создание модели архитектуры информационной системы	125
6.3. Объект «Приложение» (Application Component)	126
6.4. Построение пузырьковых диаграмм для приложений	135
6.5. Связи между приложениями и данными.....	137
6.6. Матричные взгляды на приложение.....	140
6.7. Объект «Интерфейс» (Interface).....	152
6.8. Модель технологической архитектуры.....	157
6.9. Построение графика Гантта.....	163
6.10. Два упражнения для закрепления материала.....	165
6.11. Задание по выполнению практикума.....	172
6.12. Содержание отчета.....	172
6.13. Варианты заданий.....	173
6.14. Контрольные вопросы.....	176
Заключение	177
Литература.....	178

Предисловие

Целью настоящего практикума является получение компетенций студентами, обучающимся по направлению «Менеджмент» в области моделирования архитектуры предприятий. Практикум содержит необходимые сведения из теории моделирования архитектуры предприятия, инструкции по работе с системой моделирования архитектуры предприятия ADOit [10], примеры моделирования, индивидуальные задания для практического выполнения процесса моделирования.

Практикум детально знакомит будущих менеджеров, специализирующихся в области бизнес-информатики, с технологией моделирования архитектуры предприятия. В качестве программных средств моделирования бизнес-процессов используется система ADOit, нашедшая широкое применение для управления предприятиями во многих странах мира и для обучения менеджеров в университетах и бизнес-школах. Для выполнения заданий практикума используется бесплатная версия системы (common edition), которая по основным параметрам мало отличается от профессиональной. Выбор примеров моделирования архитектуры предприятия обусловлен актуальностью задач из различных областей деятельности менеджеров. Здесь рассматриваются задачи моделирования предприятий малого бизнеса, университета, банка, специализированных предприятий, ориентированных на информационные технологии.

Материал соответствует рабочей программе дисциплины «Моделирование архитектуры предприятия», входящей в учебный план подготовки студентов по направлению «Менеджмент» и отработан в процессе проведения лабораторных работ с системой в течение нескольких лет.

Автор выражает благодарность рецензентам профессорам Севастопольского государственного университета В.Я. Коппу и Ю.Е. Обжерину за ценные замечания, сделанные при рецензировании лабораторного практикума.

Введение

Настоящий практикум продолжает учебных пособий, предназначенных для изучения теории и практики моделирования бизнеса. В первой книге этой серии [7] рассматривались вопросы моделирования бизнес-процессов. Данный практикум расширяет вопросы моделирования бизнеса на задачи моделирования архитектуры предприятия.

Под архитектурой предприятия в книге понимается описание предприятия как системы, с ее целями, компонентами, отношениями компонент друг к другу и окружающей среде, а также основополагающие принципы, задающие ее деятельность и развитие [3,9].

Модель архитектуры предприятия предназначена для решения многих задач менеджмента. Первая группа задач – это анализ существующей системной организации предприятия, выявление узких мест, «белых пятен» и проблем. Вторая группа задач – это синтез оптимальной архитектуры предприятия. Здесь важными являются такие задачи, как оптимальное соотношение структурных элементов архитектуры предприятия и стратегии предприятия, согласование архитектуры информационной системы предприятия с архитектурой бизнеса, выяснение связей архитектуры предприятия и внешней среды предприятия и другие.

В качестве основы лабораторного практикума выбрана система моделирования архитектуры предприятия ADOit [10], разработанная австрийской фирмой BOC. Существуют две версии системы: коммерческая версия и бесплатная версия. Для лабораторного практикума используется бесплатная версия системы ADOit-CE].

Целью практикума является закрепление теоретических знаний, навыков и компетенций студентов, специализирующихся в области применения информационных технологий в менеджменте, в сфере моделирования бизнеса. путем реального моделирования архитектуры предприятия с помощью системы ADOit-CE. При выполнении практикума осваивается работа с хранилищем артефактов архитектуры на примере описания принципов архитектуры и принципов функционирования предприятия.

При выполнении задания практикума студенты получают практические навыки как в описании существующей архитектуры в организации (as-is), так и проектируемой архитектуры (as to be). При этом моделирование рассматривается не только как средство документирования архитектуры, а как аналитический инструмент для анализа и оптимизации архитектуры предприятия.

Процесс моделирования архитектуры предприятия включает два типа моделей: достаточно простых графических моделей на основе специальных нотаций моделирования и некоторых аналитических моделей типа моделей связей, пузырьковой диаграммы, графика Ганта.

Под нотацией моделирования понимается стандартная совокупность графических элементов, которые используются для построения бизнес-моделей деятельности организации [1,2]. В лабораторном практикуме используется нотация фирмы BOC, соответствующая стандарту TOGAF (The Open Group Architectural Framework), разработанному международной группой OMG (Object Management Group), специализирующейся на создании стандартов в области информационных технологий и организационного управления.

Наличие бесплатной версии системы дает возможность студентам использовать эту систему не только в университете, но также на своих домашних компьютерах и на компьютерах организаций во время практик.

Структурно учебное пособие состоит из 6 разделов.

В первом разделе приводятся описание и правила работы с системой ADOit. Студенты осваивают интерфейс системы и получают общие навыки работы с системой.

Во втором разделе рассматриваются архитектурные принципы и хранение их в системе. Приводятся упражнения и индивидуальные задания по описанию принципов и хранению их в системе.

В третьем разделе рассматривается моделирование организационной структуры предприятия. Изучаются различные виды организационных структур. Приводятся упражнения и индивидуальные задания по моделированию организационной структуры предприятия.

В четвертом разделе рассматривается архитектура бизнеса и в частности, модель создания ценности на предприятии. В том числе изучаются такие модели, как модель клиента, модель взаимодействия с клиентом, модель дистрибуции ценности, возможности бизнеса и функции бизнеса.

В пятом разделе изучаются возможности системы по моделированию архитектуры бизнес-процессов. Рассматриваются связи бизнес-процессов с другими элементами архитектуры бизнеса и архитектуры данных.

В шестом разделе изучается моделирование архитектуры информационной системы предприятия. Рассматриваются модели приложений, интерфейсов и инструментальных средств.

Глава 1. Начало работы с системой ADOit

1.1. Главное меню системы

Для выполнения лабораторного практикума используется система ADOit. Перед началом работ надо установить систему. При работе на своем домашнем компьютере можно установить систему непосредственно на свою основную машину или загрузить виртуальную машину с системой ADOit в систему виртуальных машин Oracle VM Virtual Box. Система виртуальных машин также является бесплатной системой [11]. Подготовленную копию виртуальной машины с предварительно установленной системой можно получить у преподавателя. Инсталляционный пакет можно скачать с сайта фирмы VOC [10]. Фирма также предлагает пробную версию коммерческого облачного варианта системы. Рекомендуется использовать виртуальную машину, так как система является достаточно сложной и использует систему управления базами данных MS SQL, что может привести к непредвиденным сбоям. В случае использования виртуальной машины всегда можно вернуться к исходному варианту. При работе системы на главной машине в случае сбоя может потребоваться вмешательство профессионального системного программиста.

При запуске системы стандартными средствами операционной системы Windows появляется стартовая страница, показанная на рис. 1.1.

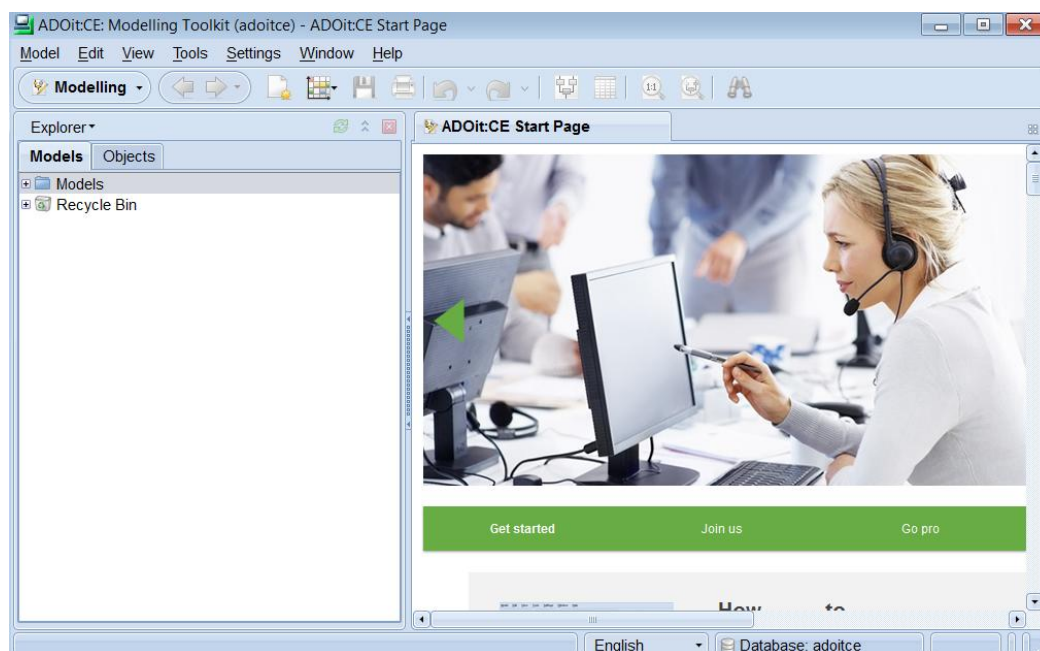


Рисунок 1.1 - Стартовая страница системы

Пользователю предлагается просмотреть рекламу и видеофильмы с интернета по работе с системой на английском языке. В коммерческой

версии возможен просмотр на других языках, но русского среди них на данный момент времени нет. Данную закладку удалить в бесплатной версии невозможно, но на этом месте открываются рабочие окна при моделировании и в дальнейшем она не мешает работе. Следует обратить внимание на главное меню в системе, показанное на рис. 1.2.



Рисунок 1.2 – Главное меню системы

Первый пункт меню задает режимы работы системы. На рис. 1.3 показан раскрывающийся список режимов.

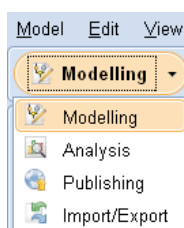


Рисунок 1.3 – Режимы работы системы

Система может работать в следующих четырех режимах: основной режим – режим моделирования архитектуры (Modelling); второй режим – режим анализа (Analysis). В этом режиме в данной версии системы можно формировать запросы к базе данных архитектуры; третий режим – режим сохранения результатов моделирования (Publishing). В этом режиме можно вывести результаты моделирования во внешние файлы; четвертый режим – режим импорта и экспорта результатов моделирования для передачи данных в другие совместимые системы (Import/Export).

Две зеленые стрелки позволяют перейти к другим открытым в настоящий момент режимам, окнам и формам. Раскрывающийся список на рис. 1.4 содержит информацию об активных модулях системы.

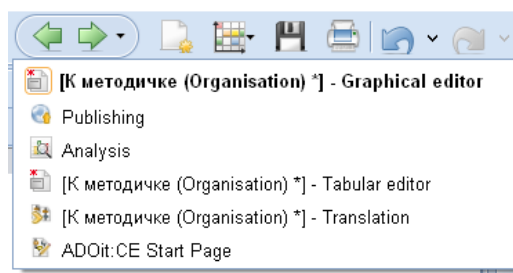


Рисунок 1.4 – Список активных модулей

Все рабочее пространство системы разбито на несколько окон. Открыть или закрыть соответствующее окно можно в пункте главного меню 'Window'. Открытые окна также закрываются обычным способом с помощью кнопки с крестиком в верхнем правом углу. Рабочее поле закрыть нельзя.

В первую очередь следует обратить внимание на окно 'Explorer', в котором показана система директориев (папок), с информацией об архитектуре предприятия. Директории разбиты на два следующих класса: директории моделей (Models) и директории объектов (Objects). Последний еще называется депозитарием архитектуры, то есть системой словарей и справочников всех объектов или сущностей архитектуры. Также эта часть модели архитектуры в стандарте называется хранилищем артефактов. Система поставляется с набором предустановленных папок и примерами файлов. В бесплатной версии изменять структуру папок с примерами нельзя, но их содержание можно изменять и копировать. На рис. 1.5 показан пример окна 'Explorer' с директориями для моделей архитектуры.

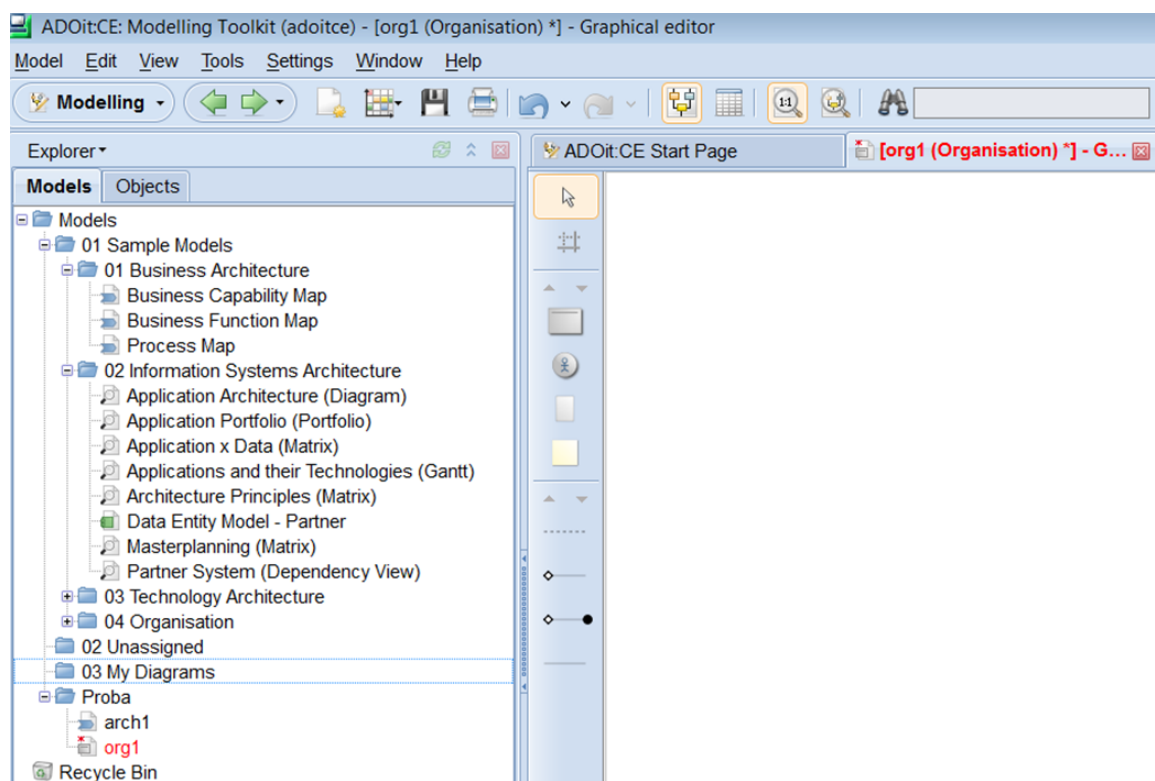


Рисунок 1.5 - Пример структуры директориев и моделей архитектуры

На рис. 1.6 показан пример окна с директориями хранилища артефактов. В основном здесь показаны директории демонстрационного примера.

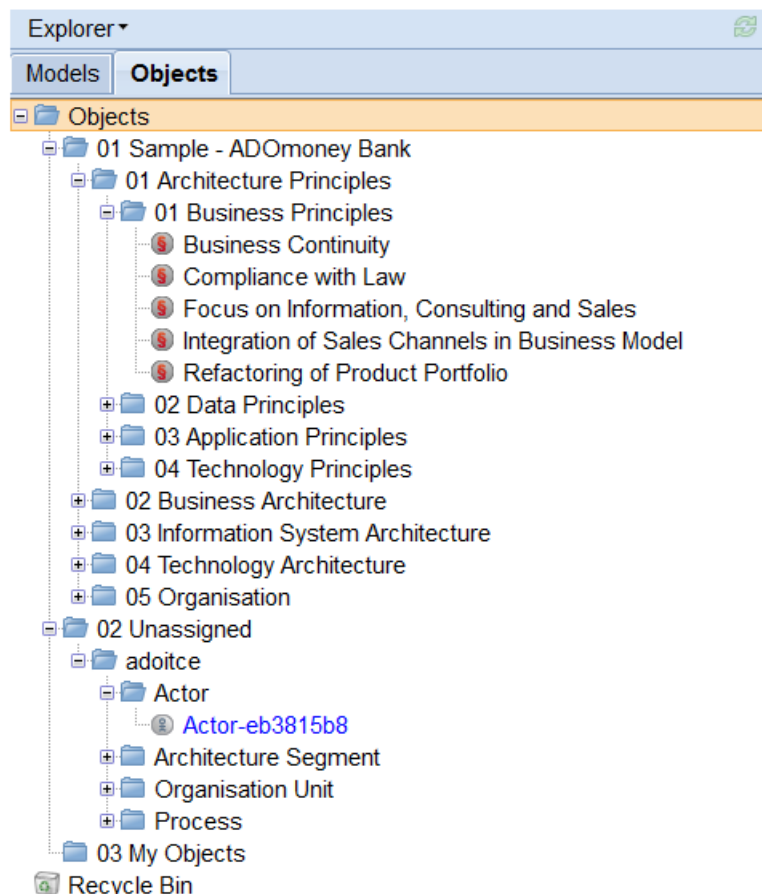


Рисунок 1.6 - Пример директориев и объектов архитектуры

На рис. 1.7 показаны оставшиеся кнопки панели быстрого доступа.



Рисунок 1.7 – Кнопки быстрого доступа

Первая слева кнопка в виде чистого листа бумаги служит для создания новой модели. Вторая кнопка с координатами и матрицей для формирования матричной модели. Эта кнопка и ее подменю будет рассматриваться отдельно в практикуме по построению матричных моделей архитектуры. Третья кнопка как обычно служит для сохранения модели в базе данных. Четвертая кнопка с изображением принтера как обычно для вывода модели на печать. Две кнопки с голубыми стрелками как обычно служат как обычно для перехода к предыдущему действию и к последующему действию. Кнопка с элементом графической схемы для перехода к графическому редактору. Кнопка с изображением таблицы для представления модели в виде элементов базы данных. Первая кнопка с лупой служит для

изменения масштаба изображения. Вторая кнопка с лупой служит для подгонки изображения модели к размеру рабочего окна. Последняя кнопка с изображением бинокля служит для поиска объектов модели в текущем изображении модели.

1.2. Создание, переименование и удаление папки

Для выполнения действий с папками надо нажать правую кнопку мыши на соответствующем месте существующей папки. В зависимости от того, показана папка в сжатом или раскрытом виде надпись в верхней части открывающегося окна может быть слово Expand (Раскрыть) или слово Collapse (Свернуть). В результате нажатия правой кнопкой мыши на открытой папке откроется окно с контекстным меню, показанное на рис. 1.8.

В заголовок автоматически в апострофах вставляется имя соответствующей редактируемой папки. На рисунке имя редактируемой папки - 'Models'. При нажатии первого пункта меню в зависимости от представления папки будет происходить либо сжатие, либо развертывание папки.

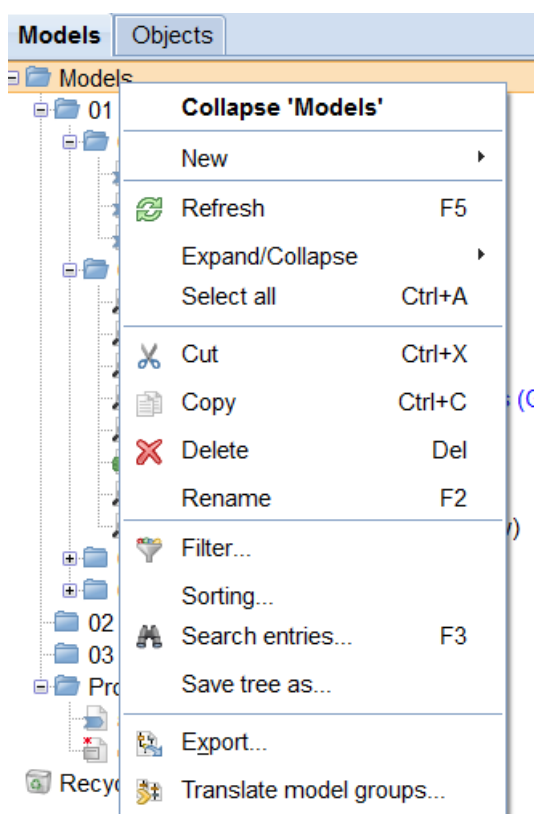


Рисунок 1.8 - Окно для действий над директориум

Меню содержит стандартные пункты работы с папками файлов, такие как «Вырезать», «Скопировать», «Удалить», «Переименовать» (Cut, Copy, Delete, Rename), не требующие пояснений.

Рассмотрим дополнительные пункты меню, введенные разработчиками системы ADOit-CE.

Refresh – обновляется содержание базы данных.

Expand/Collapse – раскрытие или сжатие директория. При нажатии на этот пункт меню открывается окно с новым контекстным меню, показанное на рис. 1.9.

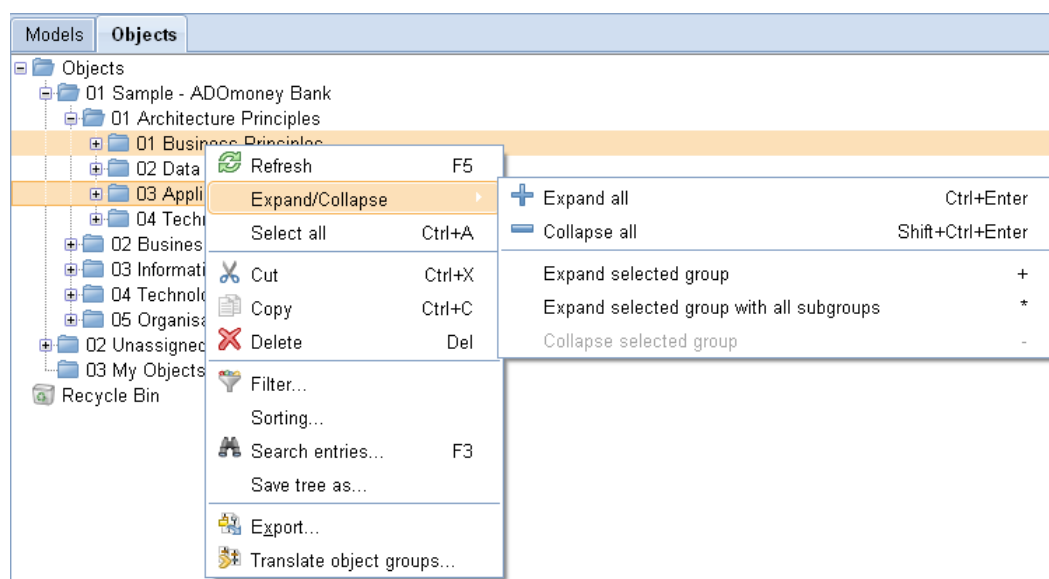


Рисунок 1.9 – Контекстное меню для представления папок

В этом случае возможно раскрытие всех папок директория Objects при выборе пункта меню Expand all, сжатие всех папок директория Objects при выборе пункта меню Collapse all, раскрытие одновременно всех отмеченных папок (Expand selected group), сжатие всех отмеченных папок (Collapse selected group) и раскрытие всех выбранных групп со всеми вложенными структурами (Expand selected group with all subgroups).

Select all – выбрать все. В этом случае будут отмечены все пункты директория.

Filter – включить фильтр данных по виду используемых моделей. При нажатии этого пункта меню открывается окно, показанное на рис. 1.10.

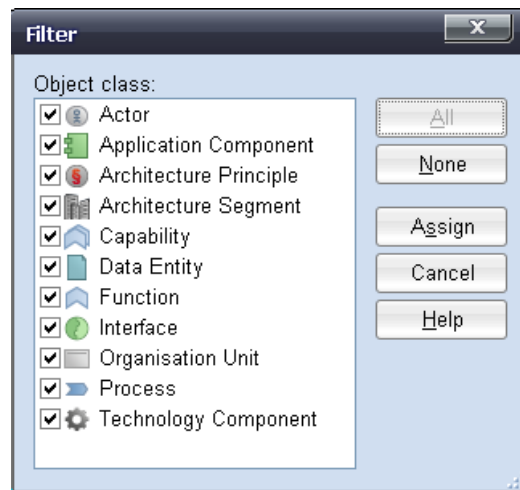


Рисунок 1.10 – Задание фильтра данных

Фильтр позволяет уменьшить число показываемых объектов в хранилище. При этом фильтр позволяет выбрать следующие классы объектов архитектуры предприятия:

- Actor – сотрудник, участник исполнения бизнес-процесса;
- Application Component – программное приложение;
- Architecture Principle – архитектурный принцип;
- Architecture Segment – архитектурный сегмент;
- Capability – возможность, потенциал предприятия;
- Data Entity – сущность базы данных;
- Function – функция предприятия;
- Interface – программный интерфейс;
- Organization Unit – организационная единица;
- Process – бизнес-процесс;
- Technology Component – технологический компонент.

Sorting – при выборе этого пункта меню открывается окно, показанное на рис. 1.11.

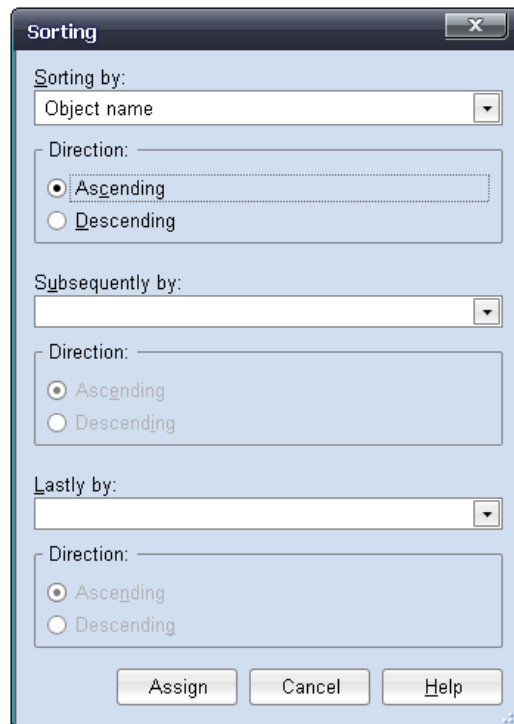


Рисунок 1.11 – Окно для сортировки представления папок

Имеется возможность отсортировать по алфавиту папки в представлении директория. В окне есть открывающийся список первичных объектов сортировки (Sorting by), как показано на рис. 1.12.

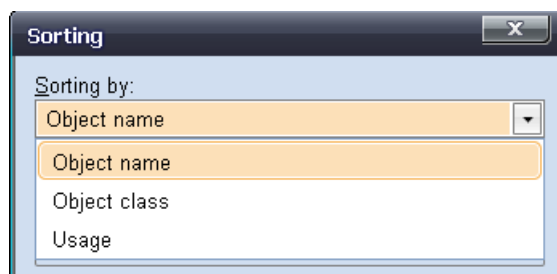


Рисунок 1.12 – Объекты сортировки

Сортировать можно по имени объекта (Object name) или по классу объектов (Object class) или по использованию (Usage). Можно сортировать по всем трем параметрам, указав в том числе второй параметр сортировки (Subsequently by) и последний параметр сортировки (Lastly by). При этом сортировка может осуществляться как по возрастанию кодов символов (Ascending), так и по убыванию (Descending).

Search entries – при выборе этого пункта меню открывается окно, показанное на рис. 1.13.

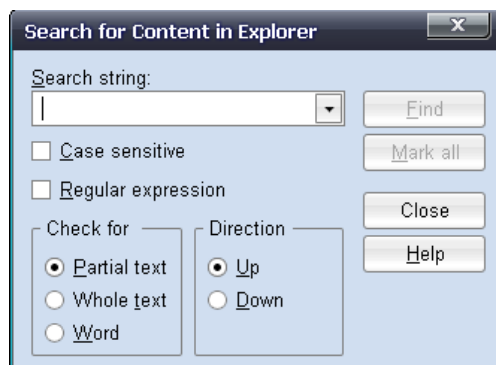


Рисунок 1.13 – Окно поиска информации по тексту

В системе имеется возможность поиска информации по текстовой строке (Search string). При этом можно указать учет регистра (Case sensitive), стандартное выражение (Regular expression), искать часть текста (Partial text), искать полный текст (Whole text) или искать отдельное слово (Word).

Save tree as – сохранить дерево директория в отдельном файле. При выборе этого пункта меню открывается стандартное окно записи файла, как показано на рис. 1.14.

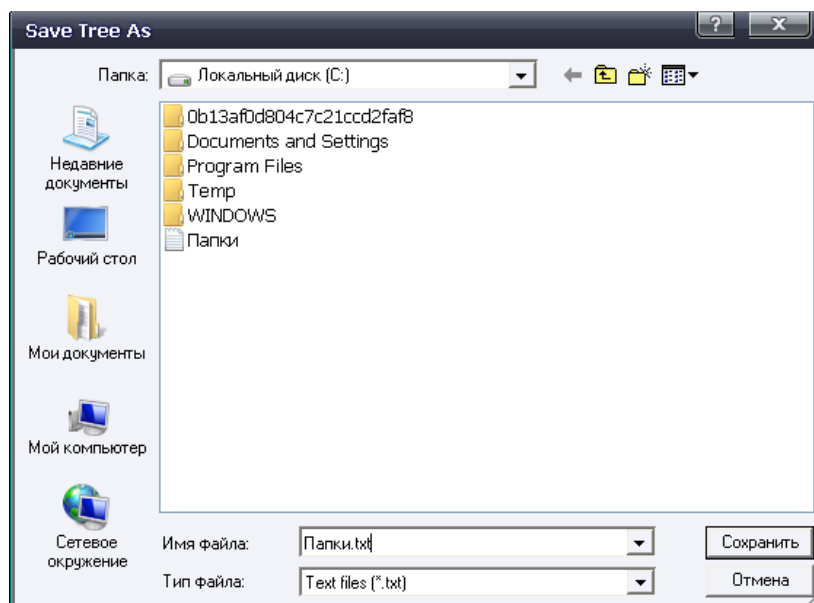


Рисунок 1.14 – Окно для сохранения файла со структурой директория

После выбора имени файла и его формата при успешном выполнении операции будет выдано сообщение, пример которого показан на рис. 1.15.

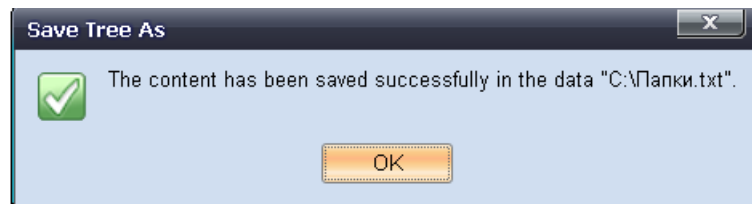


Рисунок 1.15 – Сообщение об успешном сохранении файла

Фрагмент тестового файла со структурой директория показан на рис. 1.16.

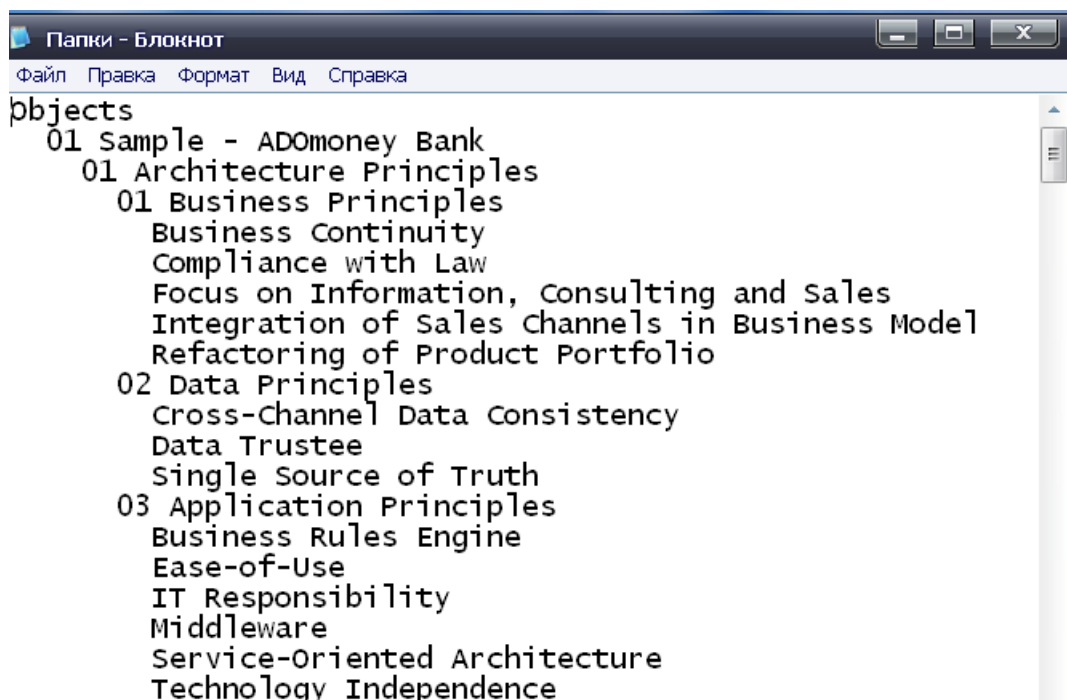


Рисунок 1.16 – Фрагмент файла со структурой директория

Export – при выборе этого пункта меню открывается окно, показанное на рис. 1.17.

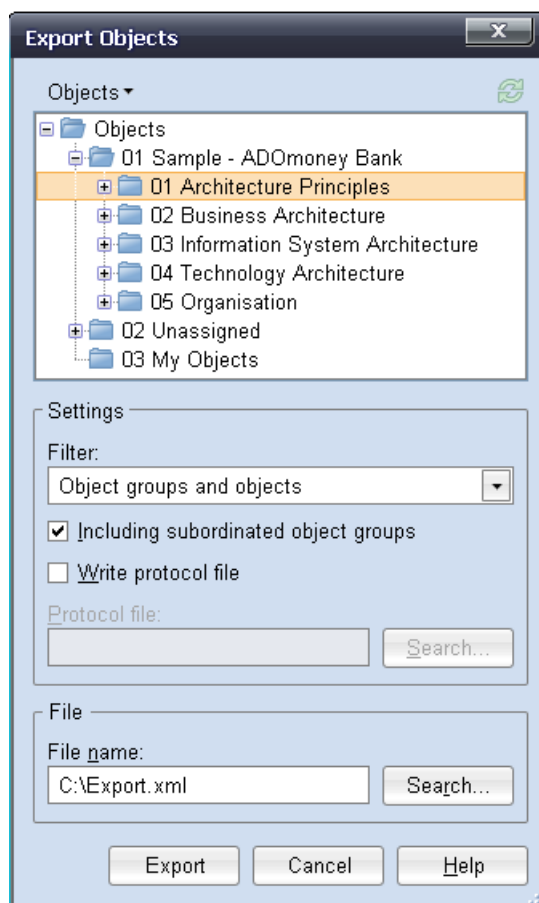


Рисунок 1.17 – Окно для сохранения xml файла

Все данные по артефактам сохраняются в виде xml файла и могут быть переданы в другие системы, которые распознают соответствующие формы документов. На рис. 1.18 показан пример сообщения об удачном сохранении файла.

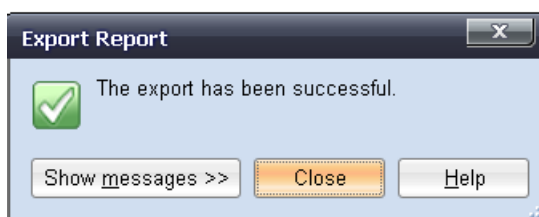


Рисунок 1.18 – Сообщение об удачном сохранении файла

Translate model groups – данный пункт меню используется для перевода терминов с одного языка на другой в рамках загруженных словарей языков в ADOit. На рис. 1.19 показан пример перевода терминов с английского языка на немецкий. Русский язык пока системой не поддерживается.

ADOit:CE Start Page		Object Groups Translation	
Object groups for translation:			
Model groups	Source language: English	Target language: German	Targe...
[-] Objects	Objects	Objekte	Objects
[-] 01 Sample - ADOmoney Bank	01 Sample - ADOmoney Bank	01 Beispiel-Repository - ADOmoney Bank	01 S...
[-] 01 Architecture Principles	01 Architecture Principles	01 Architekturprinzipien	01 Ar...
[-] 02 Business Architecture	02 Business Architecture	02 Geschäftsarchitektur	02 B...
[-] 03 Information System Arch...	03 Information System Architecture	03 Informationssystemarchitektur	03 Inf...
[-] 04 Technology Architecture	04 Technology Architecture	04 Technologiearchitektur	04 Te...
[-] 05 Organisation	05 Organisation	05 Organisation	05 Or...
[-] 02 Unassigned	02 Unassigned	02 Nicht zugeordnet	02 U...
[-] 03 My Objects	03 My Objects	03 Meine Objekte	03 M...
[-] Relations	Relations	Beziehungen	Relac...

Рисунок 1.19 – Пример перевода терминов

Для создания новой папки надо выбрать пункт "New", при этом открывается окно, показанное на рис. 1.20. В системе можно создавать любые новые папки. При этом существуют предопределенные папки, которые нельзя переименовать.

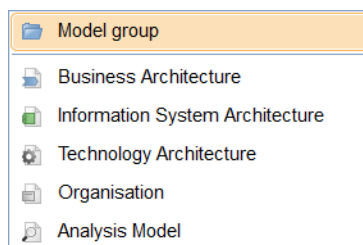


Рисунок 1.20 - Создание новой папки

Все папки разбиты по основным группам артефактов архитектуры предприятия. В данной версии системы возможно создать следующие папки:

Model group - группа моделей. При выборе этого пункта меню создается неопределенная новая папка;

Business Architecture - создание папки для артефактов бизнес архитектуры;

Information System Architecture - создание папки для артефактов архитектуры информационной системы;

Technology Architecture - создание папки для артефактов технологической архитектуры.

Organization - создание папки для артефактов организационной структуры предприятия;

Analysis Model - создание папки для результатов анализа. Это папка, создается для особых артефактов, специфичных для системы ADOit:

Если нажать на кнопку "Model group", то будет создана новая папка с рабочим именем 'Unassigned' (неопределенно), как показано на рис. 1.21.

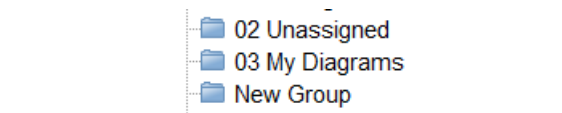


Рисунок 1.21 - Новая папка создана

Затем созданную папку можно переименовать (Пункт меню 'Rename' на рис. 1.8). Пример переименованной новой папки показан на рис. 1.22.



Рисунок 1.22 - Новая папка переименована

Глава 2. Лабораторный практикум «Архитектурные принципы»

2.1. Принципы в архитектуре предприятия

Термин «принцип» означает основу, первоначало, основное правило поведения или деятельности. Принципы должны быть короткими, легко запоминаться и использоваться для создания и улучшения архитектуры предприятия.

Принципы архитектуры предприятия можно разбить на два класса: принципы системной структуры предприятия и принципы моделирования архитектуры предприятия.

В качестве примеров общих системных принципов можно рассмотреть следующие принципы:

Принцип соразмерности. Соразмерность есть стройная гармония отдельных членов самого предприятия и соответствие отдельных частей и всего целого одной определенной части, принятой за исходную.

Принцип согласованности. Принцип предполагает, что частичное знание системы предприятия позволяет предсказать остальные ее части. Связан, конечно, с принципом соразмерности, но является более общим и носит алгоритмический или логический характер.

Принцип ортогональности. Функции различных элементов системы предприятия должны быть независимыми друг от друга. Функции не должны дублировать друг друга.

Принцип полноты. Архитектура предприятия должна наиболее полно описывать его функции.

В качестве примеров принципов моделирования архитектуры можно рассмотреть следующие принципы:

Принцип «сверху-вниз». Модель архитектуры должна носить блочный иерархический характер и начинать моделирование надо с самых общих блоков.

Принцип пользователя. Модель архитектуры предприятия создается для людей и их взаимодействия, коммуникации друг с другом, поэтому она должна быть понятна пользователям.

Принцип оптимальной сложности. Модель архитектуры не должна быть слишком простой, чтобы не потерять адекватность реальному предприятию и не должна быть слишком сложной, чтобы не потерять возможность понимания людьми.

Архитектура предприятия состоит из трех основных частей: архитектуры бизнеса, архитектуры информационной системы и архитектуры инфраструктуры. Каждая из этих частей строится на своих принципах.

В качестве примеров принципов бизнес-архитектуры можно рассмотреть следующие принципы:

Принцип законопослушания. Предприятие должно быть в согласии с законами, актами и внешней политикой государства. Предполагает связь с принципом постоянного повышения образованности сотрудников и доступ их к правилам и законам. Также предполагает связь с принципом постоянного отслеживания изменений в законах и официальных документах, которые должны отражаться в бизнес-процессах и технологиях.

Принцип непрерывности, постоянного роста бизнеса. Принцип тесно связан с принципами устойчивости бизнеса, учета рисков и развития собственности. Этот принцип предполагает, что не стоит менять бизнес, если он приносит прибыль.

Принцип учета рисков. Принцип предполагает, что заранее должны прогнозироваться риски остановки бизнеса и определяться мероприятия менеджмента по их предотвращению или уменьшению. Менеджмент должен рисковать разумно и периодически тестировать бизнес на уязвимость и незащищенность и разрабатывать мероприятия для поддержки функциональной непрерывности бизнеса. Является более общим принципом для принципа восстанавливаемости.

Принцип состязательности. Принцип предполагает постоянное создание конкурентных преимуществ, регулярный контроль за конкурентноспособностью. Принцип связан с принципом компромисса и как следствие с принципом превращения конкурентов в партнеров.

Принцип ориентированности на клиента. Бизнес должен быть направлен на пользу клиентов, увеличивать лояльность клиентов, должен быть ответственным, честным и бизнесу должны доверять. Принцип связан с принципом общественной значимости бизнеса, с принципом пользы заинтересованным лицам и принципом прозрачности для заинтересованных лиц.

Принцип эффективности бизнеса. Принцип отражает факт, что бизнес должен быть бережливым. Принцип связан с принципом рациональности бизнеса, принципом концентрации ресурсов на главных направлениях бизнеса, принципом развития креативности, принципом внедрения инноваций, принципом специализации и принципом дифференциации продукта. Из принципа вытекает следствие о постоянном совершенствовании предприятия.

В качестве примеров принципов архитектуры информационных систем можно рассмотреть следующие принципы:

Принцип безопасности. Информационные системы должны быть защищены как от внешнего вреда, так и от возможности разрушения от воздействия внешней среды.

Принцип законного сбора и распространения информации. Принцип тесно связан с принципом законопослушания в бизнес-архитектуре.

Принцип владения. Для всех данных должен быть четко обозначен их владелец. Все сущности в базах данных предприятия должны иметь одну систему идентификации. Тесно связан с принципом ответственности за данные, который предполагает, что ответственный владелец данных должен поддерживать возможности работы с данными всех допущенных пользователей, отвечать за качество данных и поддерживать ссылки на источники данных. Принцип также требует наличия на предприятии системы контроля качества данных.

Принцип доступности. Приложения предприятия должны быть доступны из любой точки Мира и поддерживать стандарты доступа. Предполагается легкость использования, дружелюбность приложений, прозрачность, эргономичность и понятность бизнес-пользователям. Приложения не должны требовать специального обучения в области информационных технологий. Язык интерфейсов должен быть русский.

Принцип стандартизации. Все новые приложения должны быть стандартизованы.

Принцип минимизации. Принцип предполагает минимизацию разнообразия технологических инструментов.

Принцип стандартизации. Использование одинаковых технологических программных и инженерных устройств на предприятии и постоянную актуализацию стандартов.

Принцип мобильности. Приложения должны поддерживаться на мобильных устройствах.

Принцип сервис-ориентированной архитектуры. Использовать центральный сервер для уменьшения дублирования приложений.

В качестве примеров принципов построения технической архитектуры можно рассмотреть следующие принципы размещения предприятия:

Принцип наличия трудовых ресурсов. Принцип требует, чтоб в месте расположения предприятия имелось в наличии требуемое число работников нужной квалификации, согласных с уровнем зарплаты на предприятии. Также важно наличие соответствующих жилищных условий, городского или специализированного транспорта, социальной инфраструктуры. Если это отсутствует, то необходимо рассмотреть возможность перемещения персонала, его обучения и создания соответствующих условий быта.

Принцип обеспечения сырьем и материалами. Принцип требует наличия в месте расположения предприятия сырья и материалов или транспортных коммуникаций, позволяющих достаточно эффективно доставлять эти ресурсы.

Принцип наличия транспортных коммуникаций. Этот принцип требует учета наличия автомобильных и железных дорог. Важно наличие морских и речных портов. Особенно, если речь идет о производстве круп-

нотоннажных изделий. Может иметь большое значение близость к аэродрому.

Принцип наличия развитой местной инфраструктуры. Здесь надо отметить учет наличия газа, электричества, воды, систем дренажа и утилизации отходов. Важно учитывать качество местной воды, ее доступность и степень чистоты. Важно также учесть наличие банковских, рекламных и других организаций, которые могут оказать услуги предприятию.

Принцип государственного регулирования. Важными факторами, влияющими на выбор места предприятия, являются факторы государственного регулирования, относящиеся к региону, такие как правительственная помощь, дотации для новых сооружений и рабочих мест, предоставление займов на благоприятных условиях, освобождение от налогов, доплаты работникам, скидки за обучение определенных категорий работников.

Принцип конкурентоспособности. При размещении предприятия необходимо предусматривать наличие конкурентов.

2.2. Создание папок с принципами бизнеса

Многие артефакты, созданные для объектов в модели архитектуры, создаются в хранилище системы автоматически. Для принципов необходимо создать папки для их хранения и описать сами принципы.

Упражнение. Необходимо создать две новые папки для выполнения лабораторного практикума. Первая папка создается для артефактов модели архитектуры с именем следующей структуры "Фамилия студента"- модель. Например, Иванов-модель. Если на этом компьютере уже имеется папка с данным именем, то к фамилии надо добавить цифру. Вторую папку надо создать для артефактов (объектов архитектуры). Структура имени папки - "Фамилия студента" - объекты. Затем надо создать папки для вновь создаваемых архитектурных принципов, как показано на рис. 2.1.

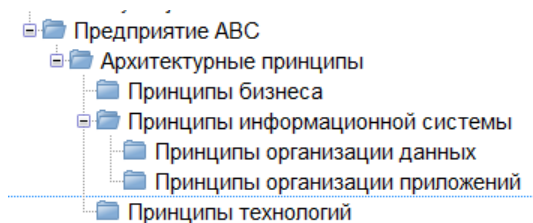


Рисунок 2.1 - Создание системы новых папок

В качестве шаблона нового принципа можно использовать пример модели архитектуры, поставляемый вместе с системой. Для этого надо открыть папки поставляемого примера архитектуры, как показано на рис. 2.2.

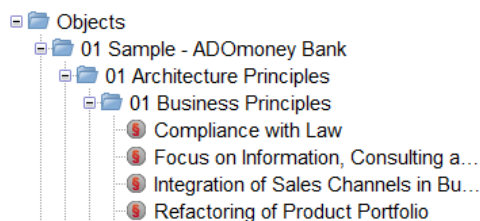


Рисунок 2.2 - Примеры файлов с принципами бизнеса

Скопировать в буфер любой из имеющихся принципов. Для этого щелкнуть правой кнопкой мыши на объекте. Например, на объекте 'Compliance with Law' (Соответствие с законом). Откроется окно с меню возможных действий с данными, как показано на рис. 2.3.

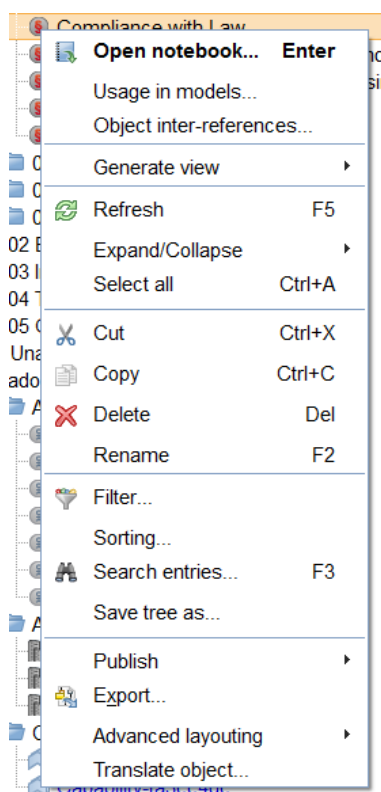


Рисунок 2.3 - Меню возможных действий с элементом архитектуры

В данном меню есть новые пункты. Пункт меню Open notebook – открыть блокнот. При выборе этого пункта меню открывается разработанное фирмой ВОС приложение для ввода, удаления и коррекции данных объекта моделирования. Это приложение по форме похоже на блокнот.

Следующие пункты меню будут подробно разбираться в дальнейшем:

Usage in models – использование в модели;

Object inter-references – внутренние ссылки на объект;

Generate view – сгенерировать взгляд на объект.

Advanced layouting – расширенные функции представления.

Пункт меню Publish (Опубликовать данные) открывает окно, показанное на рис. 2.4.

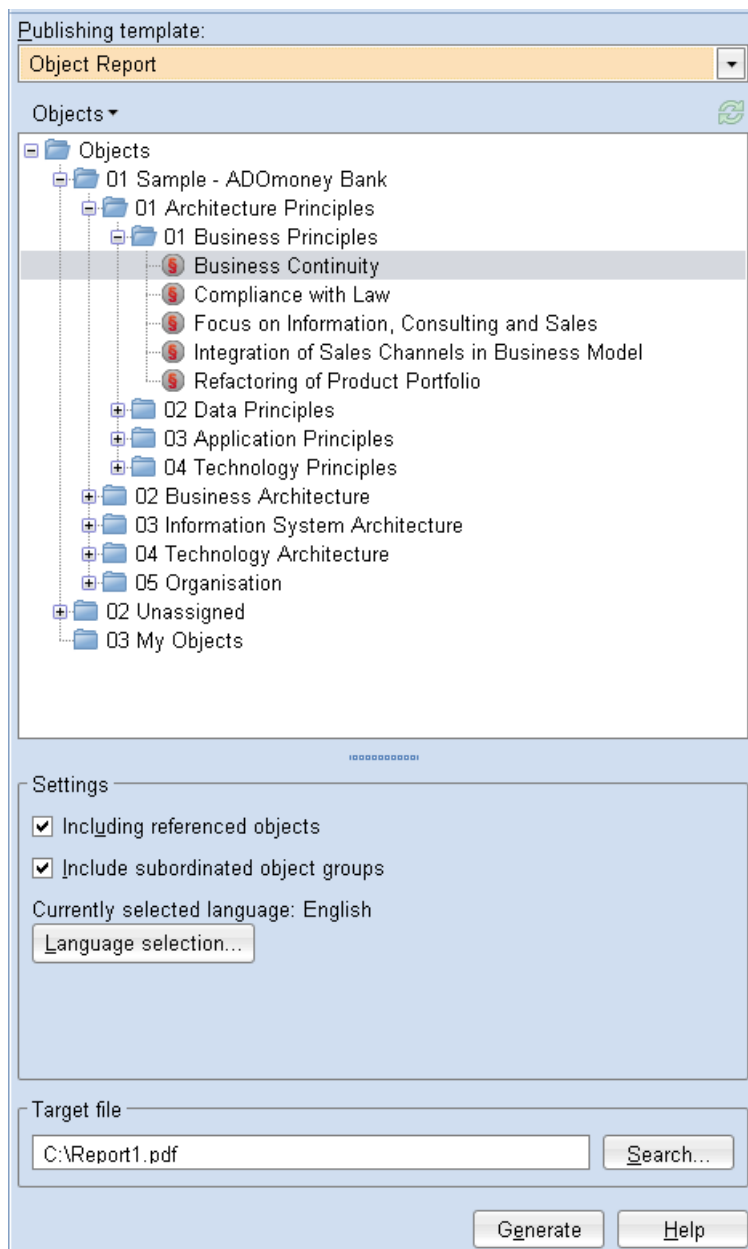


Рисунок 2.4 – Вывод в файл отчета по данным объекта архитектуры

В формируемый отчет можно включить связанные объекты (Отметив галочкой поле «Including referenced objects») и иерархически связанные группы объектов (Поле «Include subordinated object groups»). Необходимо задать имя файла, в котором сохраняется отчет и нажать кнопку **Generate**.

В результате в файле будет сформирован отчет по выбранному объекту архитектуры. Фрагмент файла отчета показан на рис. 2.5.

Objects of type 'Architecture Principle'

Business Continuity

Architecture Principle

General Information	
Name	Business Continuity
Statement	Enterprise operations are maintained in spite of system interruptions.
Implications	* Dependency on shared system applications mandates that the risks of business interruption must be established in advance and managed. Management includes but is not limited to periodic reviews, testing for vulnerability and exposure, or designing mission-critical services to assure business function continuity through redundant or alternative capabilities. * Recoverability, redundancy, and maintainability should be addressed at the time of design. * Applications must be assessed for criticality and impact on the enterprise mission, in order to determine what level of continuity is required and what corresponding recovery plan is necessary.
Priority	No Entry
Action Required	No
Lifecycle	
Valid From	---,---,----
Valid Until	---,---,----
Change History	
Date of Last Excel Import	---,---,----
Attachments	
Attachments	This table attribute does not contain any entries.
Representation	
Display as Icon	Yes

Рисунок 2.5 – Фрагмент файла отчета

Для копирования принципа надо нажать на пункт меню 'Copy' и перейти к созданной папке "Принципы бизнеса". Нажать правой кнопкой мыши на этой папке. При этом откроется меню, рассмотренное в предыдущей главе. Выбрать пункт меню 'Paste' (Вставить). В результате в папку из буфера будет вставлена копия записи принципа из примера, как показано на рис. 2.6.

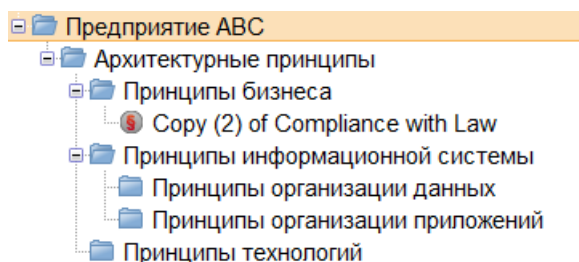


Рисунок 2.6 - Вставка копии элемента архитектуры

При попытке создать уже существующее имя принципа система выдаст предупреждение, как показано на рис. 2.7. В предупреждении сказано: «Объект хранилища «Главный принцип бизнеса» уже существует. Вы желаете заменить текущий объект «Главный принцип бизнеса» на уже существующий объект «Главный принцип бизнеса» или ввести другое имя объекта?»

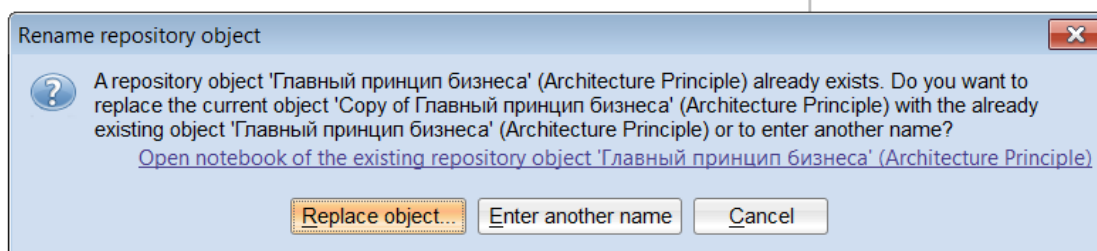


Рисунок 2.7 - Предупреждение об уже имеющемся объекте с данным именем

Здесь возможно три варианта: заменить объект (Кнопка **Replace object..**), вставить объект с другим именем (Кнопка **Enter another name**) или отменить операцию (Кнопка **Cancel**).

Если выбрать кнопку **Replace object...**, то система предложит два варианта замены, как показано с помощью инфограммы на рис. 2.8. В первом варианте можно оставить объект в предыдущем директории, как было без изменений (Копия объекта удаляется). Во втором варианте удалить объект из предыдущего директория (Сохраняется только копия объекта). Следует учесть, что удалить из директория примеров в бесплатной версии невозможно.

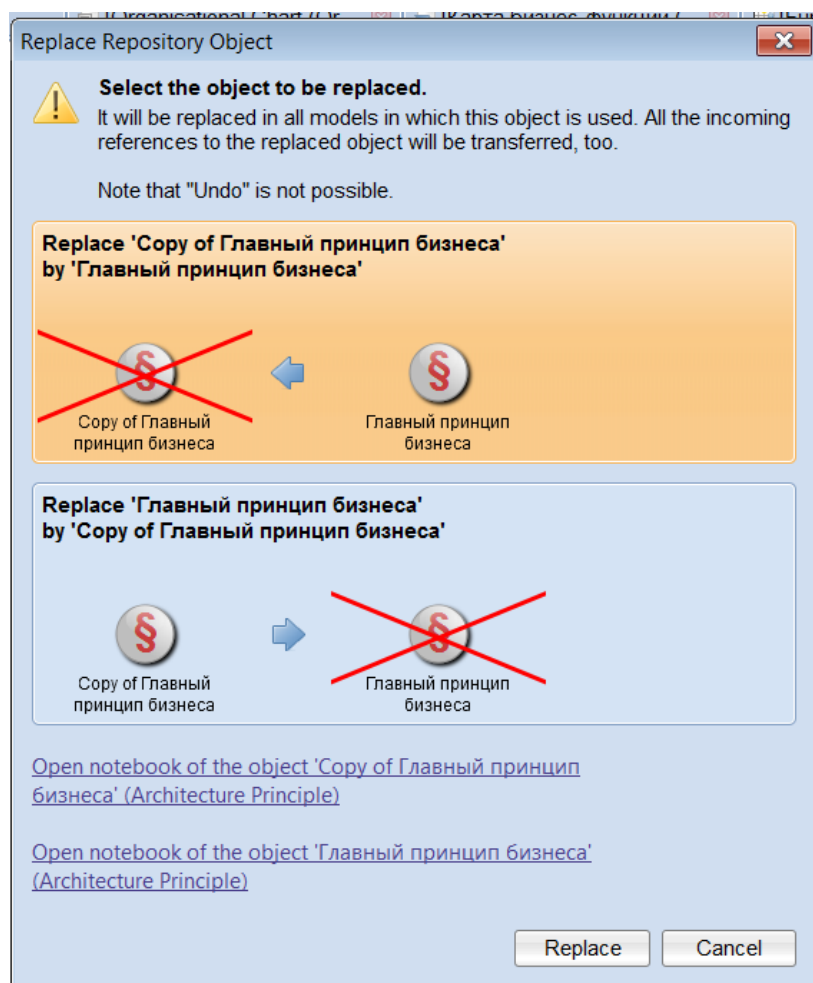


Рисунок 2.8 - Варианты замены имени объекта

2.3. Описание принципа

При двойном щелчке на созданном файле или с помощью меню, рассмотренном в предыдущем разделе открывается "блокнот", как показано на рис. 2.9. Организация хранения информации об объекте соответствует общей методологии фирмы ВОС, которая ориентирована на используемую реляционную СУБД (Как правило, MS SQL).

Рисунок 2.9 - "Блокнот" для информации об объекте "Принцип"

Рассмотрим информационные поля на закладке общего описания объекта (General Information). В верхней части закладки общего описания расположены следующие текстовые поля:

- Name - краткое название принципа;
- ID - кодовое обозначение принципа;
- Statement – краткое описание принципа в виде одного предложения;
- Rational – аргументация необходимости введения данного принципа;
- Implications – результаты и возможные последствия введения принципа;
- Risks of implementation - риски применения принципа;
- Risks of Non-implementation - риски не использования принципа.

Нижняя часть описания общей информации о модели принципов показана на рис. 2.10. Для перехода на эту часть используется движок прокрутки справа.

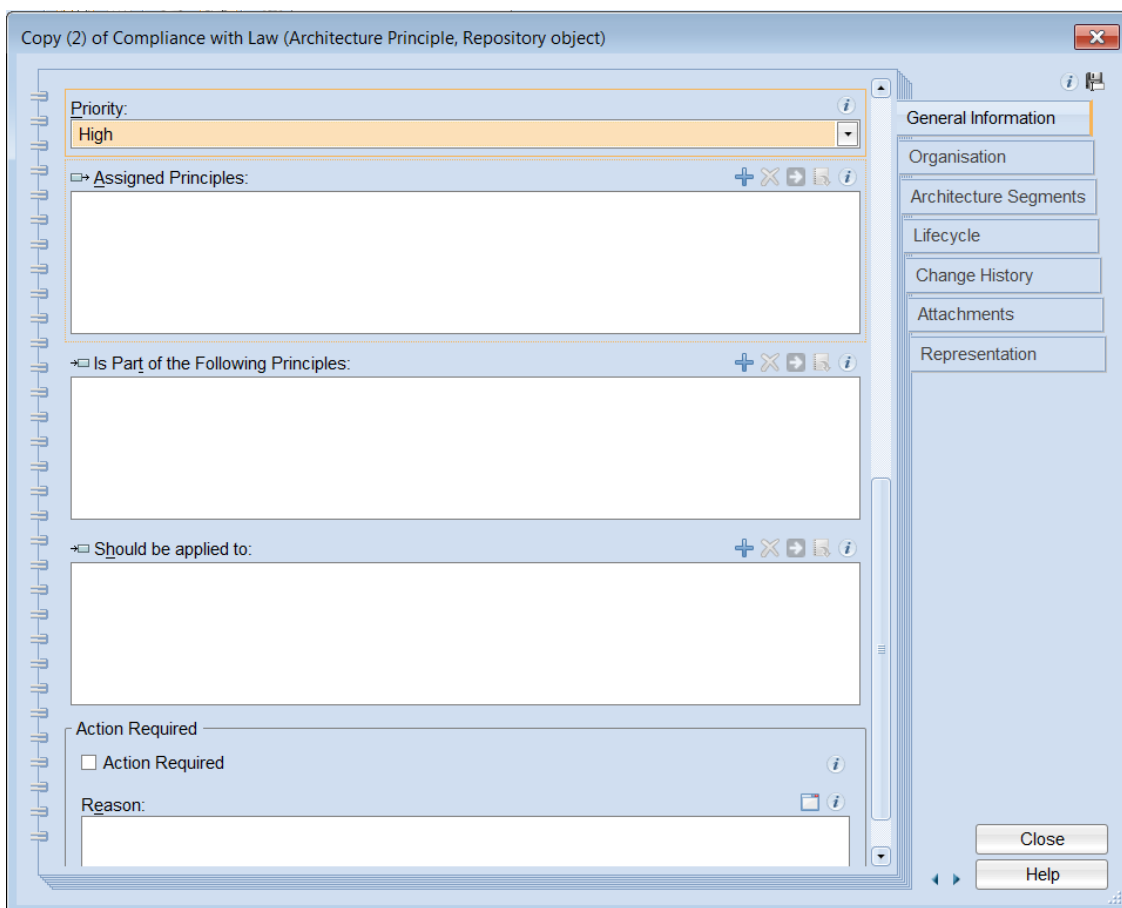


Рисунок 2.10 - Нижняя часть блокнота с общим описанием принципа

Здесь используются следующие информационные поля:

Поле Priority - приоритет принципа. Принципы могут быть упорядочены по своей важности (приоритету). При этом в системе возможны уровни приоритета, которые задаются открывающимся списком (Кнопка с крестиком), показанным на рис. 2.11.

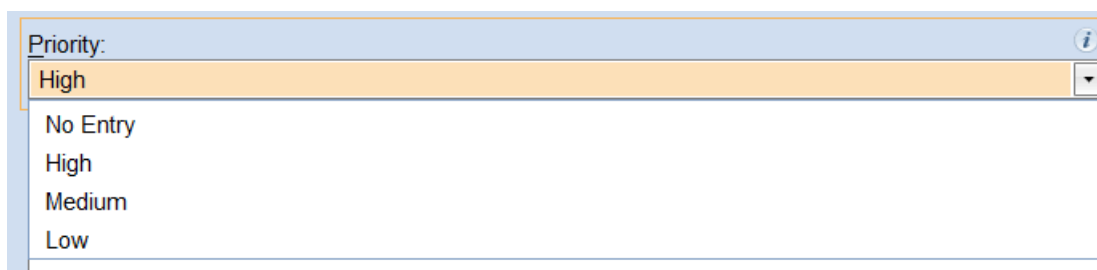


Рисунок 2.11 - Список с уровнями приоритета.

Возможно задать следующие уровни приоритета:

No Entry - приоритет не указывается; High - высокий приоритет; Medium - средний приоритет; Low - низкий приоритет.

Поле Assigned Principles – содержит ссылки на принципы, связанные с данным принципом. При нажатии кнопки с крестиком перед этим полем открывается часть директория с принципами, как показано на рис. 2.12.

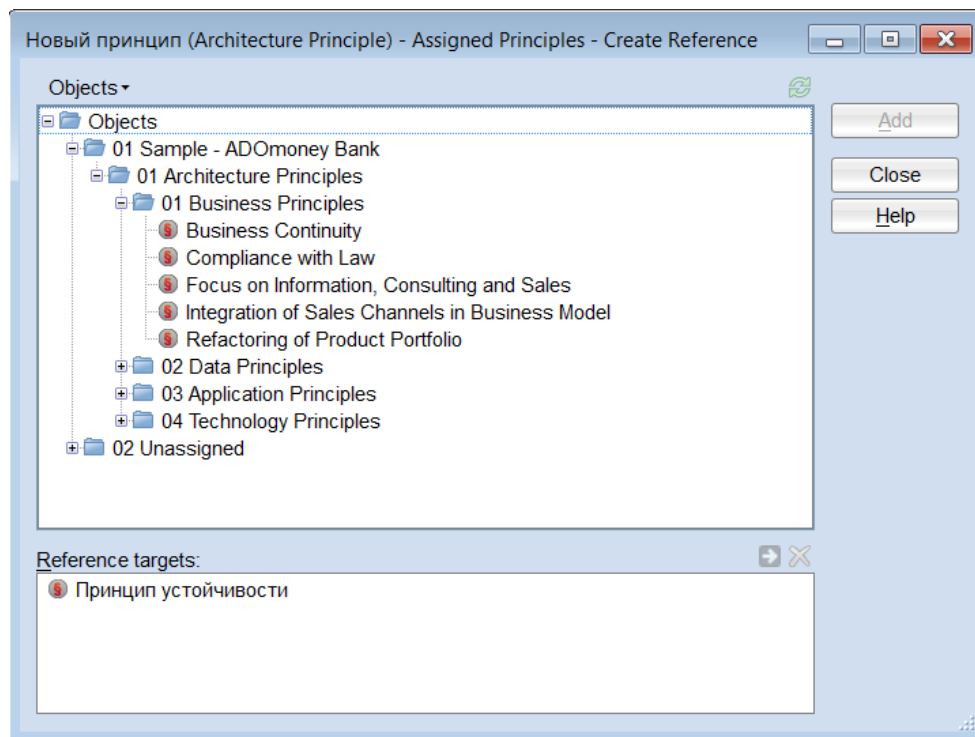


Рисунок 2.12 - Выбор связанного принципа

Поле Is part of the Following Principles - данный принцип является частью другого принципа.

Поле Should be applied to - здесь указываются ссылки на объекты архитектуры, для которых применяется данный принцип, такие как персонал, бизнес-процессы, бизнес-единицы и т.д.

Поле Action Required - здесь ставится маркер и дается описание того факта, что для реализации принципа требуется определенное действие.

Пример заполнения полей первой части главной закладки показан на рис. 2.13.

Принцип устойчивости (Architecture Principle, Repository object)

Name: Принцип устойчивости

ID: 12345

Statement: Принцип устойчивости бизнеса, то есть способность бизнеса (предприятия) выдерживать неблагоприятные воздействия, предоставлять важные услуги в нормальном режиме, без перерывов - в соответствии с оговоренным уровнем сервиса

Rationale: Использование принципа ведет к надежности предприятия.

Implications: В результате предприятие и его владельцы и сотрудники чувствуют уверенность в будущем, а клиенты больше доверяют предприятию.

Risks of Implementation: Риски за счет снижения гибкости и консерватизма пропустить вперед конкурентов.

Risks of Non-Implementation: Риски уменьшения жизненного цикла предприятия.

Close Help

Рисунок 2.13 - Пример заполнения первой части общего описания принципа

Пример заполнения второй части главной закладки показан на рис. 2.14.

Принцип устойчивости (Architecture Principle, Repository object)

Priority: High

Assigned Principles:

- Забота о сотрудниках
- Согласие с законом
- Честность

Is Part of the Following Principles:

- Главный принцип бизнеса

Should be applied to:

- Бухгалтерия

Action Required:
☐ Action Required

Reason:

Close Help

Рисунок 2.14 - Пример заполнения второй части общего описания принципа

В блокноте также имеются следующие закладки.

Закладка Organization – содержит связи принципа с организационной структурой. Эта закладка будет рассматриваться в лабораторной работе по построению модели организационной структуры.

Закладка Architecture Segments – содержит связи принципа с сегментами архитектуры. Эта закладка будет подробно рассматриваться в лабораторном практикуме по построению модели сегментов архитектуры бизнеса.

Закладка Lifecycle – содержит описание жизненного цикла принципа. Пример закладки показан на рис. 2.15.

The screenshot shows a software window titled 'Основной принцип бизнеса (Architecture Principle, Repository object)'. The 'Lifecycle' tab is selected in the right-hand sidebar. The main area contains a 'Valid From' date field set to '01.09.2015' and a 'Valid Until' date field set to '02.09.2020', both with red 'X' icons indicating errors. Below these is a text area labeled 'Additional Information on Lifecycle:'. The sidebar also includes tabs for 'General Information', 'Organisation', 'Architecture Segments', 'Change History', 'Attachments', and 'Representation'. At the bottom right are 'Close' and 'Help' buttons.

Рисунок 2.15 - Закладка с информацией о жизненном цикле принципа

Здесь задается срок действия принципа. В поле Valid From: (Используется с даты) задается начальная дата применения принципа. В поле Valid Until: (Используется до) задается конечная дата применения принципа. В поле Additional Information on Lifecycle (Дополнительная информация о жизненном цикле) можно указать в текстовом виде пояснения к жизненному циклу. Для принципов бизнеса жизненный цикл не актуален, так как эти принципы должны быть длительными, но для других принципов, например, для разработки архитектуры информационной системы они могут быть более короткими.

Закладка Change History - автоматически отражает все изменения, которые были сделаны с записью об объекте архитектуры. На рис. 2.16 показан пример содержания закладки.

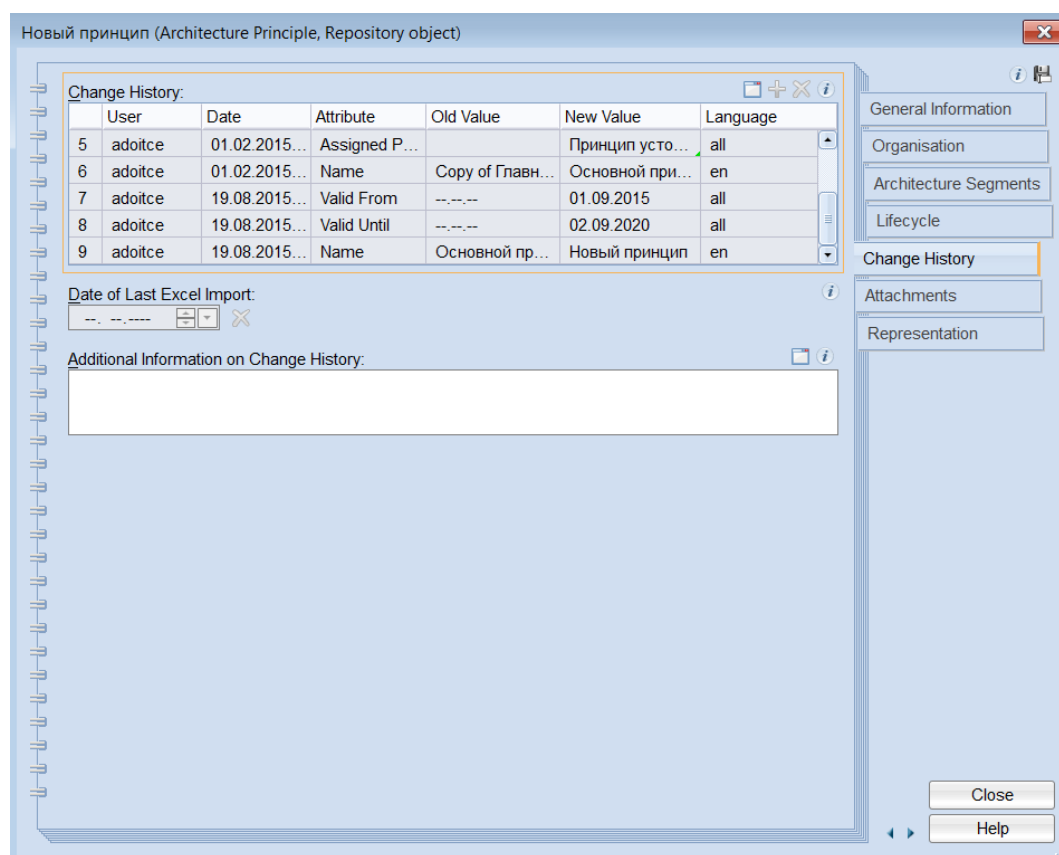


Рисунок 2.16 - Закладка "Change History"

Поле Change History содержит таблицу с изменениями атрибутов. В таблице формируются следующие столбцы: User – кто произвел изменения, Date – дата и время изменения, Attribute – наименование атрибута объекта, Old Value – старое значение атрибута, New Value – новое значение атрибута, Language – используемый язык. Так в примере на рис. 37 старое значение наименования принципа Copy of... заменено на новое значение «Основной принцип».

В коммерческой версии системы возможен импорт данных из файла Excel. Для этого введено поле “Date of Last Excel Import”.

В поле “Additional Information on Change History” можно ввести пояснения к истории изменений атрибутов.

Закладка Attachments (Приложения) – содержит ссылки на файлы, в которых может быть дополнительная информация об объекте или необходимые программные приложения. Пример закладки показан на рис. 2.17.

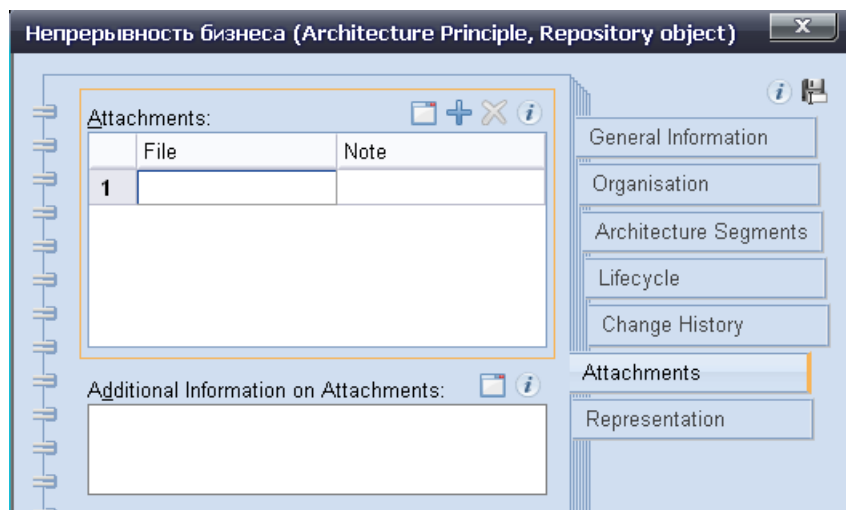


Рисунок 2.17 - Закладка “Attachments”

Закладка Representation – в данной версии системы для объекта «Принцип» имеет только одно значение – представление в виде иконки, и практического значения не имеет.

2.4. Задание по выполнению практикума

1. Выполнить упражнение, описанное в методичке.
2. Описать пять принципов бизнеса предприятия согласно варианту. Примеры принципов заданы в параграфе 2.8. Задать связи между принципами.
3. Сохранить отчет с описанным принципом в отдельном pdf файле с помощью пункта меню Publish (Опубликовать).

2.5. Содержание отчета

Отчет должен содержать следующую информацию: задание на выполнение практикума, копии окон экрана с выполненными упражнениями, копии окон экрана с выполненным индивидуальным заданием, выводы.

2.6. Варианты заданий

№	Принципы	№	Принципы	№	Принципы
1	1, 5, 7, 9,12	11	25,26,27,28,29	21	22,23,24,25,26
2	2,3,4,8,32	12	1,5,9,13,17	22	23,25,27,28,30
3	3,10,11,14,30	13	2,6,10,14,18	23	19,25,29,31,32
4	2,6,8,10,13	14	3,7,11,15,20	24	1,32,2,31,3
5	3,7,9,11,15	15	5,9,13,17,21	25	2,31,3,30,4
6	4,8,12,16,20	16	10,14,18,22,26	26	3,30,4,29,5
7	5,6,7,8,9	17	13,17,21,25,29	27	4,29,5,28,6
8	10,11,12,13,14	18	17,19,21,23,25	28	5,28,6,27,7
9	15,16,17,18,19	19	19,21,23,25,27	29	15,16,17,14,13
10	20,21,22,23,24	20	20,23,26,29,31	30	12,21,11,20,19

2.7. Контрольные вопросы

1. Назначение системы **ADOit**.
2. Что такое виртуальная машина?
3. Что такое артефакт архитектуры?
4. Почему при создании принципа необходимо предварительно скопировать уже имеющийся принцип?
5. Привести пример двух взаимосвязанных принципов. Задать их модель в системе **ADOit**.
6. Привести пример двух иерархически связанных принципов. Задать их модель в системе **ADOit**.
7. Изменить имя одного из созданных в лабораторной работе принципов и показать предысторию изменений.
8. Привести пример последствий, если не учитывать принцип риска в архитектуре предприятия.

2.8. Примеры принципов бизнеса

№	Принцип
1	Бизнес должен устойчиво расти
2	Бизнес должен быть прозрачным для заинтересованных лиц
3	Бизнес должен быть направлен на пользу клиентов
4	Бизнес должен иметь запас устойчивости
5	Бизнес должен давать чувство удовлетворения
6	Бизнес должен быть честным
7	Бизнесу должны доверять
8	Бизнес должен быть основан не только на интуиции, но и на аналитике
9	Максимально использовать информационные технологии
10	Быть всегда на шаг впереди конкурентов
11	Превращать конкурентов в партнеров
12	Бизнес должен быть бережливым
13	Бизнес должен быть направлен на пользу общества
14	Постоянный контроль за бизнесом
15	Повышать мотивацию персонала
16	Специализироваться
17	Постоянно учиться и учить персонал
18	Не менять бизнес, если он приносит прибыль
19	Рисковать разумно
20	Постоянно совершенствовать организацию
21	Создавать конкурентные преимущества
22	Бизнес должен быть эффективным
23	Беречь свою конкурентно-способность как зеницу ока
24	Диверсифицировать свой продукт
25	Кадры решают все
26	Бизнес должен быть точным
27	Бизнес должен отвечать требованиям экологии
28	Создавать новые рабочие места
29	Соблюдать законы
30	Бизнес должен быть устойчивым
31	Фокусироваться на информированности, консалтинге и продажах
32	Реструктурировать портфель продуктов

Глава 3. Лабораторный практикум «Моделирование организационной структуры предприятия»

3.1. Виды организационной структуры предприятия

Архитектура предприятия во многом определяется его организационной структурой. Организационная структура предприятия — это совокупность бизнес-единиц, персоналий и связей между ними [5,6].

Линейно-функциональная структура - наиболее распространенный тип организационной структуры. В этом случае предприятие делится на функциональные подсистемы, такие как отдел маркетинга, производство, отдел исследований и разработок, управление финансами, управление персоналом и т. д. По каждой подсистеме управления формируется иерархия других подсистем, пронизывающая все предприятие сверху донизу. К преимуществам линейной структуры следует отнести: жесткую систему управления и соответствующих подразделений, наличие единоначалия, хорошо выраженную ответственность руководителей. К недостаткам такой системы надо отнести: слабую гибкость и адаптивность, перекладывание ответственности при решении проблем на другие подразделения, низкую скорость реагирования на изменение ситуации.

Линейно-штабная иерархия — включает в себя кроме функциональных подразделений специальные подразделения (штабы). Штабы не принимают управленческие решения, а только занимаются планированием и анализом. В остальном эта структура соответствует линейной. Преимущества такой структуры: качественное планирование, глубокая проработка стратегических вопросов и возможность привлечения внешних консультантов и экспертов. К недостаткам, кроме совпадающих с линейной структурой, можно отнести недостаточно четкое распределение ответственности (поскольку лица, готовящие решение, не участвуют в его выполнении).

Дивизионные структуры управления — в этом случае производится структуризация бизнес-единиц по дивизионам. Структуризация может проводиться: по выпускаемой продукции (изделиям или услугам), по ориентации на определенные группы потребителей или по обслуживаемым территориям (региональная специализация). К преимуществам дивизионной структуры относятся: возможность управления многопрофильными предприятиями с общей численностью сотрудников порядка сотен тысяч человек и территориально удаленными подразделениями; большая гибкость и более быстрая реакция на изменения по сравнению с линейной и линейно-штабной, при этом устанавливается более тесная связь производства с потребителями. К недостаткам такой структуры можно отнести: большое количество ступеней управленческой вертикали, разобщенность отделений и штаба компании, дублирование функций на разных ступенях.

Процессные структуры. Такие организационные структуры построены на принципе множественного подчинения исполнителей. С одной стороны, они подотчетны непосредственному руководителю функциональной бизнес-единицы, которая предоставляет персонал, ресурсы и техническую помощь при создании ценности для потребителя, с другой — руководителю проекта или целевой программы, менеджеру продукта, владельцу бизнес-процесса, который наделен необходимыми полномочиями для осуществления процесса управления. Часто такие структуры называют матричными, проектными, процессными системами управления. Такие структуры имеют следующие достоинства: ориентация на целевые показатели (продуктовые, проектные или процессные); возможность снижения расходов и повышения эффективности использования ресурсов; гибкое и эффективное использование персонала организации, специальных знаний и компетентности сотрудников; повышение мотивации персонала; эффективный обмен информацией. К недостаткам такой системы управления надо отнести следующее: сложность такой системы управления; необходимость высокой подготовки менеджеров в области теории управления и владения компьютерными системами управления предприятием; возникновение конфликтов между линейными руководителями и руководителями процессов; увеличение административных расходов; необходимость постоянного улучшения и изменений системы управления; психологическая сложность для работы сотрудников с двойным, а иногда и множественным подчинением (Сотрудник может участвовать в нескольких бизнес-процессах).

Адаптивные организационные структуры. Предприятия с такими структурами могут менять свою модель управления в зависимости от изменения рынка или государственного регулирования. В основе таких систем могут лежать как функциональные системы управления, так и процессные системы управления. Примерами таких систем являются бригадная организационная структура и проектная организационная структура. В этом случае для решения актуальных задач, стоящих перед руководством предприятия могут создаваться временные коллективы, бизнес-единицы, проектные команды, творческие коллективы и другие, самые необычные структуры. К достоинствам таких систем можно отнести следующее: скорость реагирования на запросы клиентов, эффективное использование ресурсов, финансовая эффективность, скорость выполнения проектов. К недостаткам таких систем относятся: психологические сложности работы с персоналом (время на адаптацию людей друг с другом, угроза потерять работу, страх перед изменениями); увеличенные риски неправильных решений.

3.2. Модель организационной структуры в системе ADOit

Модель организационной структуры создается в директории Models.

Первоначально необходимо создать систему папок для моделей архитектуры предприятия. Система папок создается по технологии, описанной в предыдущих практикумах. Пример созданной такой системы папок показан на рис. 3.1. На нижнем уровне создана папка с именем «Оргструктура», в которой будет храниться модель организационной структуры предприятия.

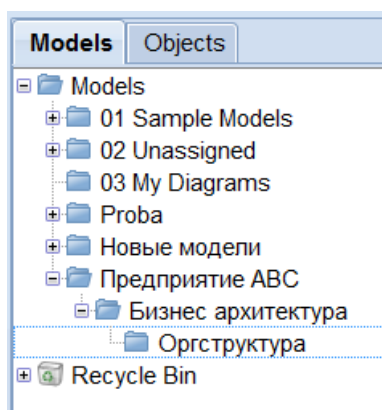


Рисунок 3.1 - Место новой папки для хранения модели оргструктуры

Упражнение 1. Создание системы папок для организационной структуры. Необходимо создать папку для бизнес-архитектуры своего предприятия. В этой папке создать подпапку для моделей организационной структуры. Имена папок должны иметь оригинальные названия.

При нажатии правой кнопкой мыши на созданной папке и выборе пункта New откроется меню, показанное на рисунке 3.2.

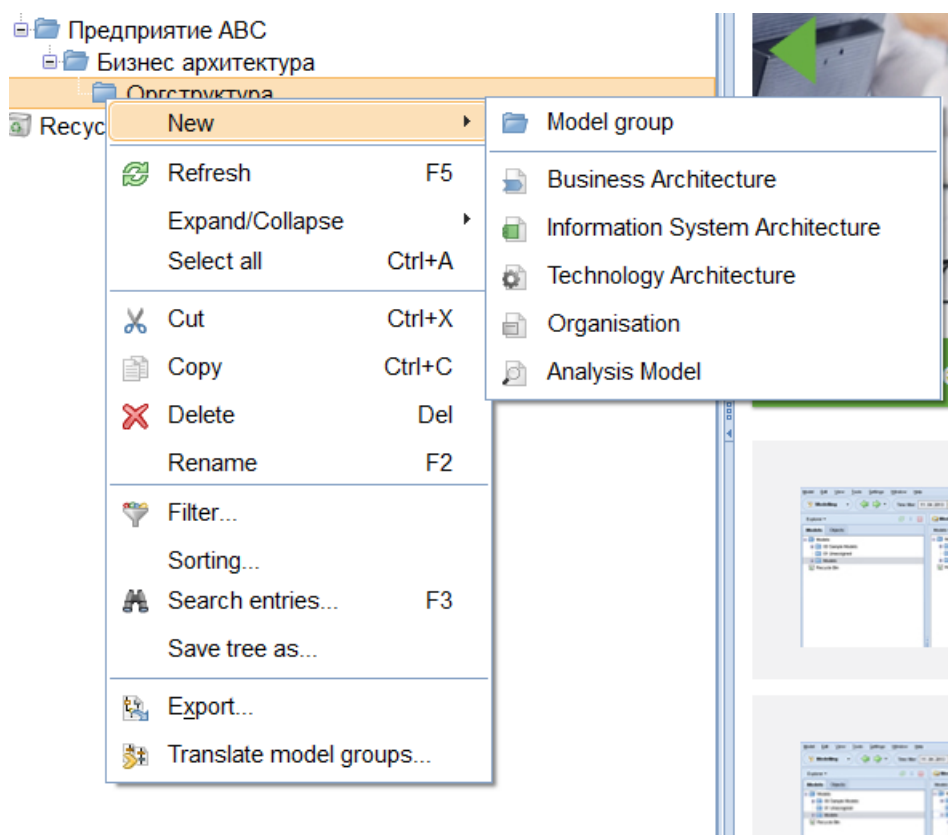


Рисунок 3.2 - Выбор типа модели

Для создания модели организационной структуры необходимо выбрать тип модели "Organization". При нажатии мышкой на этот пункт меню создается в папке новый файл с рабочим условным именем, как показано на рис. 3.3.

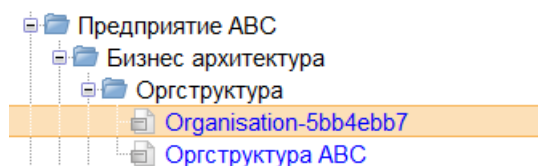


Рисунок 3.3 - Создан прототип файла для новой модели

Рабочее имя созданного файла надо заменить на актуальное имя, то которое соответствует создаваемой архитектуры предприятия.

При нажатии правой кнопкой мыши на имени файла открывается окно, показанное на рис. 3.4.

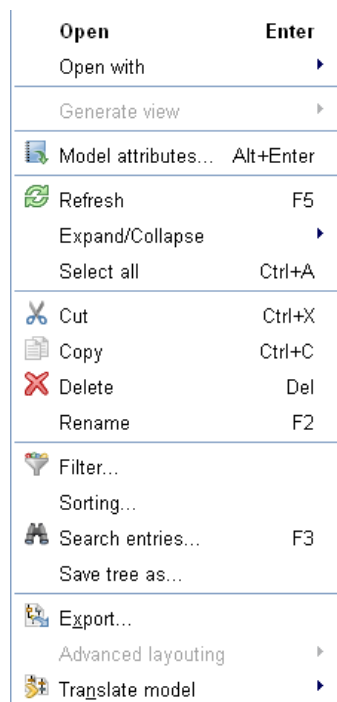


Рисунок 3.4 – Меню действий с файлом модели

Пункт меню “Open with” предназначен для выбора редактора представления модели. Как показано на рис. 3.5 существует два типа редактора: графический редактор (Graphical editor) и табличный редактор (Tabular editor).

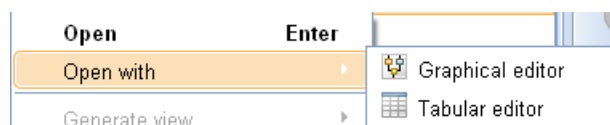


Рисунок 3.5 – Выбор редактора модели

При выборе табличного редактора информация о модели будет представлена в виде, показанном на рис. 3.6.

Objects		
Organisation Unit		[0]
Actor		[0]
Block		[0]
Note		[0]
Relations		
Has Note		[0]
Subordinated Organisation Units		[0]
Related Organisation Units		[0]
Connector		[0]

Рисунок 3.6 – Информация для табличного редактора

При выборе пункта меню “Model attributes” (Атрибуты модели) открывается окно, показанное на рис. 3.7.

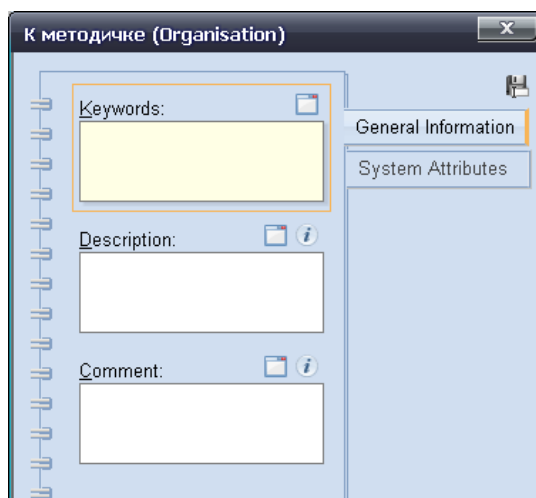


Рисунок 3.7 – Окно с общими атрибутами модели

Окно оформлено в виде традиционного «блокнота» и имеет две вкладки. Первая вкладка содержит общее описание модели (General Information). Она содержит три поля для описания модели. Первое поле “Keywords” предназначено для ключевых слов, второе поле “Description” для тестового описания модели и третье поле “Comment”, как и ранее для дополнительного комментария.

Вторая вкладка “System Attributes” содержит некоторые системные атрибуты модели. Содержимое вкладки показано на рис. 3.8.

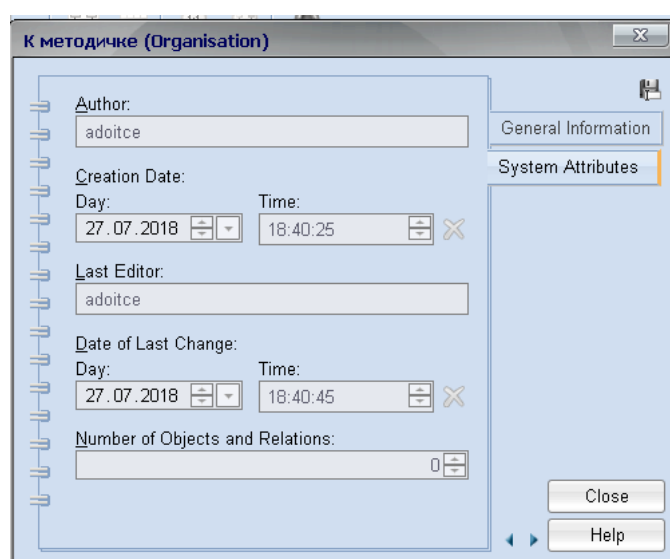


Рисунок 3.8 – Вкладка с системными атрибутами модели

Вкладка содержит следующую автоматически формируемую информацию: идентификатор автора модели (Author), идентификатор редактора последней модификации (Last Editor), дату создания модели (Creation Date), дату последних изменений (Date of Last Change) и число объектов и связей в модели (Number of Objects and Relations).

Пункт меню “Translate model”, как и было описано ранее в практикуме 1, дает возможность перевести термины модели на другой язык. При этом есть возможность перевести как термины всей модели, так и только ее атрибуты, путем выбора соответствующего пункта контекстного меню, как показано на рис. 3.9.



Рисунок 3.9 – Выбор переводимых терминов

Конечно, язык перевода должен поддерживаться системой. Система дает предупреждение об этом, как показано на рис. 3.10.

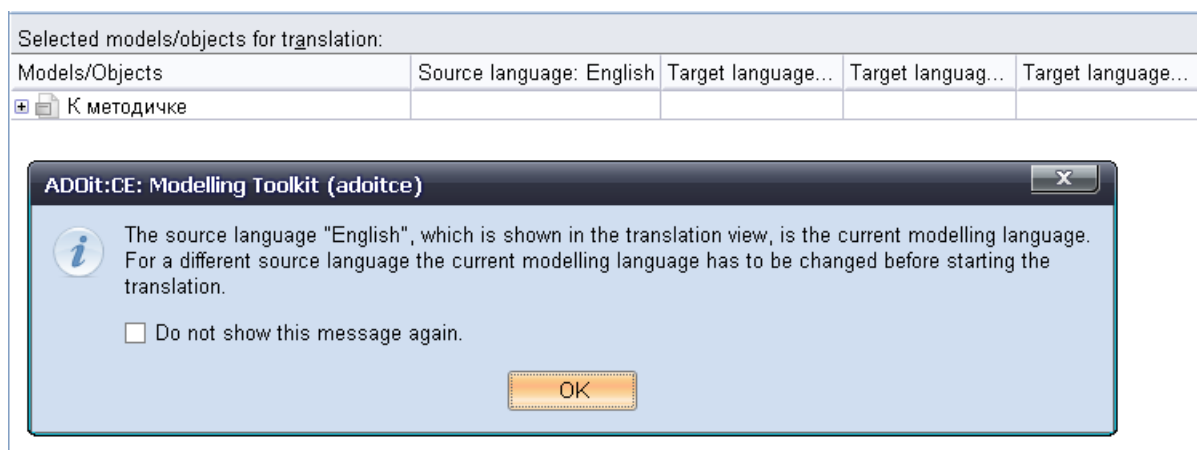


Рисунок 3.10 – Предупреждение об используемом языке

Следует обратить внимание на красную звездочку возле имени файла, как показано на рис. 3.11. Она сигнализирует о том, что файл еще не сохранен.

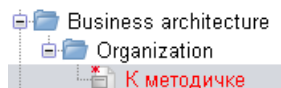


Рисунок 3.11 – Отметка о несохраненном файле

Одновременно с созданием файла открывается рабочее поле графического редактора для моделирования. В левой стороне рабочего поля

представлена палитра инструментов для создания модели, как показано на рис. 3.12.



Рисунок 3.12 - Палитра инструментов для создания модели оргструктуры

Рассмотрим объекты и связи организационной модели предприятия.

Объект "Бизнес-единица". Графический элемент для обозначения этого объекта показан на рис. 3.13.

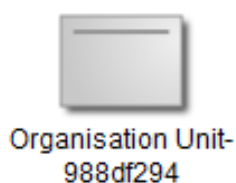


Рисунок 3.13 - Графическое обозначение бизнес-единицы

Объект переносится на рабочее поле путем перетаскивания с помощью мыши. При этом автоматически создается рабочий идентификатор (условное имя) бизнес-единицы.

При нажатии правой кнопки мыши на объекте открывается окно, показанное на рисунке 3.14. Это окно является стандартным для возможных действий с объектами графических моделей.

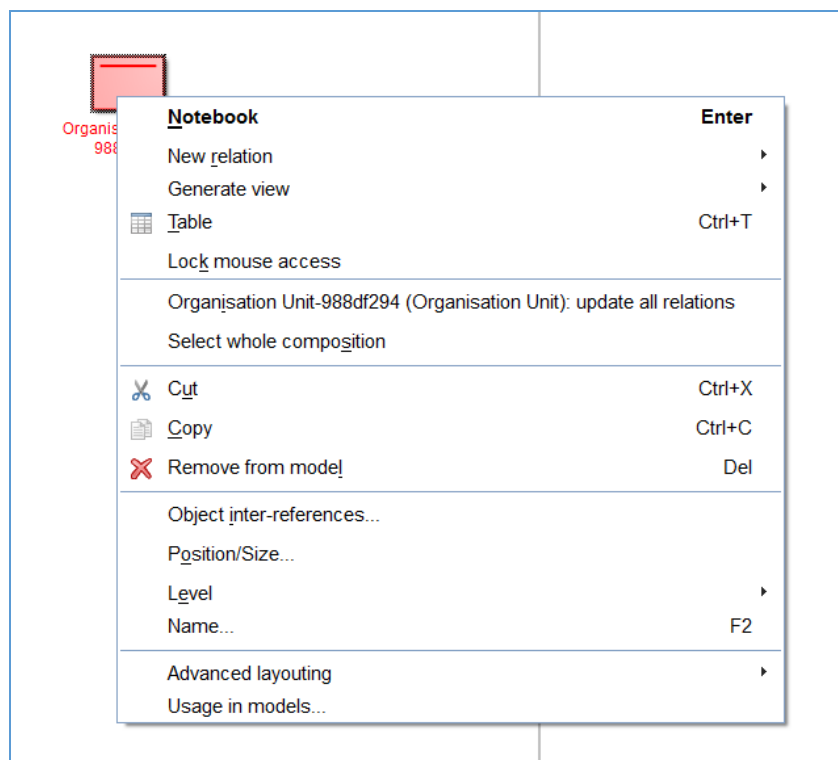


Рисунок 3.14 - Меню действий с объектом

Пункт меню “New relation” позволяет создавать новые связи объекта без перетаскивания изображения связи с помощью мыши из палитры. На рис. 3.15 показан открывающийся список возможных связей объекта типа бизнес-единицы. “outgoing” – исходящие связи. “incoming” – входящие связи. Описание связей будет представлено ниже.

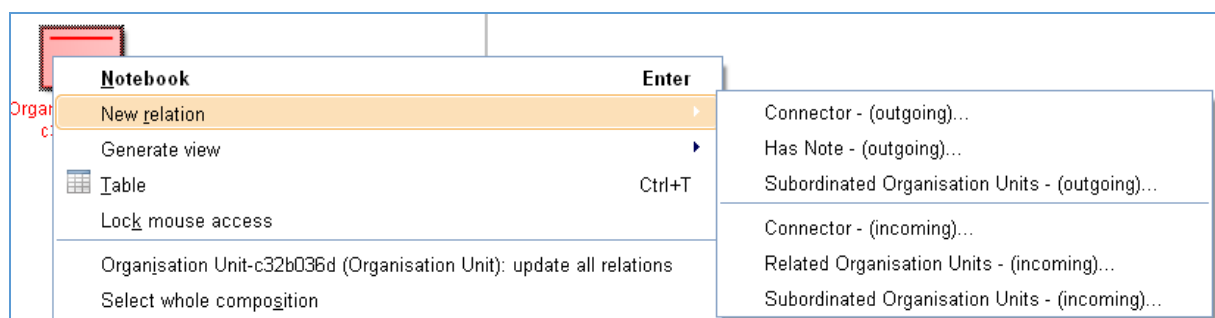


Рисунок 3.15 – Список возможных связей бизнес-единицы

На рис. 3.16 показан открывающийся список формирования взглядов (Generate view) представления моделей.

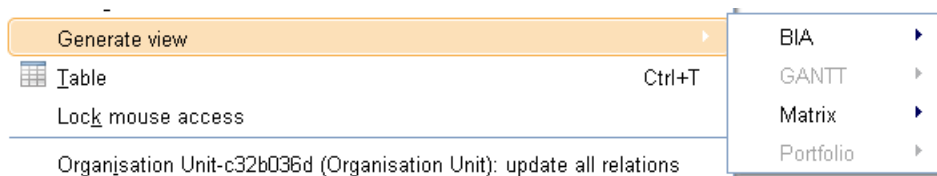


Рисунок 3.16 – Формирование видов модели

Для модели организационной структуры можно сформировать два взгляда: BIA - Business Impact Analysis – бизнес-анализ влияний и “Matrix” – формирование матричной модели. На рис. 3.17 показан список возможных взглядов для организационной структуры.

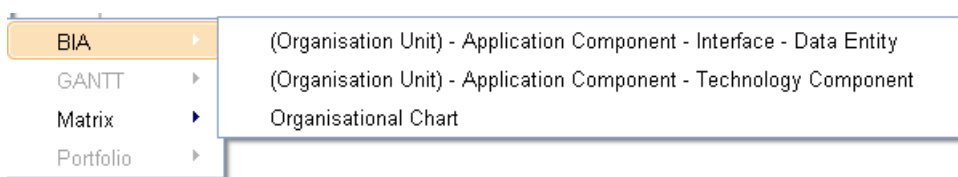


Рисунок 3.17 – Список возможных взглядов для организационной структуры

Если открыть пункт меню “Organization Chart” (Организационная схема), то откроется новая модель, показанная на рис. 3.18.

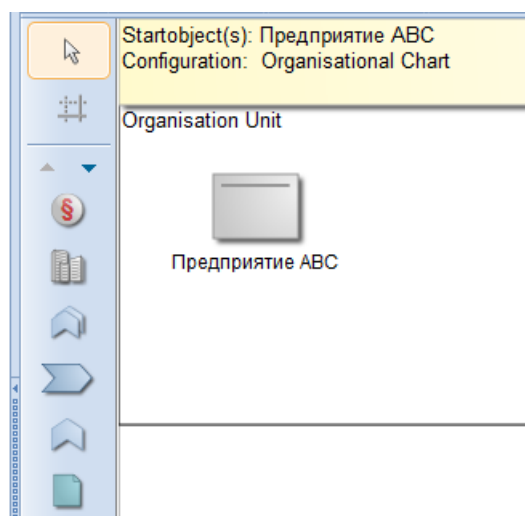


Рисунок 3.18 – Организационная схема в режиме BIA

В этом случае в палитре объектов и связей появляются все возможные объекты, в том числе объект «Принцип», обозначаемый кружком со знаком параграфа внутри. Для полученной модели можно строить все возможные графические схемы архитектуры. Так на рис. 3.19 показана возможная диаграмма связей бизнес-единицы и принципов, построенная в режиме BIA.



Рисунок 3.19 – Схема связей бизнес-единицы и принципов

Для организационной структуры в данной версии системы можно построить только один матричный взгляд, как показано на рис. 3.20. К этому пункту можно будет вернуться после рассмотрения архитектуры бизнес-процессов-предприятия.

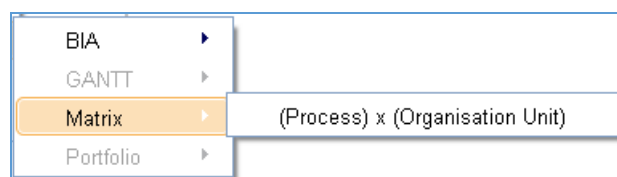


Рисунок 3.20 – Пункт выбора построения матричного взгляда

Подробная работа со взглядами будет рассматриваться позднее.

При выборе пункта меню “Incoming Object References (Входящие ссылки на объект)” открывается окно, показанное на рис. 3.21.

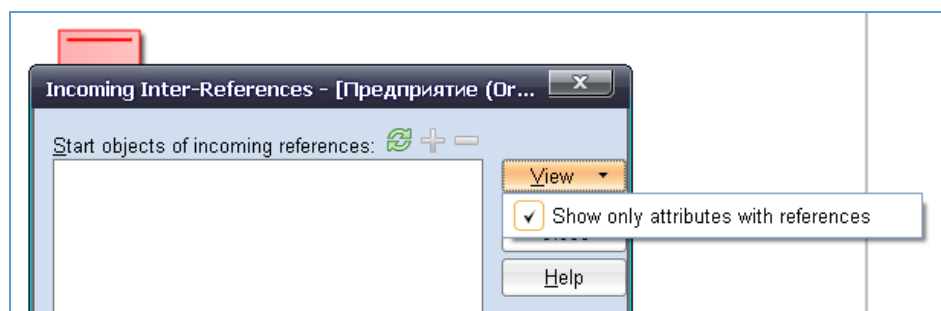


Рисунок 3.21 – Окно для имеющихся ссылок на объект

Пока ссылок на объект нет.

Пункт меню “Advanced layouting” (Расширенные связи), показанный на рис. 3.22, дает возможность построить кластерную карту (Cluster Map), которая будет рассматриваться позднее.

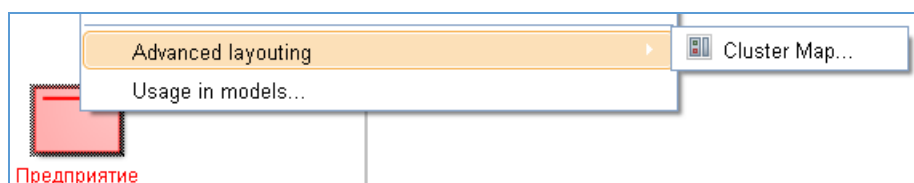


Рисунок 3.22 – Формирование кластерной карты

Последний пункт меню “Usage in Models” открывает окно со списком всех артефактов хранилища, связанных с данным объектом, как показано на рис. 3.23. Единственный артефакт здесь – папка с именем «К методичке».

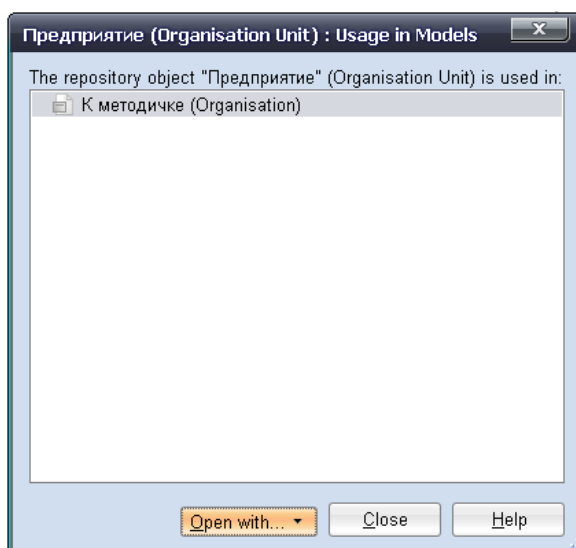


Рисунок 3.23 – Список артефактов, связанных с объектом

Более подробно пункты меню будут объясняться в дальнейшем по мере необходимости.

Для задания нового имени бизнес-единицы можно воспользоваться пунктом "Name...". В этом случае открывается окно, показанное на рисунке 3.24.

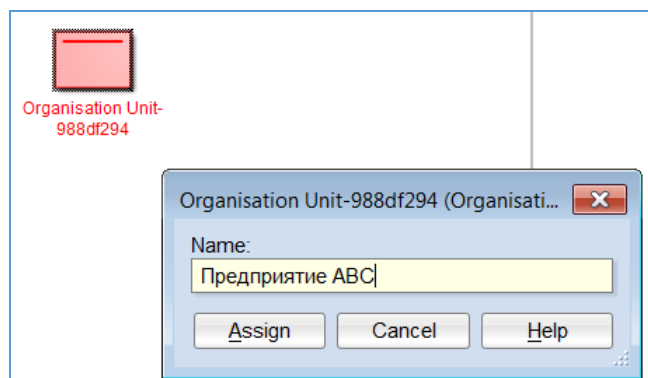


Рисунок 3.24 - Окно для замены имени объекта

Как обычно в системах фирмы ВОС вся информация об объекте хранится в блокноте. Блокнот открывается либо выбором пункта "Notebook" в меню, показанном на рисунке, либо двойным щелчком мыши на объекте.

Пример блокнота для бизнес-единицы показан на рис. 3.25.

Рисунок 3.25 - Блокнот для бизнес-единицы

Кроме полей, описанных в предыдущем практикуме, главная вкладка "General Information" имеет два следующих специфических поля: "Subordinated Organization Units" и "Higher Level Units". Первое поле задает список

подчиненных бизнес-единиц. Второе поле задает список бизнес-единиц более высокого уровня, связанных с данной единицей.

Упражнение 2. В данном упражнении нужно создать главную бизнес-единицу для предприятия. Переименовать рабочее имя. Например, можно назвать объект "Предприятие ABC".

Связь между бизнес-единицами. Связь между бизнес единицами предусмотрена только по подчиненности (Subordinated Organization Units). Графически это изображается линией с пустым ромбиком с одного конца. Связь создается, как это организовано во всех программных продуктах фирмы ВОС, путем переноса соответствующего объекта с палитры на рабочее поле с помощью мыши и соединением пиктограмм на поле. Ромбик на связи направлен к старшей бизнес-единице. Проводить связь мышкой надо от старшей единицы к младшей единице. Также, как это было описано выше, можно воспользоваться меню объекта. Иногда это более удобно, при этом можно задавать связь в обратном направлении от подчиненной бизнес-единицы к главной.

Для связи существует также простейший блокнот, показанный на рис. 3.26.

Subordinated Organisation Units (Предприятие ABC -> Продажи)

Description:

Action Required

☐ Action Required

Reason:

General Information

Lifecycle

Close

Help

Рисунок 3.26 - Блокнот для связи

Описания полей были приведены в практикуме «Архитектурные принципы».

Упражнение 3. В данном упражнении надо создать модель организационной структуры, показанной на рис. 3.27.

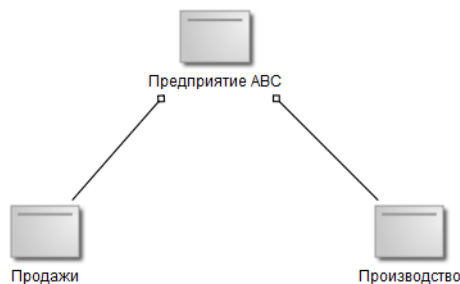


Рисунок 3.27 - Пример простейшей организационной структуры

Графические изображения линий связей между объектами можно с помощью мыши представлять ломанными, как показано на рис. 3.28.

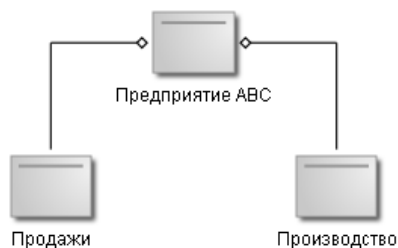


Рисунок 3.28 – Новый вариант модели

При создании графической модели в блокноты связанных бизнес-единиц автоматически будет заносится информация о связях, как показано на рис. 3.29 для бизнес-единицы "Продажи".

Name:

ID:

Description:

Subordinated Organisation Units:

Higher Level Organisation Units:

Рисунок 3.29 - Информация о связи в блокноте

Можно добавить связь непосредственно в блокнот. Для этого воспользоваться кнопкой с крестиком и выбрать соответствующий элемент. Связи в блокноте можно установить с любыми бизнес-единицами, имеющимися в хранилище артефактов архитектуры. При этом графический элемент не создается. С другой стороны, при удалении графического объекта типа связи запись о связи в блокноте не удаляется (Здесь недоработка фирмы или недостаток бесплатной версии). Для удаления графического изображения связи из модели необходимо нажать правую кнопку мыши на этой связи. Откроется окно, показанное на рис. 3.30.

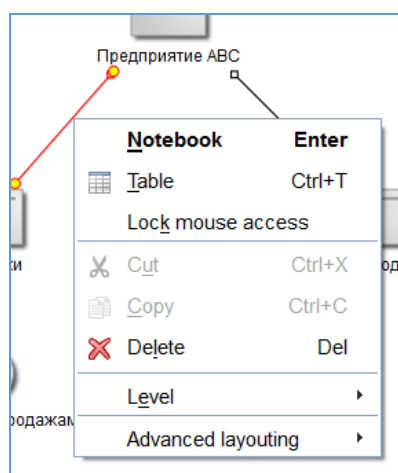


Рисунок 3.30 - Действия с объектом типа связи

Пункт меню "Delete" удаляет связь.

Одновременно с созданием модели организационной структуры в хранилище данных создается запись с информацией о созданных объектах, как показано на рис. 3.31.

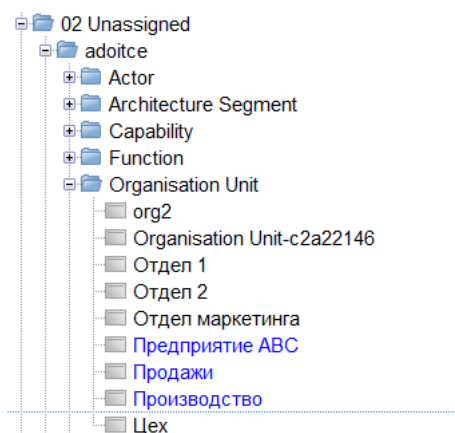


Рисунок 3.31 - Записи о новых объектах организационной структуры в хранилище данных

Система создает записи в своих рабочих папках. Затем их можно перенести в другие папки, которые необходимо создать самостоятельно. Например, как показано на рис. 3.32.

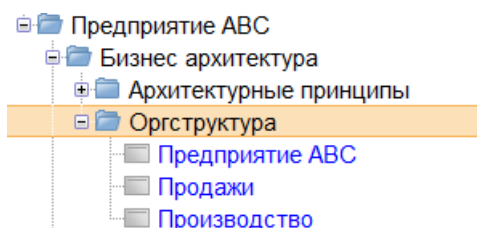


Рисунок 3.32 - Объекты организационной структуры в новой директории

Состав информации на вкладке "Organization" блокнота бизнес-единицы показан на рис. 3.33.

Рисунок 3.33 - Вкладка "Organization"

Поле "Is Responsible for:" содержит список всех артефактов архитектуры, за которые отвечает данная бизнес-единица.

Поле "Is a Customer and Uses:" содержит список объектов архитектуры, для которых данная бизнес-единица является клиентом или использует

в своей деятельности данные средства, такие как компьютерные приложения или программы.

Поле "Assigned Actors:" содержит список персоналий, которые имеют отношение к данной бизнес-единице.

Поле "Additional Information on Organization:" содержит дополнительную текстовую информацию по данной бизнес-единице.

Вкладка "Processes" будет рассмотрена позже в практикуме по архитектуре бизнес-процессов.

Вкладка "Principles" имеет вид, показанный на рис. 3.34.

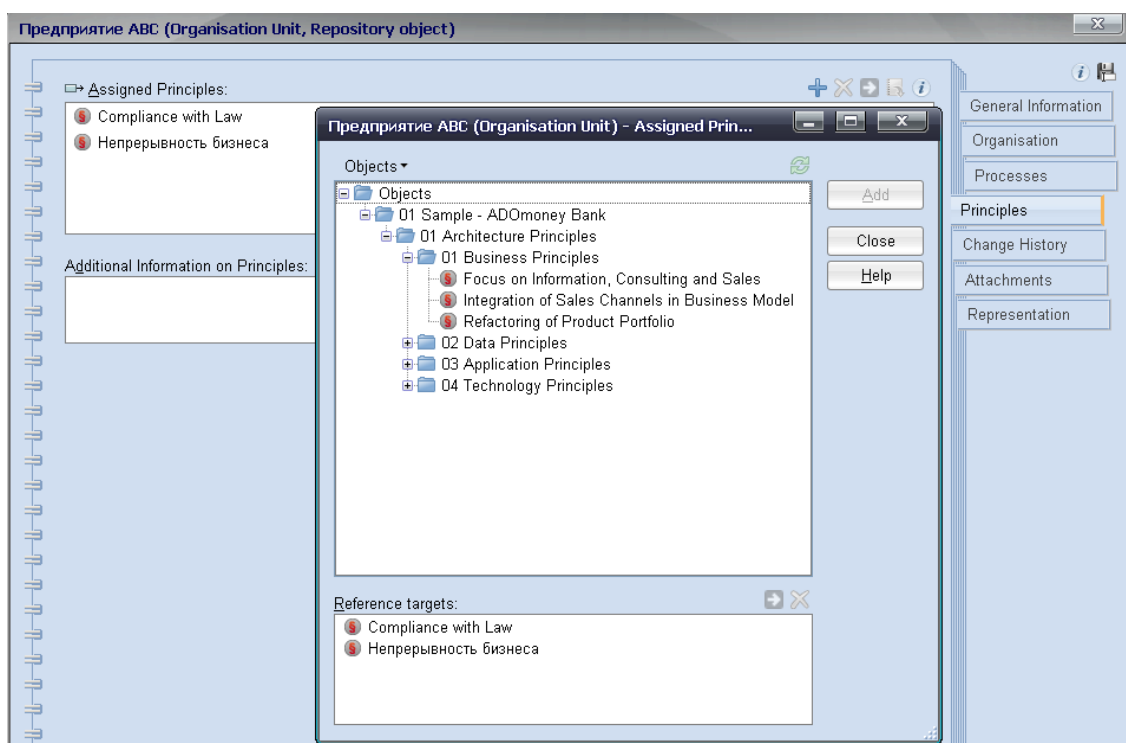


Рисунок 3.34 - Вкладка "Principles"

На этой вкладке задается список описанных ранее принципов деятельности предприятия, которые имеют отношение к данной бизнес-единице (Assigned Principles). Также можно задать дополнительные пояснения к принципам (Additional Information on Principles). Кнопка с крестиком открывает окно с объектами хранилища артефактов, как показано на рис. 3.32. Выбранные принципы записываются в блокнот и одновременно отражаются в поле "Reference targets" (Существующие ссылки), чтоб не записывать связи повторно.

Упражнение 4. В простейшей организационной структуре, созданной в упражнении 3, для главной бизнес-единицы добавить связи с любыми двумя принципами бизнеса, имеющимися в хранилище.

Объект "Участник" (Actor). Данный объект изображается в виде иконки, показанной на рис. 3.35.



Рисунок 3.35 - Изображение объекта типа "Участник"

Объект служит для представления человеческих ресурсов в архитектуре.

Для участника также автоматически создается запись в рабочей папке в хранилище данных. Ее, как и для организационной единицы, рекомендуется перенести в папку для созданной оргструктуры, как показано на рис. 3.36.

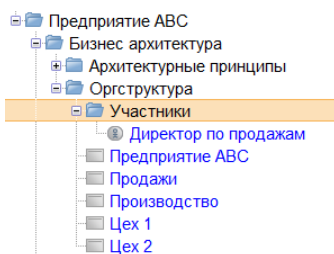


Рисунок 3.36 - Папка с информацией об участниках

Блокнот для участника имеет вид, как показано на рис. 3.37.

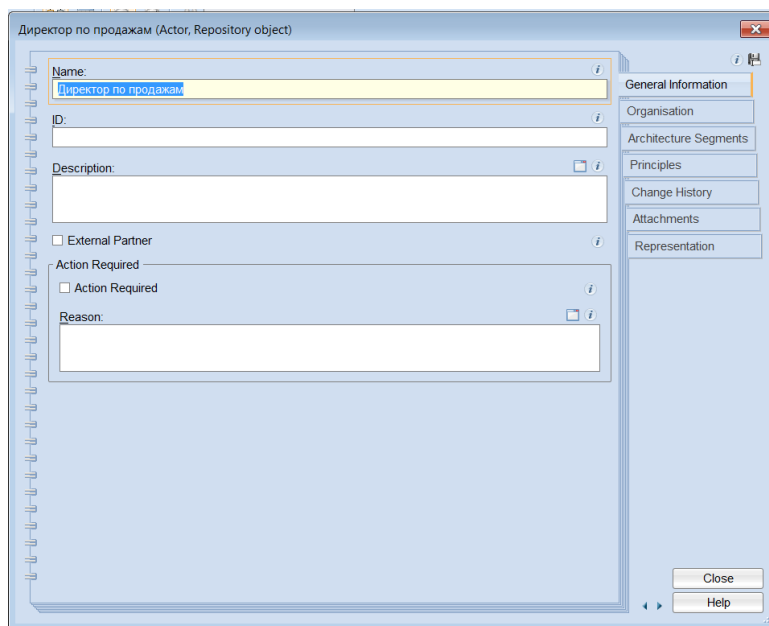


Рисунок 3.37 - Блокнот для участника

В главной вкладке появилось дополнительное поле “External Partner” (Внешний партнер) для отметки, что участник не является сотрудником организации.

На рис. 3.38 показано содержание вкладки "Organization" для блокнота участника.

Рисунок 38 - Вкладка "Organization" для участника

Поле "Related Organization Units:" (Связанные организационные единицы) в случае наличия графического объекта связи с бизнес-единицей заполняется автоматически. Это поле может быть задано также вручную из множества всех бизнес-единиц, имеющихся в хранилище артефактов.

Поле "Is Responsible for:" (Отвечает за) предназначено для списка объектов архитектуры, за описание которых (моделирование архитектуры) ответственна данная персона.

Поле "Is Responsible (Deputy) for:" (Отвечает за по должности) задает список реальных объектов, за которые несет ответственность данная персона.

Поле "Is Responsible (Development) for:" (Отвечает за развитие) задает список реальных объектов, за развитие которых несет ответственность данная персона.

Поле "Is Responsible (Operation) for:" (Отвечает за оперативное управление) задает список реальных объектов, за оперативное управление которыми несет ответственность данная персона. Как правило, это технические или программные системы.

Поле "Additional Information on Organization" (Дополнительная информация об организации) задает дополнительное описание особенностей работы участника.

Вкладка "Architecture Segments" будет рассматриваться в следующих практикумах. Вкладка "Principles" имеет такой же вид, как и в блокноте для бизнес-единицы на рис. 32. На этой связи указывается список принципов, которые относятся к деятельности данной персоны и дополнительная текстовая информация.

Связь участника с бизнес-единицей. Объект соединяется с организационной единицей с помощью связи "Related Organization Unit" (Относится к организационной единице). Связь изображается линией, на одном конце которой пустой ромбик, а на другом черный кружок. При этом линия рисуется от участника к бизнес-единице, как показано на рис. 3.39. Эту связь можно задать только с помощью мыши.

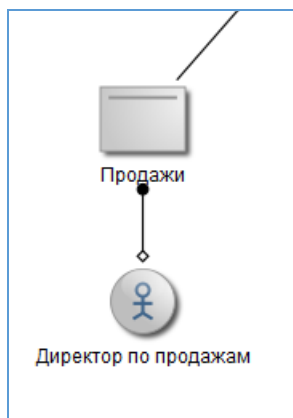


Рисунок 3.39 - Связь участника с организационной единицей

Упражнение 5. В простейшую организационную структуру, созданную в упражнении 3 добавить двух участников. Задать им имена. Связать одного с бизнес-единицей «Продажи», другого с бизнес-единицей «Производство». Для каждого из участников задать принципы, которыми они должны руководствоваться и объекты, за которые они отвечают.

Для более информативного представления модели в палитру добавлено два дополнительного объекта.

Объект “Block” (Блок). В палитре имеется универсальный объект «Блок», который может также использоваться для представления графической модели организационной структуры. Пример использования блока показан на рис. 3.40.

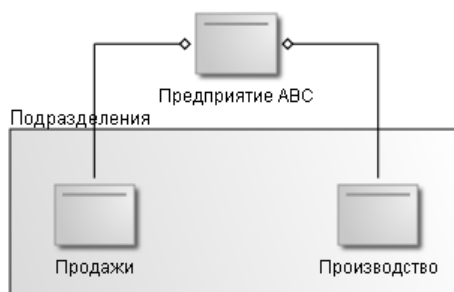


Рисунок 3.40 – Пример использования объекта типа «Блок»

К сожалению, в рассматриваемой версии системы блок не является артефактом архитектуры и не имеет своего файла в хранилище. С блоком создается блокнот, который содержит обычную общую информацию во вкладке “General Information”, показанную на рис. 3.41.

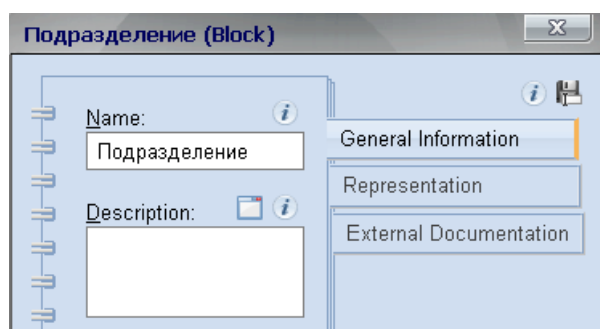


Рисунок 3.41 – Общее описание блока

Для использования блока для графических целей введены дополнительные возможности, задаваемые во вкладке “Representation” (Представление). Содержание вкладки показано на рис. 3.42.

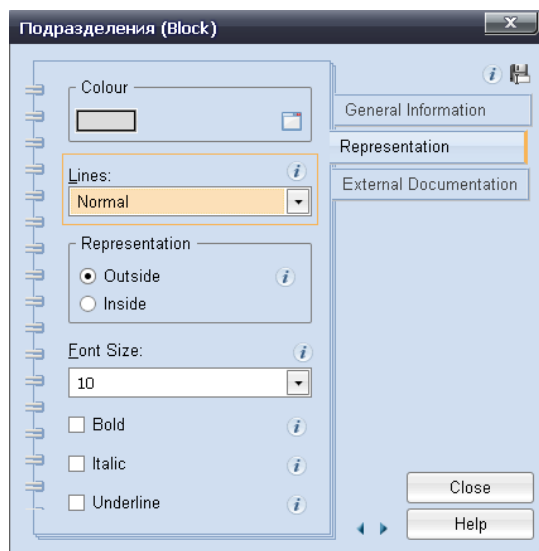


Рисунок 3.42 – Редактирование изображения блока

Имеется возможность задать вид контура блока (Lines); расположение названия блока: “Outside” снаружи блока, “Inside” – внутри блока; Размер шрифта - “Font Size” и тип шрифта. Важной характеристикой этого объекта является, что размеры блока можно изменять на схеме с помощью стандартных средств.

Объект “Note” (Заметка). Данный объект позволяет размещать на графическом представлении модели различные пояснения. На рис. 3.43 показан пример использования объекта типа «Заметка».

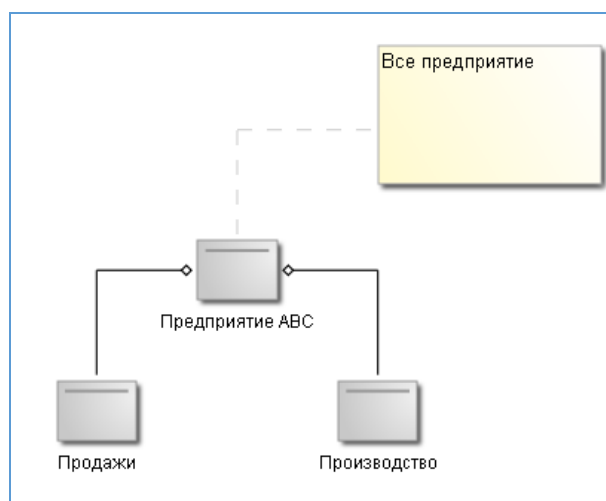


Рисунок 3.43 – Пример использования заметки

Связь “Has note” (Имеет заметку). Связь изображается в виде пунктирной линии и используется для присоединения объекта типа «Заметка» с другими объектами модели, как показано на рис. 3.43.

Объект типа «Заметка» имеет свой блокнот, показанный на рис. 3.44.

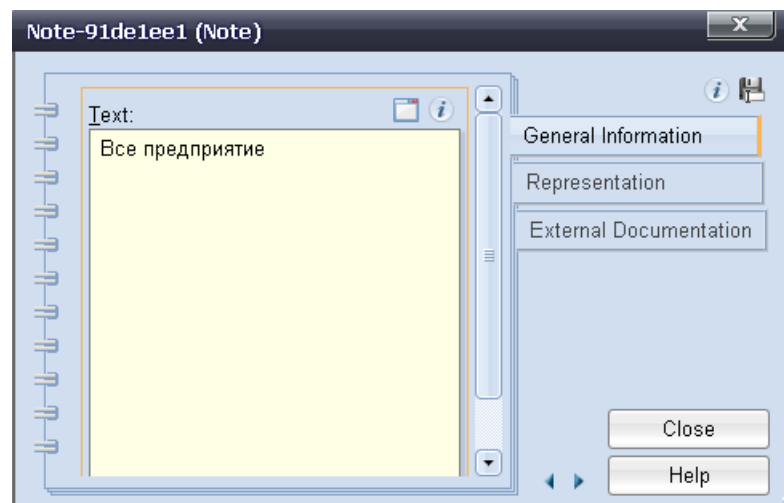


Рисунок 3.44 – Блокнот для заметок

Текст заметки вводится в блокноте. Для представления текста заметки имеется возможность простейшего редактирования, как показано на рис. 3.45.

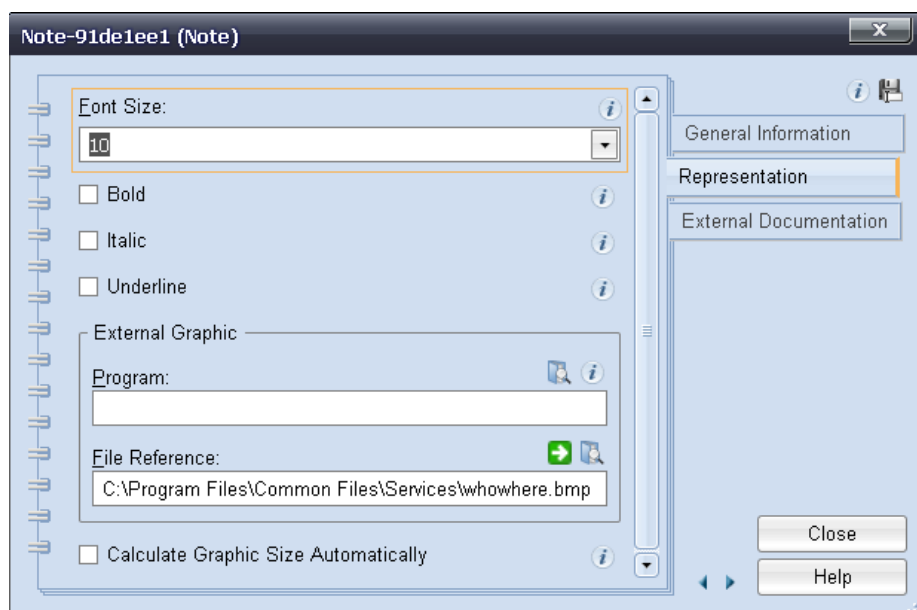


Рисунок 3.45 – Возможности редактирования заметки

В системе предусмотрена возможность включения в заметку графических объектов, подготовленных внешними графическими редакторами (External Graphic). Программу (Поле “Program”) можно не задавать, если используется стандартный рисунок типа bmp или jpg. Стрелочка позволяет просмотреть вставляемый рисунок. При этом текст не используется. На рис. 3.46 показан пример вставки графического изображения в качестве заметки.



Рисунок 3.46 – Заметка в виде графического внешнего объекта

Неозначенная простая связь (Connector). В палитре имеется простая связь, задаваемая в виде простой линии. Такая связь может быть задана, когда не существует подчиненности между бизнес-единицами. Например, на рис. 3.47 показана внешняя бизнес-единица «Клиенты». Для связи ее с подразделением «Продажи» использована простая связь.

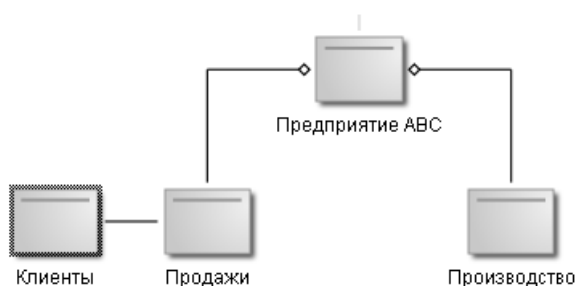


Рисунок 3.47 – Пример использования простой связи

Простая связь может использоваться для связи любых объектов модели. На рис. 3.48 показан пример использования связи бизнес-единицы и внешнего блока.



Рисунок 3.48 – Связь блока и бизнес-единицы

На рис. 3.49 показана семантическая схема изученных в данном практикуме объектов архитектуры.

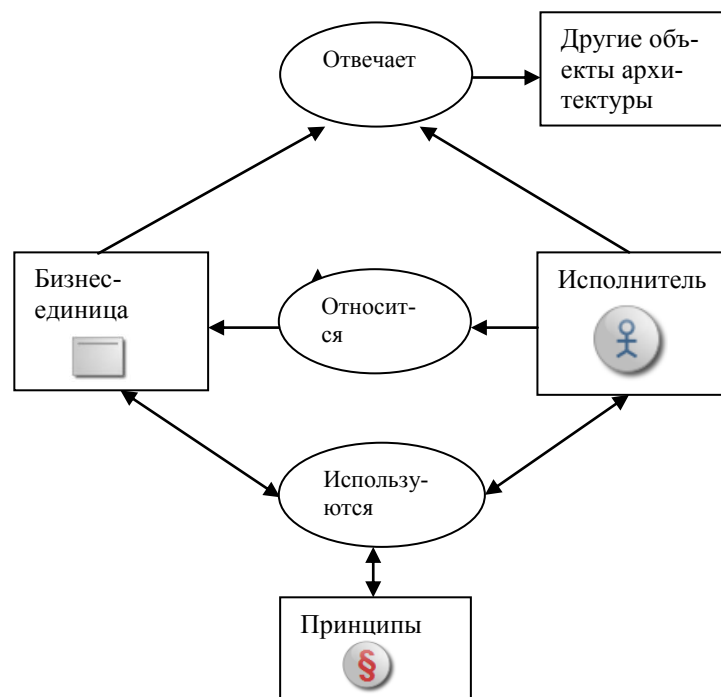


Рисунок 3.49 - Семантическая схема объектов, связанных с оргструктурой

3.3. Пример модели организационной структуры

В качестве примера построим матричную модель организационной структуры малого предприятия. Пусть условное предприятие выпускает два вида продуктов. Назовем их продукт А и продукт В. На предприятии созданы три службы: финансовая служба, служба маркетинга и производственный цех. Оба продукта выпускаются по разной технологии, имеют, как правило, различных заказчиков и рассчитаны на разные сегменты рынка. Для каждого вида продукта прописаны бизнес-процессы и назначены менеджеры бизнес-процессов. Для данной ситуации разумно построить матричную модель организационной структуры. Возможный вариант графической модели организационной структуры показан на рис. 3.50.

3.6. Варианты заданий

Вариант 1. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Подразделение развития предприятия, которому подчинены два отдела планирования и маркетинга, в каждом из которых работают по два клерка. Производственное подразделение, состоящее из двух цехов и склада. В цехах работают по два рабочих, на складе один кладовщик.

Вариант 2. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Отдел маркетинга, которому подчинены два подотдела: анализа рынка и стратегического маркетинга, в каждом из которых работают по три клерка. Непосредственно в отделе маркетинга также работают два менеджера, выполняющих различные функции. Плюс у предприятия есть два магазина.

Вариант 3. Построить модель оргструктуры турфирмы, в которой существуют следующие бизнес-единицы. Дирекция турфирмы, которому подчинены три отдела: маркетинговый, отдел по работе с клиентами и отдел по разработке новых туров. В каждом отделе работают по два работника. Непосредственно в дирекции работают секретарь, директор и бухгалтер. У фирмы есть еще два агента.

Вариант 4. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Коммерческий отдел, которому подчинены два подотдела: отдел закупок и административно-хозяйственная часть. В каждом отделе работают по три сотрудника. Непосредственно в коммерческом отделе работают коммерческий директор секретарь и бухгалтер. Кроме того имеется производство с тремя работниками.

Вариант 5. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Коммерческая служба, которой подчинены три отдела: отдел продаж, административно-хозяйственная часть и ремонтная мастерская. В каждом отделе работают по два работника. Непосредственно в коммерческой службе работают администратор и бухгалтер. Кроме того имеется склад с двумя работниками.

Вариант 6. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Отдел маркетинга, которому подчинены три подотдела: сбыта, закупок, рекламы. В каждом подразделении работают по два человека. Непосредственно в отделе маркетинга работают начальник отдела, секретарь и бухгалтер. Кроме того имеется два магазина.

Вариант 7. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Подразделение развития предприятия, которому подчинены три отдела: отдел планирования, отдел

рекламы и отдел маркетинга, в каждом из которых работают по два человека. В подразделении развития предприятия также работают три менеджера, выполняющих различные функции. Производственное подразделение состоит из двух цехов. В каждом цехе работает 3 человека. У предприятия также имеется филиал, в котором работают три человека.

Вариант 8. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Дирекция, которой подчинены три подразделения: коммерческая служба, исполнительная служба и хозяйственная часть, в каждом подразделении работают по три человека. Непосредственно в дирекции также работают директор, секретарь и бухгалтер. Кроме того есть производственный цех с тремя работниками.

Вариант 9. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Дирекция, которой подчинены четыре подразделения: отдел снабжения, отдел сбыта, отдел маркетинга и отдел кадров, в каждом из которых работают два человека. Непосредственно в дирекции также работают директор, секретарь и бухгалтер. Имеется еще магазин с тремя сотрудниками.

Вариант 10. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Отдел маркетинга, которому подчинены два подотдела: анализа рынка и стратегического маркетинга, в каждом из которых работают по три человека. Непосредственно в отделе маркетинга также работают два менеджера, выполняющих различные функции. Имеется еще цех с тремя работниками.

Вариант 11. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Отдел маркетинга, которому подчинены три подотдела: сбыта, закупок, рекламы. В каждом подотделе работают по два-три клерка. Непосредственно в отделе маркетинга работают начальник отдела, секретарь и бухгалтер. Имеется еще два магазина, в каждом по два работника.

Вариант 12. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Коммерческая служба, которой подчинены три отдела: отдел продаж, отдел закупок и ремонтная мастерская. В каждом отделе работают по два человека. Непосредственно в коммерческой службе работают администратор и бухгалтер. Имеется еще два филиала.

Вариант 13. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Первый – отдел маркетинга, которому подчинены три подотдела: сбыта, рекламы и планирования. В каждом отделе работают по три работника. Непосредственно в отделе маркетинга работают начальник отдела, секретарь и бухгалтер. Имеется еще цех.

Вариант 14. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Отдел маркетинга, которому подчинены три подотдела: сбыта, закупок, рекламы. В каждом подотделе работают по два человека. Непосредственно в отделе маркетинга работают начальник отдела, секретарь и бухгалтер. Имеются еще два представительства.

Вариант 15. Построить модель оргструктуры предприятия, в котором существуют следующие бизнес-единицы. Подразделение снабжения, которому подчинены три отдела: закупок, планирования и коммерческая служба. В каждом отделе работают по два человека. Непосредственно в подразделении снабжения работают руководитель подразделения, секретарь и маркетолог. Имеется также цех и склад.

3.7. Контрольные вопросы

- Виды организационных структур предприятия.
- Преимущества и недостатки линейно-функциональной модели.
- Преимущества и недостатки дивизионной модели.
- Преимущества и недостатки процессной модели.
- Что такое матричная модель оргструктуры предприятия?
- Преимущества и недостатки адаптивной модели.
- Что такое проектная оргструктура предприятия?
- Что такое бригадная оргструктура предприятия?
- Как связаны принципы бизнеса и оргструктура предприятия?
- Как обозначается на схеме бизнес-единица?
- Как обозначается на схеме персонал?
- Зачем нужен объект типа «Блок»?
- Какие существуют типы связей в модели оргструктуры?

3.8. Примеры принципов создания и управления организационной структурой предприятия

1. Близкие родственники не должны быть в прямом подчинении.
2. Сотрудник не может одновременно быть начальником и подчиненным. Например, командир полка не может быть одновременно командиром батальона или роты.
3. В иерархии не должно быть больше семи подчиненных подразделений.
4. Один менеджер не должен непосредственно управлять больше семи подчиненными. Иначе, надо вводить ответственных за направления деятельности.
5. Организационная структура предприятия должна соответствовать целям и задачам предприятия.
6. Разделение труда между функциональными подразделениями должно предусматривать пропорциональную нагрузку на работников.
7. Полномочия и ответственность работников должны соответствовать вертикальным и горизонтальным связям между подразделениями.
8. Должно поддерживаться оптимальное соотношение между функциями и обязанностями, полномочиями и ответственностью.
9. Организационная структура предприятия должна соответствовать и поддерживать социально-культурную среду предприятия.
10. Организационная структура управления должна соответствовать бизнес-процессам предприятия.
11. Организационная структура предприятия должна поддерживать принцип оптимального соответствия между управленческой структурой и производственной структурой.
12. Стоимость управленческой структуры должна быть оптимальной.
13. Организационная структура предприятия должна быть тесно связана с информационными системами и технологиями предприятия.

4. Лабораторный практикум «Бизнес-архитектура создания ценности»

4.1. Сегменты бизнес-архитектуры в системе ADOit

Бизнес-архитектура играет ведущую роль в построении архитектуры предприятия и непосредственно влияет на решения в области системной архитектуры, через которую оказывает влияние также на техническую архитектуру (последняя абсолютно вторична по отношению к бизнес-архитектуре). Проектирование бизнес-архитектуры необходимо начинать с построения классификатора областей, направлений и видов деятельности организации и с каталога производимых ею продуктов и услуг — с того, к чему позже будут привязаны бизнес-процессы, функции, а затем информационные системы и на основании чего будет определен их функционал.

Создание модели бизнес архитектуры предприятия начинается с определения сегментов бизнеса. Как правило, сегменты бизнеса описывают деятельность компании на уровне основных направлений бизнеса, программ или портфелей проектов и отражают главные области бизнеса (например, производство, маркетинг, логистика, финансы).

Эта категория архитектуры отвечают на вопрос **Что бизнес делает?** (WHAT).

Рассмотрим построение модели сегментов в системе ADOit.

Первоначально надо создать новую папку для модели, как показано на рис. 4.1.

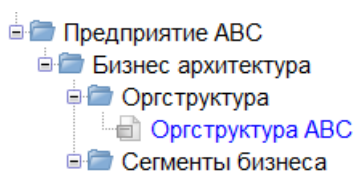


Рисунок 4.1 - Создание новой папки для модели

Для создания новой модели (New) надо выбрать пункт контекстного меню 'Business Architecture', как показано на рис. 4.2, и переименовать созданную модель.

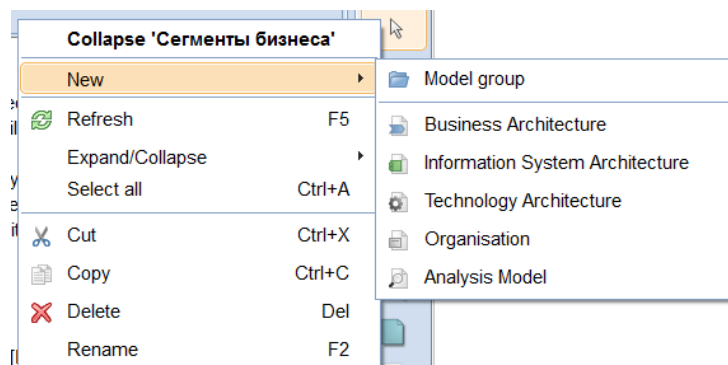


Рисунок 4.2 - Выбор типа модели

При активизации модели откроется стандартное рабочее поле с палитрой объектов и связей для архитектуры бизнеса. Объекты модели и связи между ними будем рассматривать последовательно.

Объект “Architecture Segment” (Архитектурный сегмент) может представляться в виде иконки, как показано на рис. 4.3, или в виде прямоугольного поля, как показано на рис. 4.4. Вид представления определяется в блокноте объекта на закладке “Representation” (Представление).



Рисунок 4.3 - Представление архитектурного сегмента в виде иконки

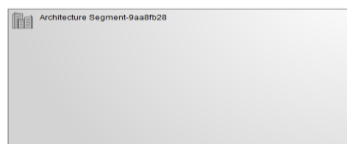


Рисунок 4.4 - Представление архитектурного сегмента в виде прямоугольного поля

Как обычно наименование сегмента бизнеса следует переименовать. Блокнот для архитектурного сегмента имеет вид, показанный на рис. 4.5.

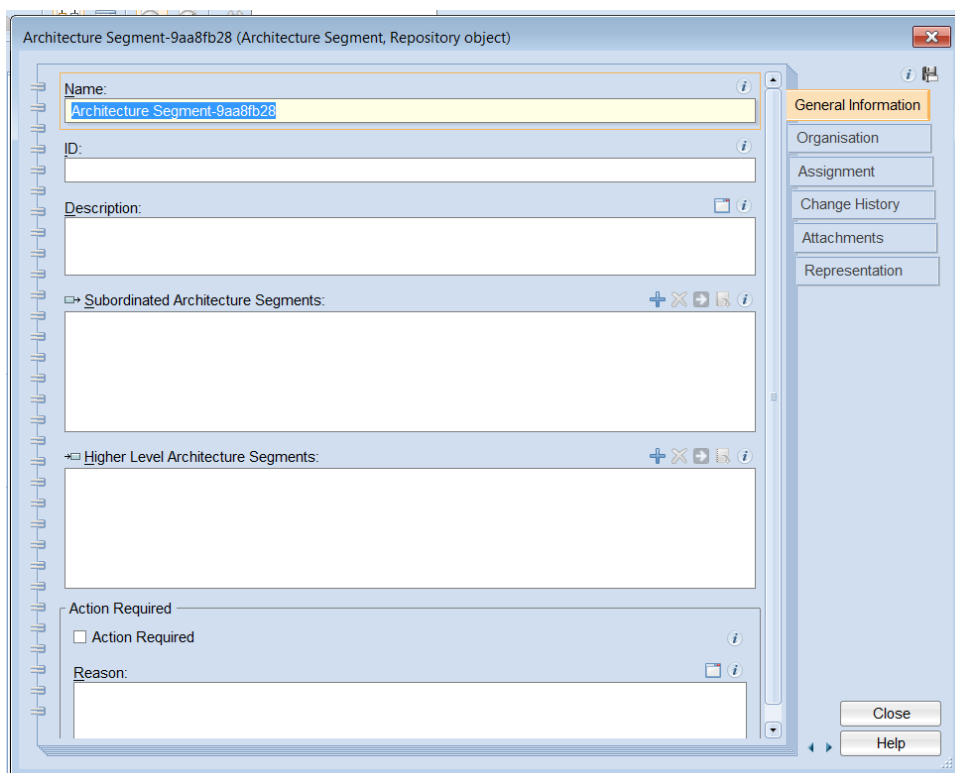


Рисунок 4.5 - Блокнот для сегмента бизнес-архитектуры

На главной вкладке кроме уже известных полей имеется два поля, отражающие иерархическую связь между сегментами. "Subordinated Architecture Segments" (Вложенные архитектурные сегменты) и "Higher Level Architecture Segments" (Сегменты более высокого уровня).

Упражнение 1. В качестве упражнения требуется создать два главных сегмента бизнеса университета. Любой университет имеет два основных направления деятельности: научно-исследовательская работа и обучение студентов. На рис. 6 показана простейшая модель сегментов бизнеса университета. Объекты переносятся на рабочее поле с помощью мыши и переименовываются.

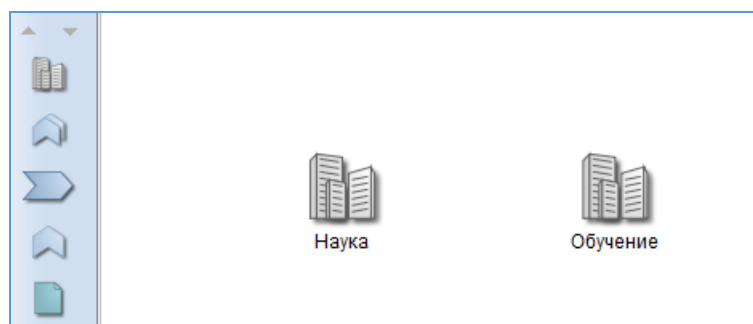


Рисунок 4.6 - Главные сегменты деятельности университета

Связь между вложенными сегментами задается с помощью штрих-пунктирной линии с незаполненным ромбиком на стороне главного сегмента бизнеса ("Subordinated Architecture Segments").

Упражнение 2. В качестве упражнения требуется создать еще один вложенный сегмент "Обучение менеджменту" и отразить на графической модели тот факт, что он входит в сегмент "Обучение". Вариант результирующей модели показан на рис. 4.7.



Рисунок 4.7 - Иерархическая модель сегментов

При этом автоматически в блокноты связанных объектов будет добавлена информация о связях, как показано на рис. 4.8 и 4.9.

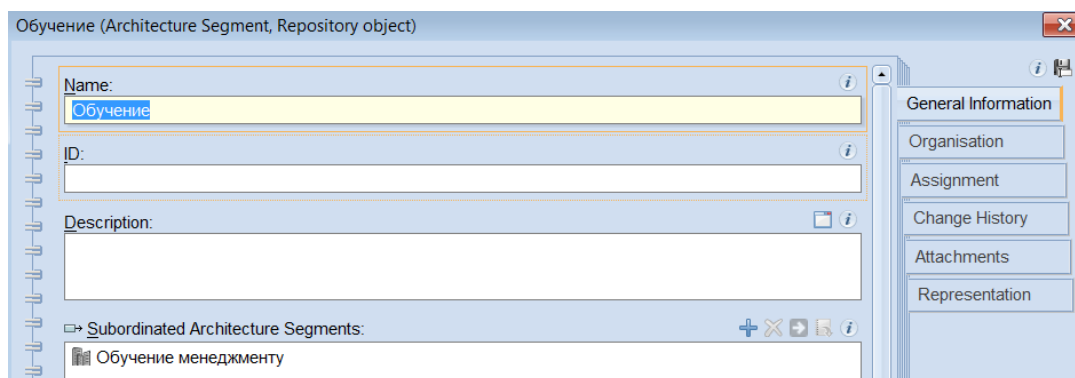


Рисунок 4.8 - Отражение связи в блокноте главного сегмента

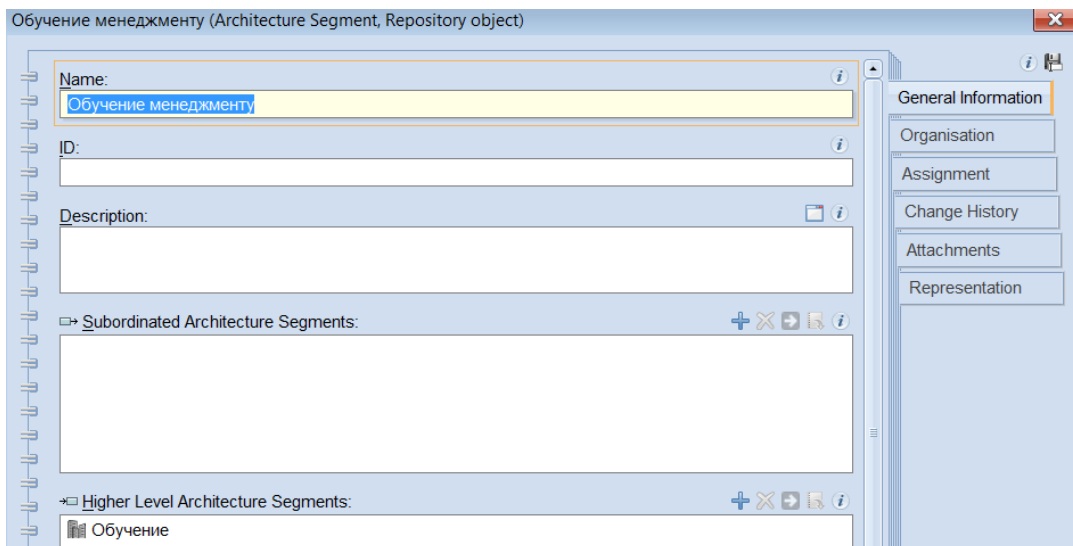


Рисунок 4.9 - Отражение связи в блокноте подчиненного сегмента

Упражнение 3. Требуется изменить форму представления для модели в предыдущем упражнении. Для этого надо представить все объекты в виде прямоугольников и расположить вложенный сегмент внутри главного. Для этого надо в блокнотах для объектов зайти во вкладку 'Representation' и убрать галочку в поле "Display as Icon" (Представить в виде иконки), как показано на рис. 4.10. Затем изменить размеры прямоугольников и переместить их на нужное место. При этом также можно изменить цвет фона прямоугольника в поле "Colour" (Цвет).

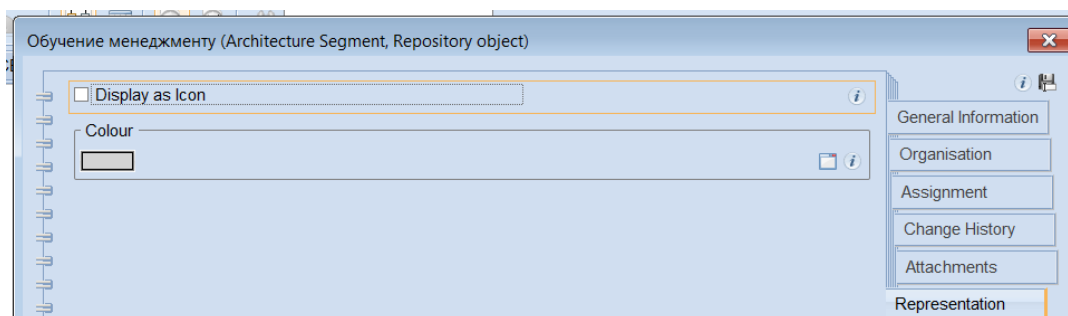


Рисунок 4.10 - Изменение формы представления сегмента

В результате должна получиться модель, показанная на рис. 4.11.

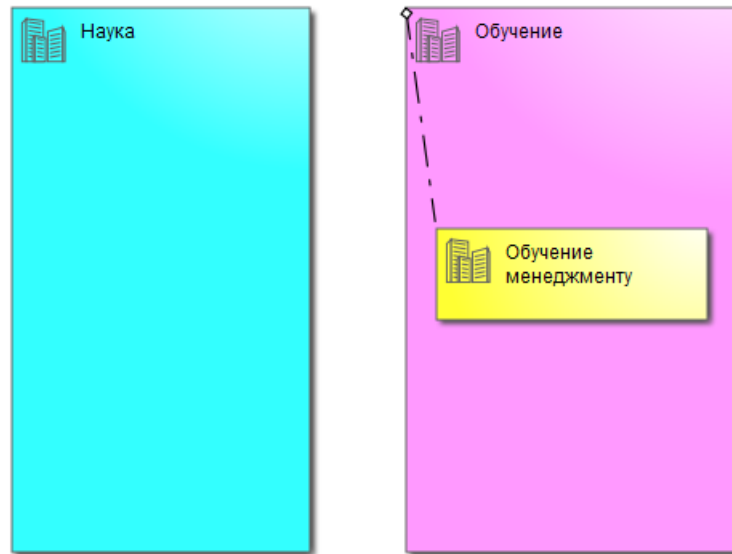


Рисунок 4.11 - Вторая форма представления модели

Рассмотрим связь сегментов бизнеса с организационной моделью. Рассмотрим простейшую модель предприятия, показанную на рис. 4.12. Условно все бизнес-единицы предприятия объединены в две основные, отвечающие за экономическую деятельность предприятия и за производственную деятельность предприятия.

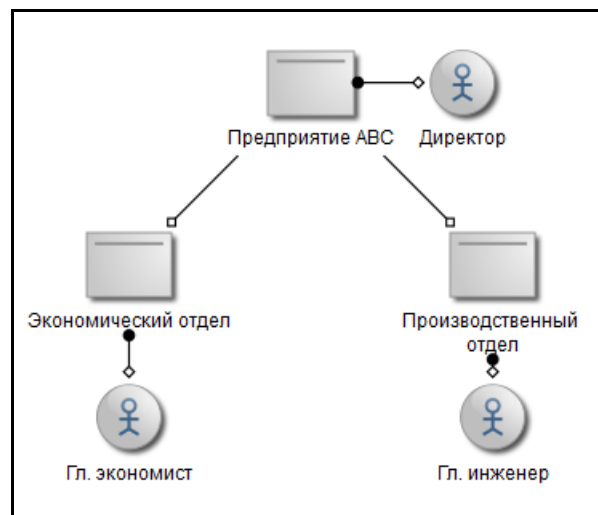


Рисунок 4.12 – Пример организационной схемы с двумя основными подразделениями предприятия

Вкладка блокнота "Organization", показанная на рис. 4.13, задает связь сегмента архитектуры с моделью организационной структуры и указывает на персоналии, которые отвечают за модель сегмента и за управле-

ние этим сегментом, как было описано в практикуме по моделированию организационной структуры.

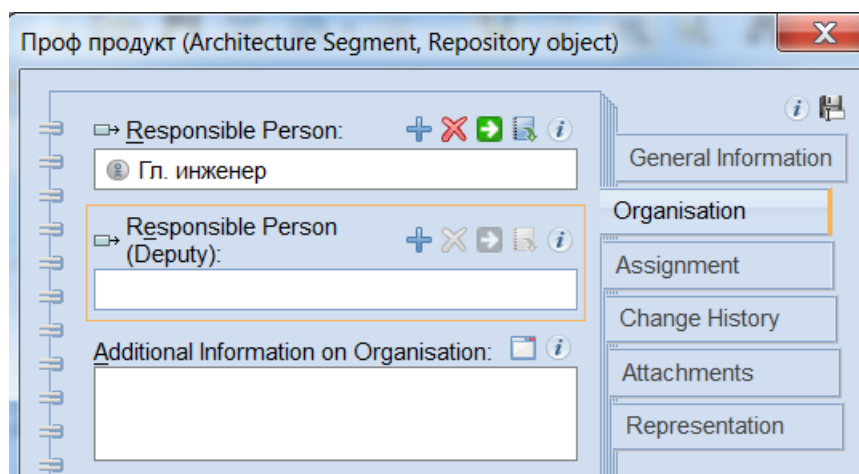


Рисунок 4.13 - Связь с ответственными персонами за сегмент

Здесь надо различать работника, ответственного за моделирование архитектуры (Responsible Person) и работника, ответственного за оперативное управление сегментом бизнеса (Responsible Person (Deputy)). При нажатии на кнопку с синим крестиком открывается хранилище данных, в котором надо выбрать нужного исполнителя, как показано на рис. 4.14.

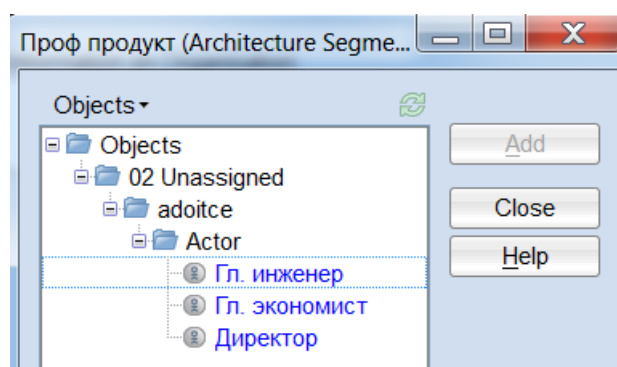


Рисунок 4.14 – Выбор ответственного исполнителя

Вкладка "Assignment" (Закрепление), показанная на рис. 4.15, задает связь с другими объектами бизнес-архитектуры. Так как еще в модели объектов для данного примера мало, то можно связать сегмент только с исполнителями и принципами. В дальнейшем по мере появления в модели новых объектов выбор связанных объектов будет расширяться.

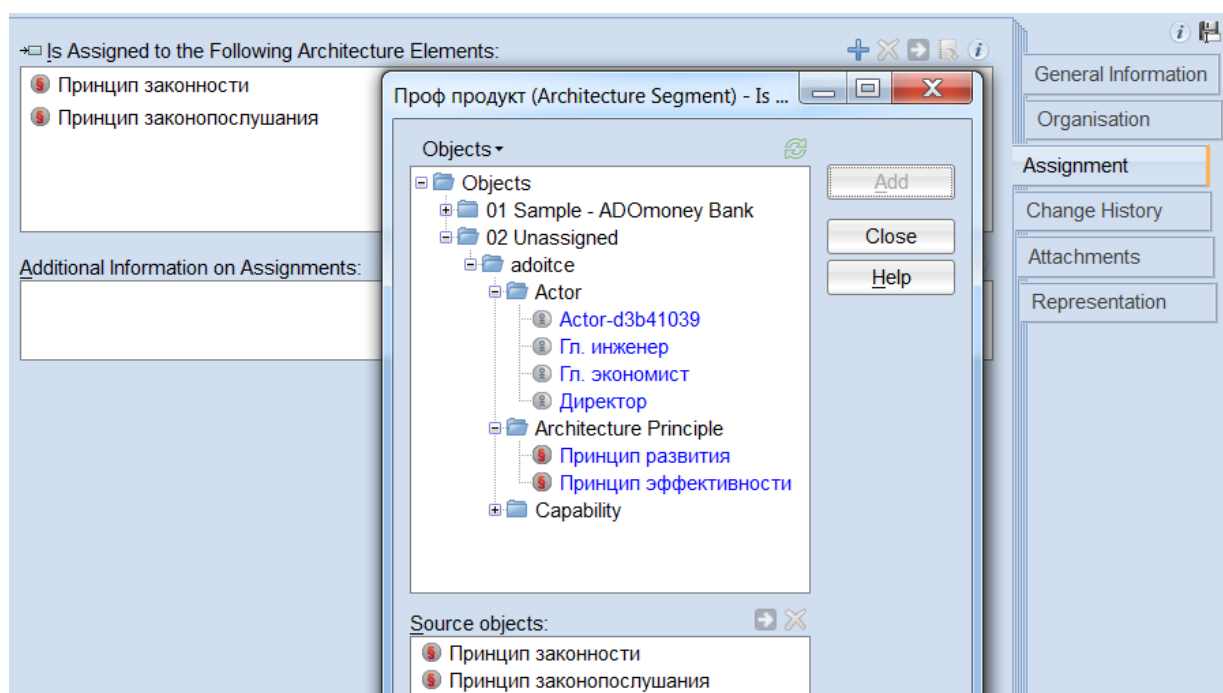


Рисунок 4.15 - Вкладка "Assignment" для указания объектов хранилища, которые относятся к данному сегменту бизнеса

4.2. Модель клиентов

Процесс построения любой бизнес-архитектуры начинается с вопроса о клиентах. В начале надо решить задачу сегментации рынка, то есть разделить рынок на группы покупателей, обладающих схожими характеристиками.

Добавим в последний пример организационной структуры два объекта «Ком.Клиенты» и «Бесп.Клиенты». Первый объект служит для включения в архитектуру бизнеса коммерческих клиентов, готовых платить за продукт с расширенными свойствами. Второй объект служит для включения в архитектуру сегмента перспективных клиентов, которые не готовы платить за продукт, но хотят познакомиться с его возможностями. Второй продукт поставляется бесплатно, но с некоторыми ограниченными функциями. Воспользуемся для обозначения клиентов на схеме объектом бизнес-единицы. Расширенная организационная схема, в которую добавлены объекты, задающие клиентов показана на рис. 4.16.

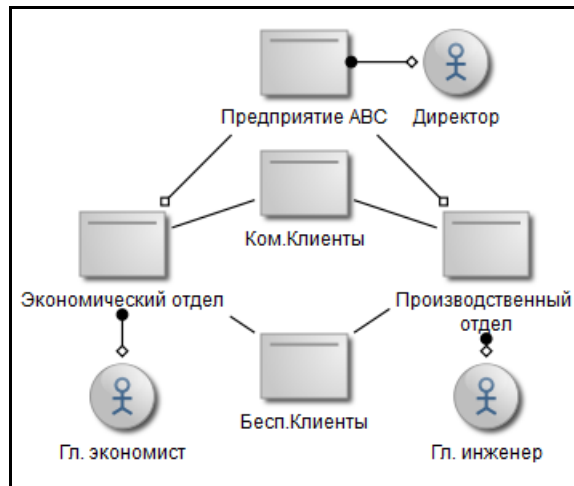


Рисунок 4.16 – Организационная схема с блоком клиентов

4.3. Модель продукта

Следующий блок бизнес-архитектуры связан с продуктом, с той ценностью, которую бизнес предлагает клиентам. К сожалению, в бесплатной версии ADOIT не введено специального обозначения для продукта. Будем обозначать продукт с помощью блока. На рис. 4.17 показана схема для описания двух типов продукта, для рассматриваемой модели предприятия.

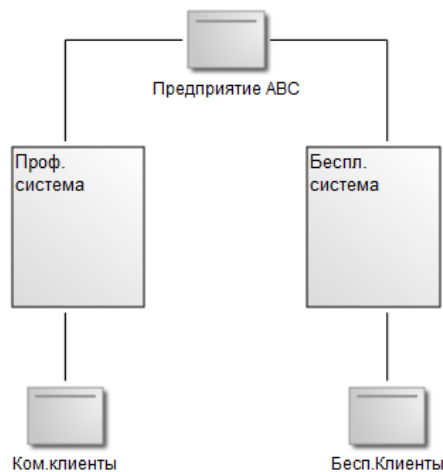


Рисунок 4.17 – Схема с двумя продуктами

На схеме показано, что каждый продукт предназначен для своего сегмента клиентов.

4.4. Система взаимодействия с клиентом

Важным блоком бизнес-архитектуры является система взаимодействия с клиентом. В качестве примеров возможных типов систем взаимодействия с потребителем можно привести следующие системы.

Персональное обслуживание. В этом случае основное внимание сосредоточено на личном контакте с клиентом. Клиента обслуживает конкретный человек. Это может осуществляться через личный контакт или через центр связи (Call center), через электронную почту (e-mail) или другими методами. Сюда можно отнести обычные магазины, где продавец находится за прилавком и обслуживает покупателя.

Специальное персональное обслуживание. Такая система основана на доверительных отношениях с индивидуальными клиентами. Примерами могут служить VIP клиенты в банке, постоянные клиенты в ресторане и другие клиенты, отношения с которыми характеризуются длительностью и доверием.

Самообслуживание. Здесь клиентам представляются средства для того, чтобы они обслужили себя сами. Например, супермаркет, где покупатели сами выбирают, взвешивают и иногда оплачивают товар.

Автоматическое обслуживание. Предполагает использование компьютерных или роботизированных технологий. Примерами могут служить интернет-банкинг или автоматическая покупка билетов на самолет. Также сюда можно отнести автоматы по продаже напитков или мелких товаров в аэропортах или на автостанциях и других местах скопления людей.

Сообщества клиентов. В этом случае клиенты объединяются в неформальные сообщества, заинтересованные в использовании и развитии продукта. В качестве примеров можно привести сообщество пользователей системы ADOIT, сообщество любителей игры LEGO и многие другие.

Совместное проектирование и производство. Здесь клиент участвует в проектировании и производстве товара. Например, клиенты могут участвовать в проектировании своего личного автомобиля, который затем изготавливается по индивидуальному заказу. Также интересные примеры такого рода бизнеса широко используются в издательском деле, где авторы принимают участие в разработке макета своей книги.

Для обозначения системы взаимодействия с клиентом будем использовать объект типа бизнес-единицы, как показано на рис. 4.18.

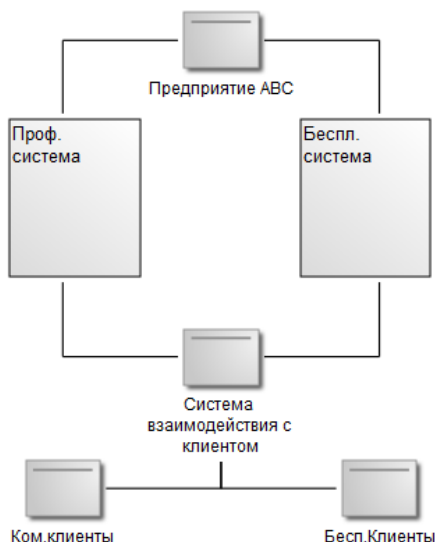


Рисунок 4.18 – Модель блока архитектуры с системой взаимодействия с клиентом

4.5. Система дистрибуции

Дистрибуцией или распределением называется процесс сбыта или доставки товара клиенту.

Важным термином при организации системы распространения товара является канал дистрибуции. Под каналом дистрибуции будем понимать совокупность организаций или отдельных лиц, которые организуют и выполняют услугу по продвижению товара от производителя к потребителю.

Различают следующие уровни каналов дистрибуции:

Канал распределения нулевого уровня: производитель сам непосредственно продает товар клиенту. Такие поставки товара еще называются прямыми.

Канал распределения первого уровня: производитель поставляет товар розничному продавцу, а тот продает товар клиенту.

Канал распределения второго уровня: производитель продает товар оптовому посреднику, тот продает товар розничному продавцу, а тот уже непосредственно клиенту.

Многоуровневые каналы еще называются косвенными.

Можно рассматривать и более сложные структуры доставки товара клиенту. Может быть больше уровней посредников, могут быть определенные производства на пути доставки товара конечному потребителю. Эти вопросы хорошо изучены в дисциплине менеджмента – цепочки поставок.

Одновременно могут использоваться несколько каналов дистрибуции разного типа.

Кроме прямой функции доставки ценности клиенту система дистрибуции может решать следующие задачи:

- повышение осведомленности среди потребителей о товарах или услугах предприятия;
- осуществление обратной связи с клиентами, например, позволять потребителям конкретизировать свои требования;
- давать возможность проводить послепродажное обслуживание.

Важным аспектом модели дистрибуции является интеграция каналов и решение вопросов собственности. Какие каналы следует оставить в собственности предприятия, а какие отдать посредникам.

Пример блока архитектуры с системами взаимодействия с клиентами и каналами дистрибуции показан на рис. 4.19.

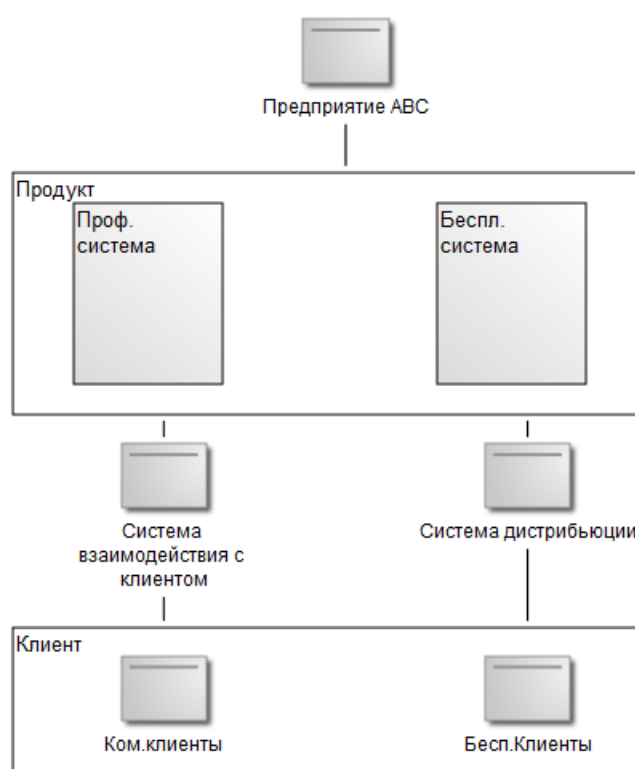


Рисунок 4.19 – Блок архитектуры «Продукт-Клиент»

4.6. Возможности (“Carability”) бизнеса

При моделировании архитектуры возможности обычно рассматриваются в широком смысле как способность совокупности организации, персонала, процессов и технологий достигать поставленных целей. Еще близкое понятие - компетентность. Здесь обычно рассматриваются следующие возможности: возможности персонала, то есть его квалификация,

компетентность, креативность и т.д.; качество и конкурентные преимущества организации бизнес-процессов; наличие современных технологий; наличие или возможность получения информации и др.

Объект "Capability" (Возможность). Объект "Возможность" в модели изображается в виде иконки, показанной на рис. 4.20, или в виде прямоугольника.



Рисунок 4.20 - Иконка для представления объекта 'Capability'

В качестве примера возможностей предприятия можно рассмотреть наличие у банка системы "Клиент - банк". Блокнот для соответствующего объекта показан на рис. 4.21.

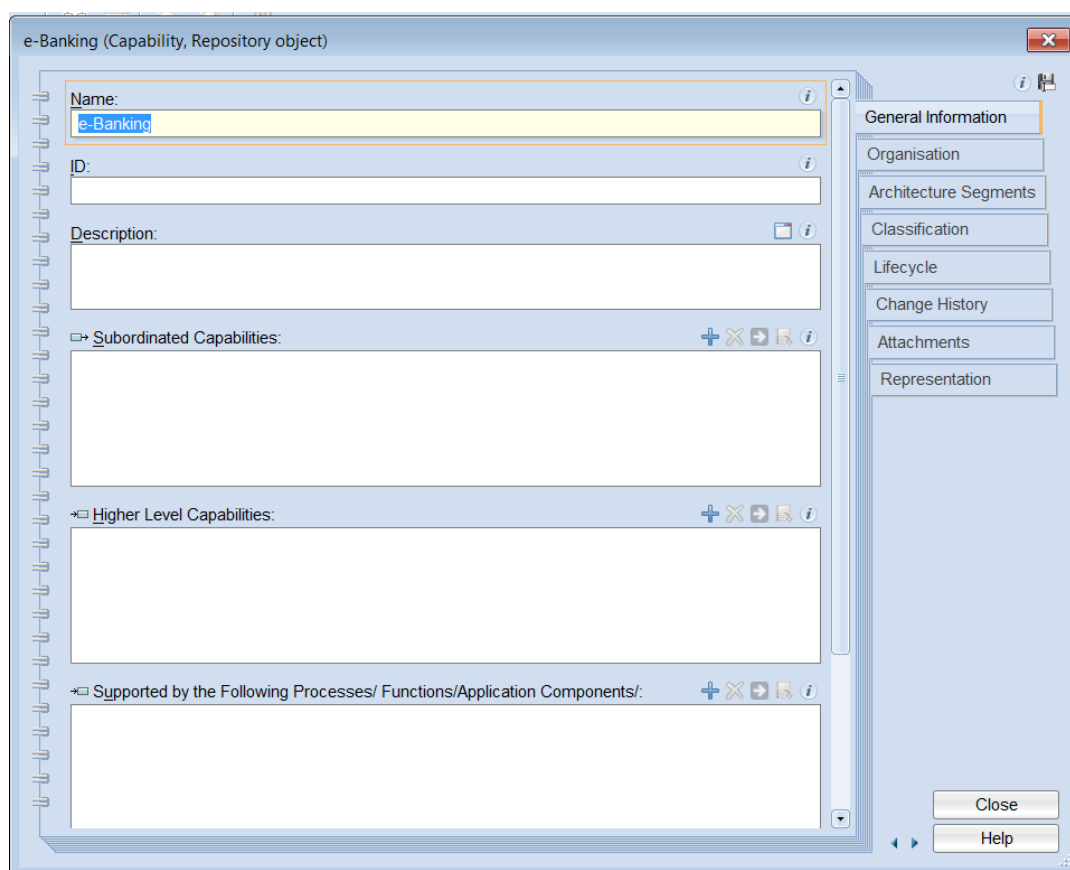


Рисунок 4.21 - Блокнот для объекта "Возможность"

На главной закладке блокнота находятся поля для связи со следующими объектами модели:

“Subordinated Capabilities” - дочерние возможности более низкого уровня;

“Higher Level Capabilities” - родительские возможности более высокого уровня;

“Supported by the Following Processes/Functions/Application Components/” - Процессы/Функции/Приложения. Здесь задаются связи с элементами архитектуры, которые поддерживают данную возможность.

Для связи возможностей с архитектурными сегментами используется простая линия "Assigned Architecture Segments" (Связанные архитектурные сегменты).

Упражнение 4. В бизнес модель архитектуры университета добавить две компетенции: "Наличие преподавателей" и "Наличие современной вычислительной техники". Вариант модели показан на рис. 4.22.

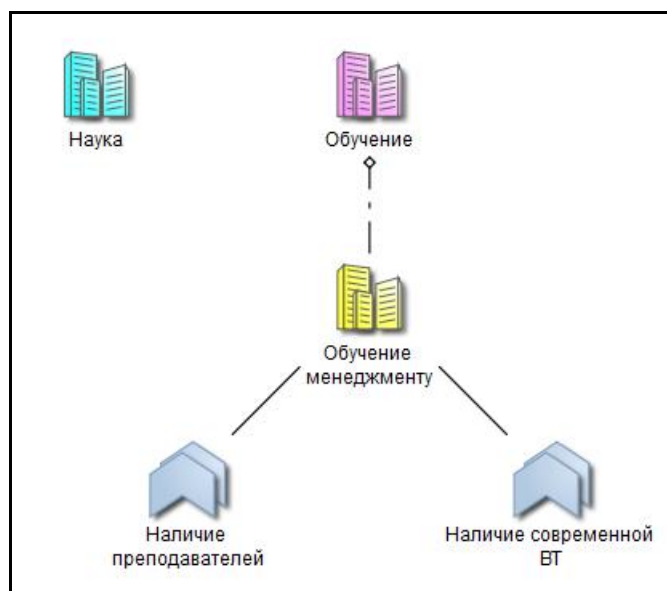


Рисунок 4.22 - Модель архитектуры предприятия с добавленными возможностями

В этом случае в закладку блокнота возможности "Architecture Segments" будет автоматически добавлена информация о новой связи, как показано на рис. 4.23.

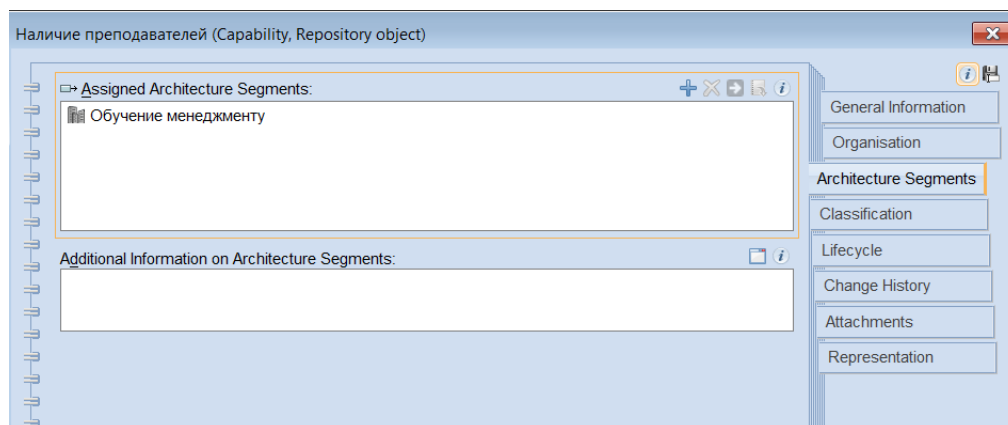


Рисунок 4.23 - Отражение связи в блокноте

При этом в блокноте для архитектурного сегмента также будет добавлена информация о новых связях, как показано на рис. 4.24.

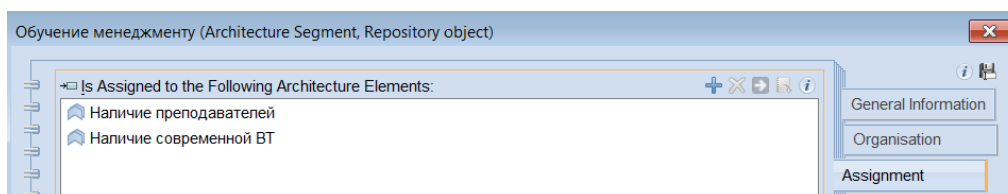


Рисунок 4.24 - Информация о связях с возможностями в блокноте для архитектурного сегмента

Вкладка блокнота "Organization", показанная на рис. 4.25, задает связь с моделью организационной структуры и указывает персоналий, которые отвечают за объект "Возможность" и за управление этим объектом, как было описано в практикуме по моделированию организационной структуры предприятия.

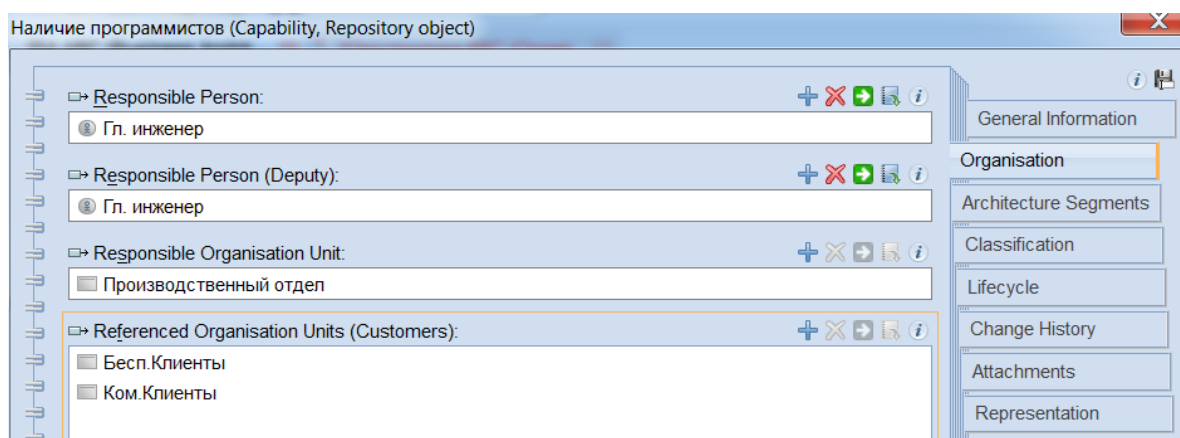


Рисунок 4.25 - Вкладка "Organization" для объекта "Возможность"

Возможности дополнительно можно связывать с ответственными за них бизнес-единицами (Responsible Organization Unit) и связанными бизнес-единицами клиентов (Referenced Organization Units (Customers)). На рис. 4.25 показана связь возможности с клиентами предприятия.

Особый интерес представляет вкладка “Classification” (Классификация), которая показана на рис. 4.26.

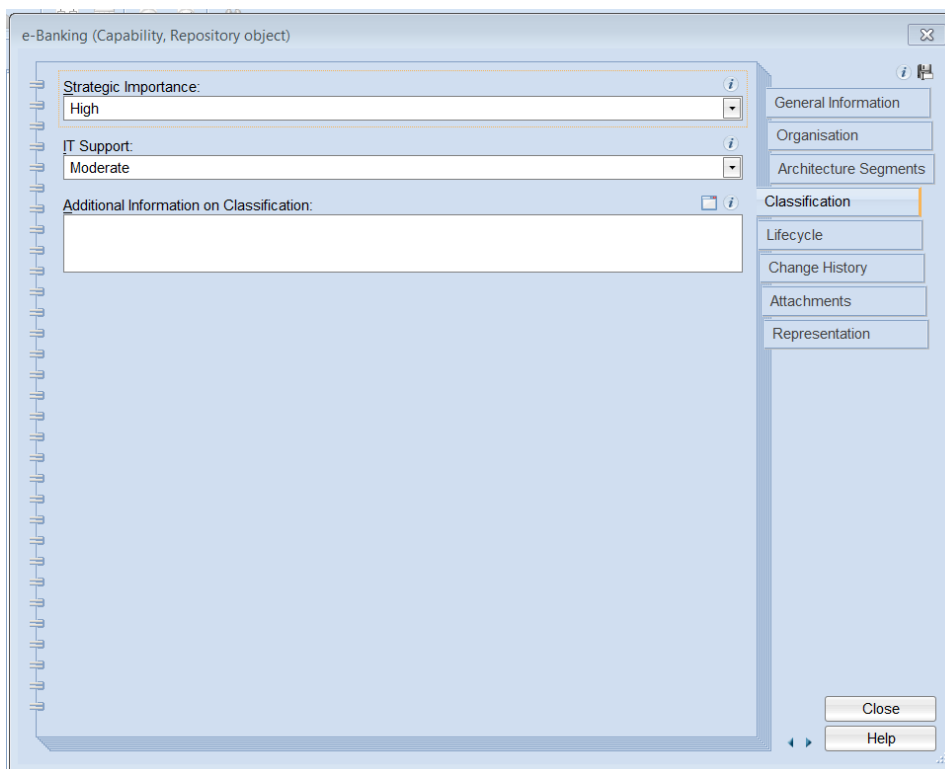


Рисунок 4.26 - Вкладка 'Classification'

Вкладка позволяет провести определенный классификационный или кластерный анализ зависимости возможности с другими характеристиками предприятия. Вкладка имеет следующие, описанные ниже поля.

Поле “**Strategic Importance**” – задает важность компетенции предприятия для стратегии предприятия. На рис. 4.27 показан список уровней для выбора возможного значения этого параметра.

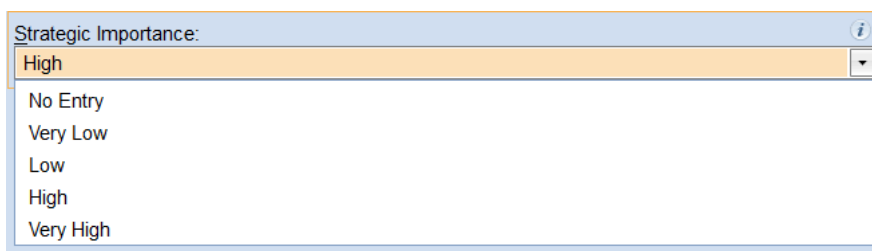


Рисунок 4.27 - Уровни стратегической важности:

Параметр может принимать следующие значения:

- “No Entry” - уровень не определен;
- “Very Low” - очень низкий уровень;
- “Low” – низкий уровень;
- “High” – высокий уровень;
- “Very High” - очень высокий уровень.

Поле **“IT Support”** – показывает степень поддержки со стороны системы информационных технологий предприятия. На рис. 4.28 показан список уровней с возможными значениями этого параметра.

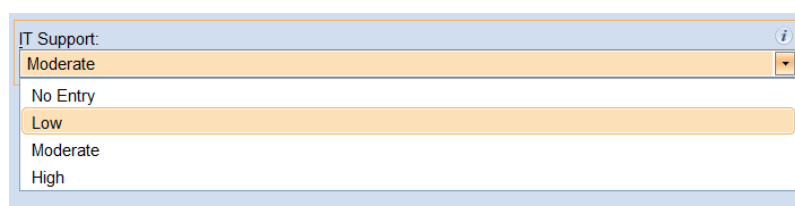


Рисунок 4.28 - Уровни IT поддержки

Параметр может принимать следующие значения:

- “No Entry” - уровень не определен;
- “Low” – низкий уровень;
- “Moderate” – средний уровень;
- “High” – высокий уровень.

В системе ADOit для объекта "Возможность (Capability)" имеется возможность графического представления результатов классификации объектов этого класса по указанным выше параметрам. Для получения такого представления необходимо отметить мышью нужные объекты (Как обычно удерживая клавишу Ctrl), как показано на рисунке 4.29.



Рисунок 4.29 - Выбор объектов для классификации

Открыть контекстное меню правой кнопкой мыши и выбрать пункты меню Generate view/Portfolio/Capability[Strategic Importance - IT Support], как показано на рис. 4.30.

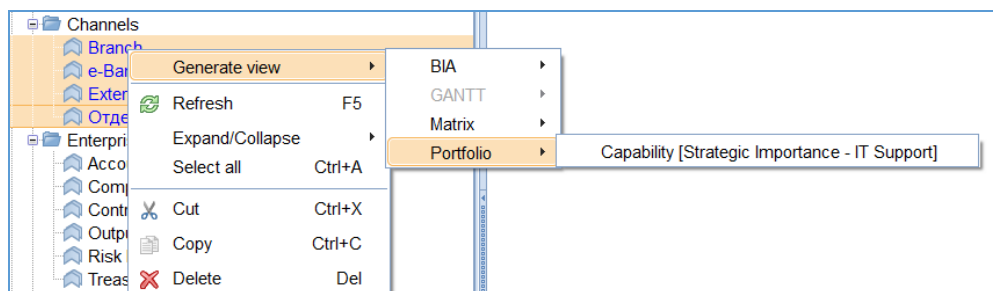


Рисунок 4.30 - Выбор графического представления

В результате будет сгенерирована пузырьковая диаграмма, показанная на рис. 4.31.

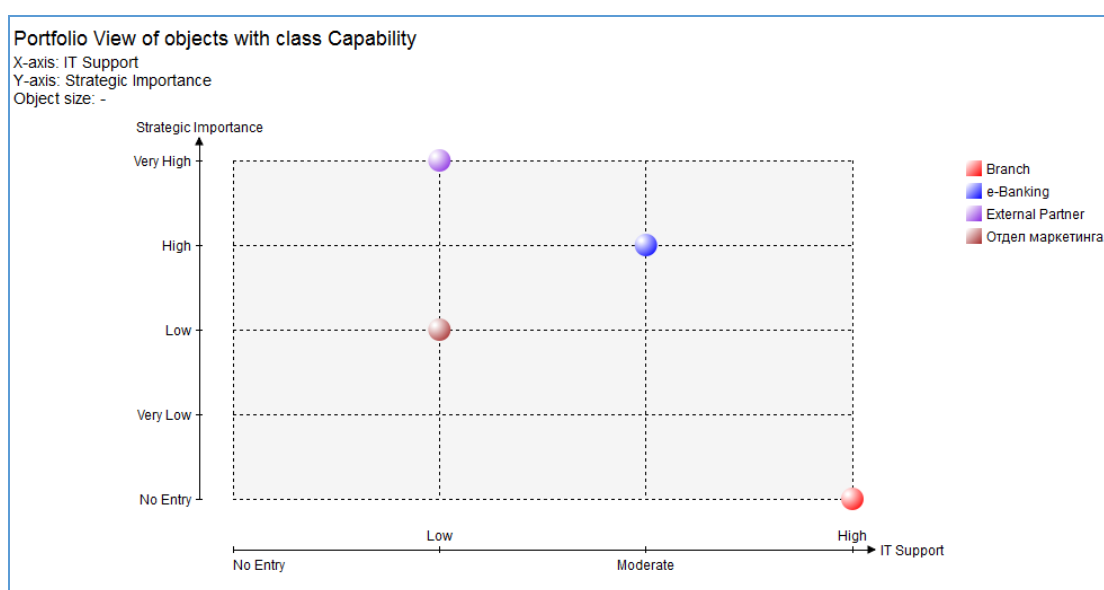


Рисунок 4.31 - Классификационная диаграмма для объектов класса "Возможность (Capability)"

Диаграмма дает возможность визуально определить группы наиболее значимых для стратегии возможностей и хорошо обеспеченных информационной поддержкой. Если загрузить данные о возможностях в электронную таблицу, то можно использовать более сложные аналитические методы анализа, такие как ранговый коэффициент корреляции Кендалла, таблицы сопряженности признаков, методы классификации признаков и другие.

Упражнение 5. Создать простейшую модель бизнес-архитектуры, показанную на рис. 4.32 со следующими четырьмя возможностями: «Наличие программистов», «Наличие вычислительной техники», «Опыт работы на рынке» и «Хорошее знание английского языка».

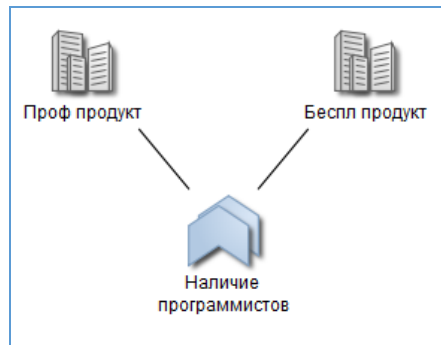


Рисунок 4.32 – Модель для упражнения

Для каждой возможности необходимо задать значения атрибутов **“Strategic Importance”** и **“IT Support”**. Значения выбрать произвольно. Построить пузырьковую диаграмму для этих атрибутов.

4.7. Бизнес функции

В отличие от возможностей функции бизнеса сфокусированы только на текущей деятельности предприятия, тогда как возможности направлены не только на текущую деятельность, но и на потенциальное поведение предприятия. Иногда используемые в реальности возможности предприятия называются функциональными возможностями (functional capability), а перспективные возможности динамическими возможностями (Dynamic capability). При этом возможности могут использоваться несколькими функциями предприятия и при реализации функции могут использовать несколько возможностей. Например, способность управления рисками может использоваться как функцией управления рисками бизнеса, так и функцией управления ИТ рисками. Функции тесно связаны с бизнес-процессами, тогда как возможности могут быть независимы от бизнес-процессов. Например, наличие патентов или специфических знаний. Функции тесно связаны с бизнес-единицами предприятия. Способности относятся более ко всему предприятию. Возможности производят декомпозицию бизнеса по целям бизнеса, а функции производят декомпозицию бизнеса по действиям (Activity).

Для представления бизнес функции в модели используется объект 'Function' - функция. Иконка для этого объекта показана на рис. 4.33.



Рисунок 4.33 - Иконка для объекта 'Function'

Объект «Функция» также может быть представлен в виде прямоугольника, как показано на рис. 4.34. В этом случае удобно представлять слоты, входящие во фрейм "Функция" путем размещения их в прямоугольнике.

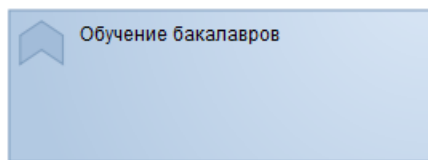


Рисунок 4.34 - Представление объекта "Функция" в виде прямоугольника

Блокнот для объекта "Функция" показан на рис. 4.35.

Рисунок 4.35 - Блокнот для объекта "Функция"

Как обычно на вкладке общей информации (General Information) имеются поля связи с другими функциями (Assigned Functions) и с функциями более высокого уровня (Higher Level Functions). Также имеется маркер, что для этой функции требуется действие (Action Required) и поле пояснения зачем (Reason). Если этот маркер активен, то на изображении объекта ставится значок с восклицательным знаком, как показано на рис. 4.36.

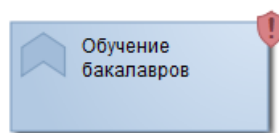


Рисунок 4.36 - Объект "Функция" с маркером необходимости действия

Вкладки "Processes" (Процессы) и 'Implementation' (Реализация) будут рассматриваться в следующих практикумах. Остальные вкладки имеют структуру, описанную в предыдущих разделах.

Функции и возможности можно связать отношением "Supported Capabilities" (Поддерживаемые возможности). Отображается простой линией с ромбиком на конце, как показано на рис. 4.37.



Рисунок 4.37 - Связь между функцией и возможностью.

Одновременно запись о созданной связи отражается в блокноте для функции (рис. 4.38) и в блокноте для возможности (рис. 4.39).

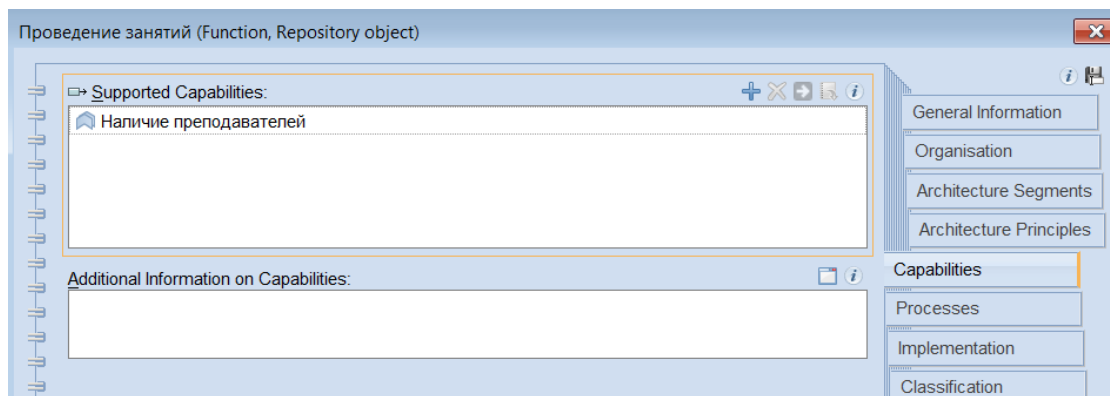


Рисунок 4.38 - Запись о связи в блокноте для функции

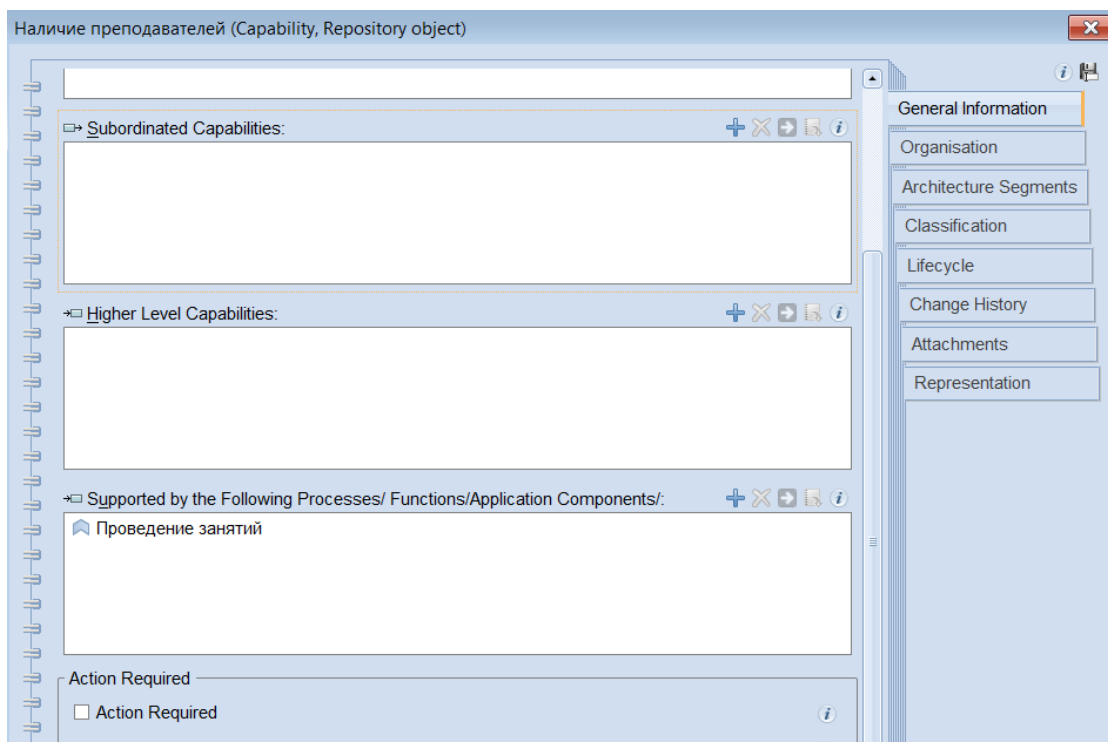


Рисунок 4.39 - Запись о связи в блокноте для возможности

Следует не забывать, что при удалении графического элемента, записи из хранилища и блокнота не удаляются.

Функции естественно должны быть связаны с исполнителями, то есть либо с бизнес-единицами, либо с персоналиями. К сожалению, рассматриваемый вариант системы не имеет возможности построить графическую модель с этими объектами в модели архитектуры бизнеса. Для решения этой задачи, как и ранее, воспользуемся возможностями модели BIA (Анализ связей). На рис. 4.40 показана последовательность контекстных меню, чтоб иметь возможность построить модель BIA и использовать все графические элементы системы ADOITCE.

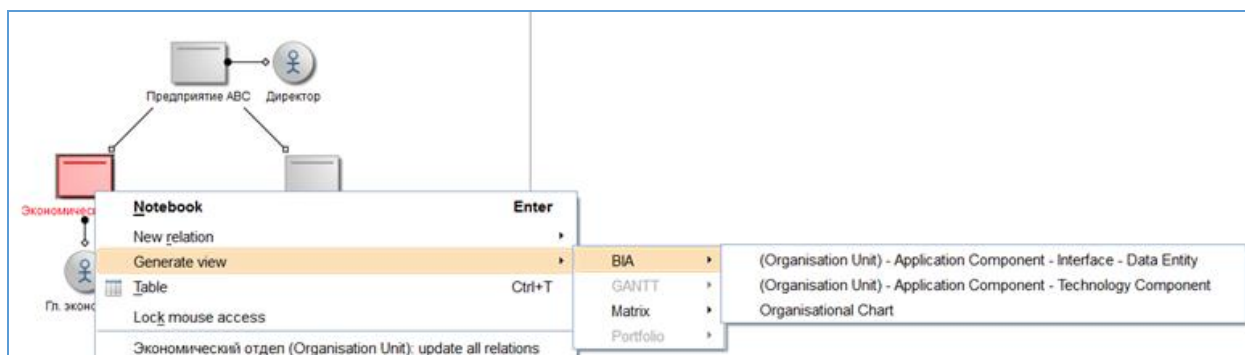


Рисунок 4.40 – Последовательность действий для проведения анализа связей

При выборе пункта последнего меню “Organization Chat” будет создана модель, показанная на рис. 4.41.

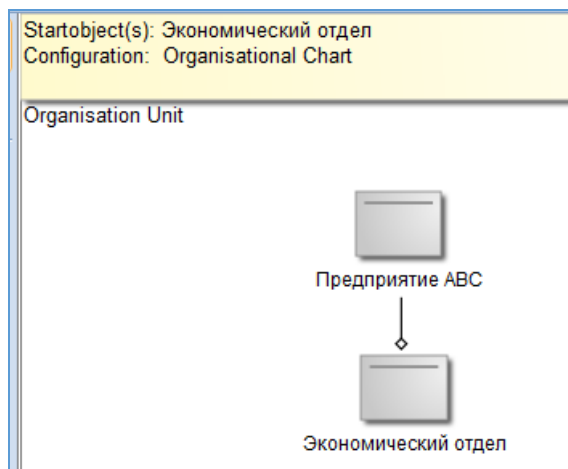


Рисунок 4.41 – Модель организационной структуры, созданная в режиме ВИА

Модель немного отредактирована для лучшего представления в тексте.

Введем для экономического отдела с помощью расширенной палитры инструментов следующие три компетенции (возможности): «Наличие квалифицированных бухгалтеров», «Наличие системы 1С» и «Наличие системы безопасности». Результат этого действия показан на рис. 4.42.

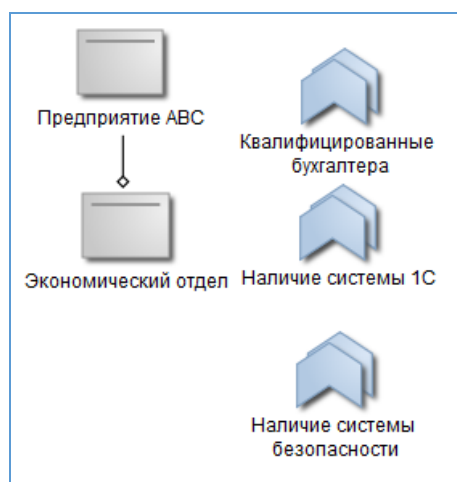


Рисунок 4.42 – Возможности экономического отдела

Комплексная модель в палитре имеет все возможные виды связей в системе. Это затрудняет выбор нужной связи. Для связи возможностей и бизнес-единицы значительно удобнее воспользоваться контекстным меню, открываемом правой кнопкой мыши на объекте. Так для объекта типа

«Возможность» открывается меню со всеми возможными связями для этого объекта, как показано на рис. 4.43.

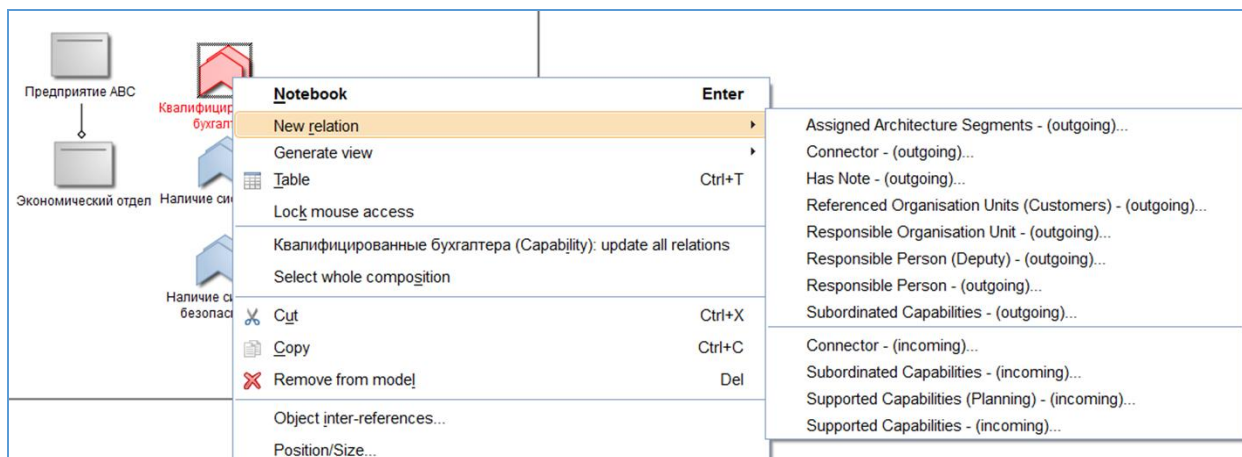


Рисунок 4.43 – Выбор типа связи для объекта

После выбора связи “Responsible Organization Unit – (outgoing)” (Исходящая связь на ответственную организационную единицу) открывается окно, показанное на рис. 4.44.

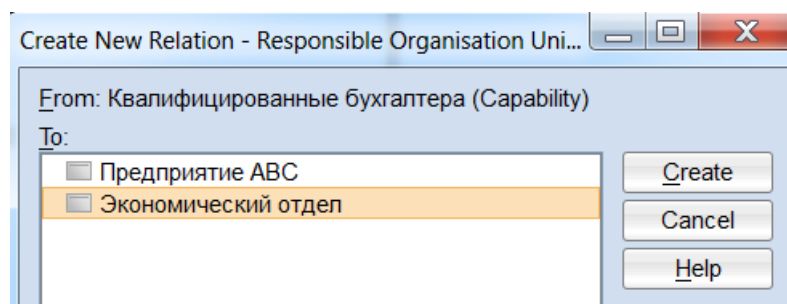


Рисунок 4.44 – Окно для выбора связанного объекта

В окне удерживая клавишу “Ctrl” можно отметить несколько объектов и связи будут установлены со всеми.

Добавим в модель четыре функции, которые выполняет экономический отдел, и свяжем их с возможностями. В результате получится модель фрагмента бизнес-архитектуры, показанная на рис. 4.45. Следует обратить внимание, что одни и те же возможности используются экономическим отделом для выполнения разных функций.

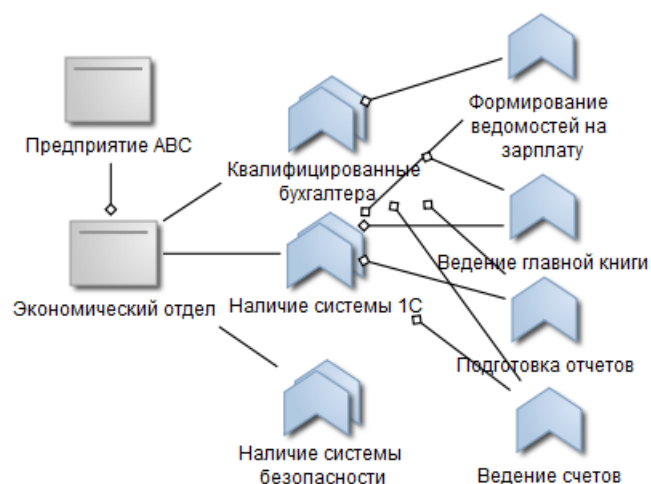


Рисунок 4.45 – Возможности и конкретные функции, выполняемые экономическим отделом

В блокнотах всех объектов будут добавлены соответствующие связи, как показано на рис. 4.46 для функции «Формирование ведомостей на зарплату».

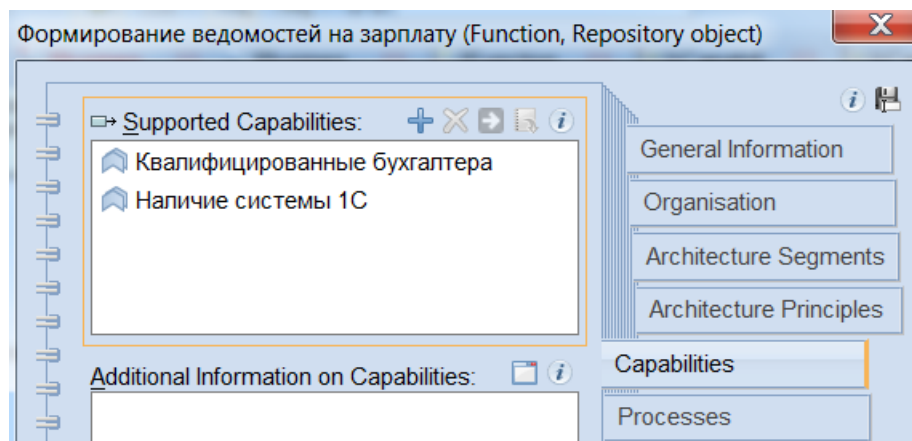


Рисунок 4.46 – Автоматическое добавление связей в блокнот

В рассматриваемом примере существуют две компетенции экономического отдела, которые используются для выполнения всех четырех функций. Третья компетенция существует, но не связана с прямыми функциями отдела.

Любой объект модели можно скопировать и использовать в модели как новый объект (New object) или как тот же самый объект (Reuse object) для повторного применения. При попытке вставить в модель скопированный объект система выдает сообщение, показанное на рис. 4.47.

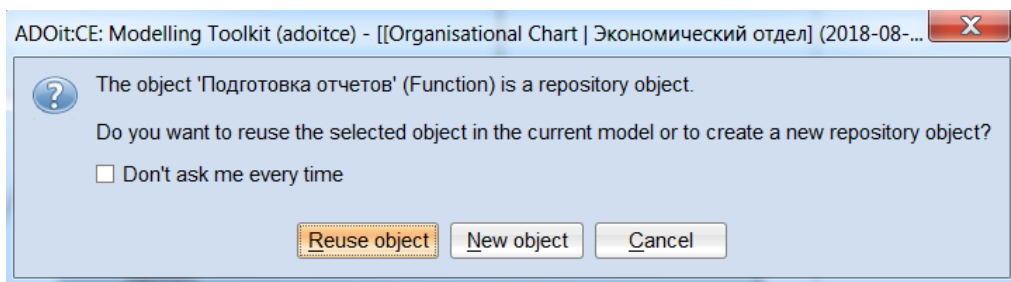


Рисунок 4.47 – Сообщение о режиме использования скопированного объекта

Схема блока архитектуры с возможностями и функциями экономического отдела можно представить и в другом, возможно более удобном виде, показанном на рис. 4.48.

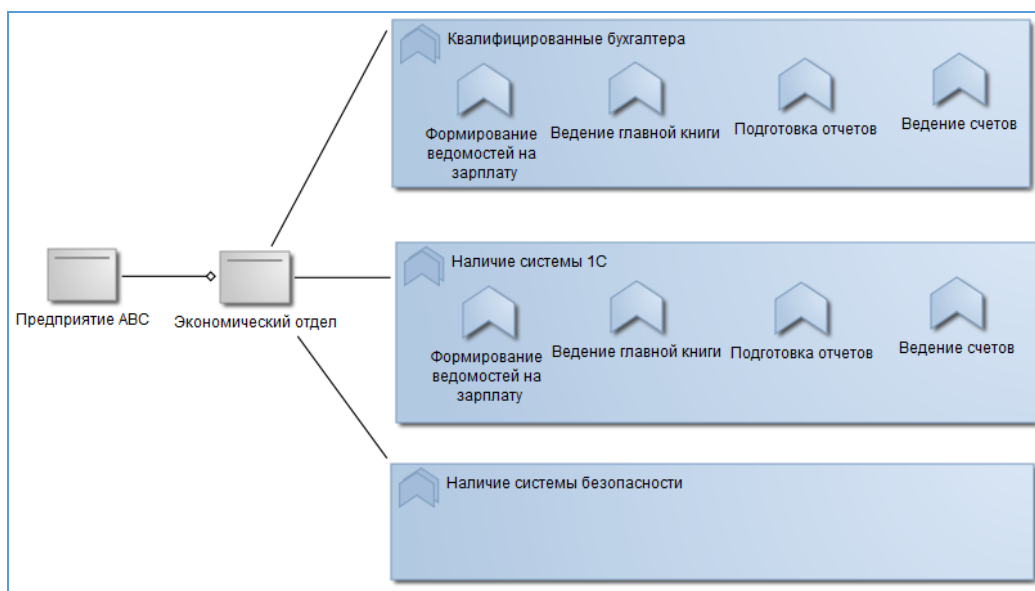


Рисунок 4.48 – Другое представление схемы возможностей и функций экономического отдела

На этой схеме для удобства возможности представлены в виде прямоугольников, а функции скопированы и вставлены в эти прямоугольники.

Окончательно рассмотренную часть бизнес-архитектуры можно представить в виде семантической схемы, показанной на рис. 4.49.

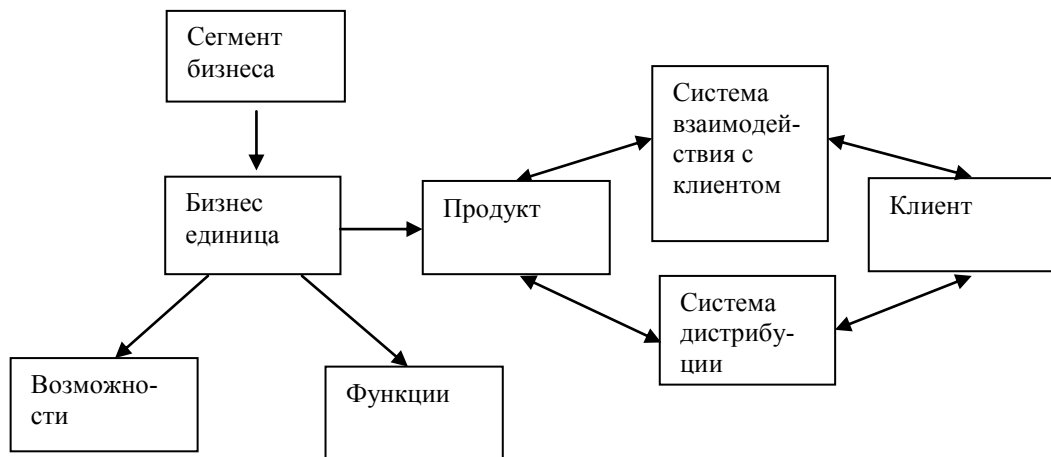


Рисунок 4.49 - Семантическая схема объектов, рассмотренных в практикуме

4.8. Задание по выполнению практикума

- Выполнить упражнения
- Создать три архитектурных сегмента бизнеса, как показано на рисунке 4.50.



Рисунок 4.50 - Архитектурные сегменты задания

Перенести записи об этих сегментах в новую папку, как показано на рисунке 4.51.

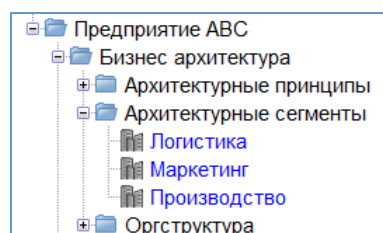


Рисунок 4.51 - Папка с бизнес сегментами

- Создать модель с четырьмя возможностями по указанному преподавателем варианту и связать их с сегментами и бизнес-единицами, созданными в предыдущем практикуме. Номера возможностей приведены в табл. 1, а варианты заданий в табл. 2. Допускается самостоятельная модификация возможностей, если они будут больше подходить к сегментам бизнеса.

- Подготовить отчет

4.9. Содержание отчета

Отчет должен содержать следующую информацию: задание на выполнение практикума, копии окон экрана с выполненными упражнениями, копии окон экрана с выполненным индивидуальным заданием, выводы.

4.10. Варианты заданий

Таблица 1 - Список возможностей

№ п.п.	Возможность
1	Квалифицированный персонал
2	Наличие патентов
3	Финансовые возможности
4	Наличие системы CRM
5	Наличие системы OLAP
6	Наличие системы ERP
7	Наличие системы BPM
8	Наличие аналитической службы
9	Уникальный продукт

Таблица 2 - Варианты выполнения работы

№ Вар.	№ Возм.1	№ Возм.2	№ Возм.3	№ Возм.4
1	1	2	3	4
2	5	6	7	8
3	9	1	2	3
4	4	5	6	7
5	8	9	1	2
6	3	4	5	6
7	7	8	9	1
8	2	3	4	5
9	6	7	8	9
10	1	3	5	7
11	9	1	3	5
12	7	9	1	3
13	5	7	9	1

14	3	5	7	9
15	2	4	6	8

Таблица 3 - Список функций

№ п.п.	Функция
1	Анализ работы отдела
2	Планирование работы отдела
3	Организация работы отдела
4	Мотивация персонала
5	Обучение персонала
6	Поиск работников
7	Работа с клиентами
8	Заказ рекламы
9	Анализ эффективности рекламы
10	Финансовое планирование
11	Стратегическое планирование
12	Управление проектами
13	Выполнение заказов
14	Работа с поставщиками
15	Работа с партнерами
16	Проведение совещаний

Таблица 4 - Варианты выполнения работы

№ Вар.	№ Функци.1	№ Функци.2	№ Функци.3	№ Функци.4	№ Функци.5
1	1	2	3	4	5
2	6	7	8	9	10
3	11	12	13	14	15
4	16	1	2	3	4
5	5	6	7	8	9
6	10	11	12	13	14
7	15	16	1	2	3
8	4	5	6	7	8
9	9	10	11	12	13
10	14	15	16	1	2
11	3	4	5	6	7
12	8	9	10	11	12
13	13	14	15	16	1
14	2	3	4	5	6
15	7	8	9	10	11

4.11. Контрольные вопросы

1. Что такое сегментация рынка?
2. Что такое лояльность клиента?
3. Виды сегментов рынка.
4. Что такое позиционирование товара?
5. Какие существуют виды отношений с клиентами?
6. Что такое дистрибуция?
7. Что такое логистика?
8. Виды каналов дистрибуции.
9. Что такое возможность бизнеса?
10. Что такое функция бизнеса?
11. Чем функция бизнеса отличается от возможности?
12. Что такое ВІА?

Глава 5. Лабораторный практикум «Моделирование архитектуры бизнес-процессов»

5.1. Архитектура бизнес-процессов

Архитектура бизнес-процессов является одной из основных составляющих бизнес-архитектуры. Часто термины «архитектура бизнес-процессов» и «система бизнес-процессов» используются в качестве синонимов.

Как правило, система бизнес-процессов предприятия строится в виде иерархической структуры, которая может быть представлена либо в виде графической модели, либо в виде иерархического справочника процессов.

Архитектура бизнес-процессов должна помогать решать такие задачи, как: оптимизация модели бизнеса; совершенствование организационной структуры; описание и регламентация бизнес-процессов; управление бизнес-процессами; автоматизация бизнес-процессов и др.

Бизнес-процессы в системе ADOit строятся в модели архитектуры бизнеса, отображаются на верхнем уровне и в обобщенном виде. Для представления объекта типа "Процесс" может использоваться три формы.

Первое представление в виде иконки, как показано на рис. 5.1.

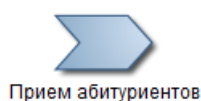


Рисунок 5.1 - Представление процесса в виде простой иконки

Второе представление в виде удлиненной стрелки, как показано на рис. 5.2.

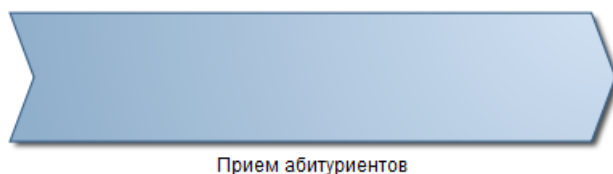


Рисунок 5.2 - Представление процесса в виде удлиненной стрелки

Третье представление в виде прямоугольника, как показано на рис. 5.3. В этом случае можно в прямоугольнике показывать подчиненные процессы.

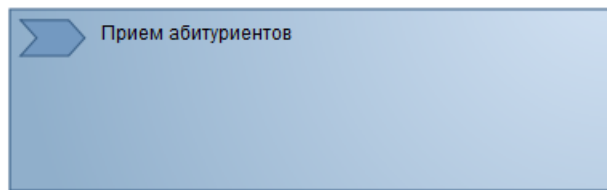


Рисунок 5.3 - Представление процесса в виде прямоугольника

На рис. 5.4 показан блокнот для объекта "Процесс". Форма представления как обычно задается на вкладке "Representation".

Рисунок 5.4 - Блокнот для объекта "Процесс"

В поле 'Relevant for Compliance' - маркер ставится, если необходима постоянная проверка отдельных операций процесса на предмет соответствия установленным требованиям, условиям контракта или заявленному описанию.

Поле 'Subordinated Processes' содержит список процессов, связанных с данным на следующем более низком уровне иерархии.

Поле 'Higher Level Processes' содержит список процессов, связанных с данным на более высоком уровне иерархии.

Поле 'Is Supported by the Following Architecture Elements' содержит список функций и приложений ИТ, которые поддерживают данный процесс.

Поле "Action Required" как обычно показывает необходимость выполнения действия, связанного с этим процессом. Пример заполнения этого поля показан на рис. 5.5.

Рисунок 5.5 - Маркер необходимости дополнительного действия и поле пояснения

Иерархия процессов может отражаться на графической модели с помощью элемента связи 'Subordinated Processes' - пунктирной линией с ромбиком . Пример графической модели связанных бизнес-процессов показан на рис. 5.6.

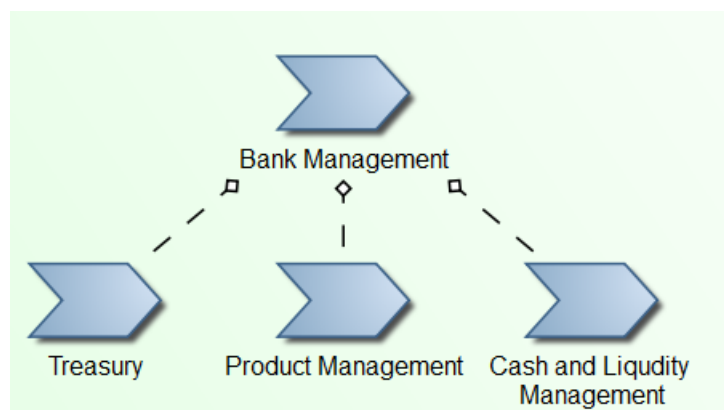


Рисунок 5.6 - Иерархия процессов

На графической модели можно также показать связь с архитектурным сегментом, как показано на рис. 5.7. Для этого используется простая линия, но с именем "Assigned Architecture Segments".

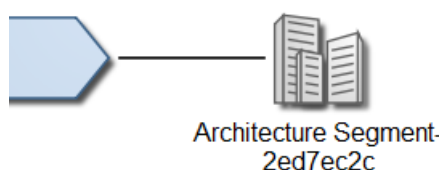


Рисунок 5.7 - Связь процесса и архитектурного сегмента

Одновременно связь будет отражена в закладке "Architecture Segments", как показано на рис. 5.8.

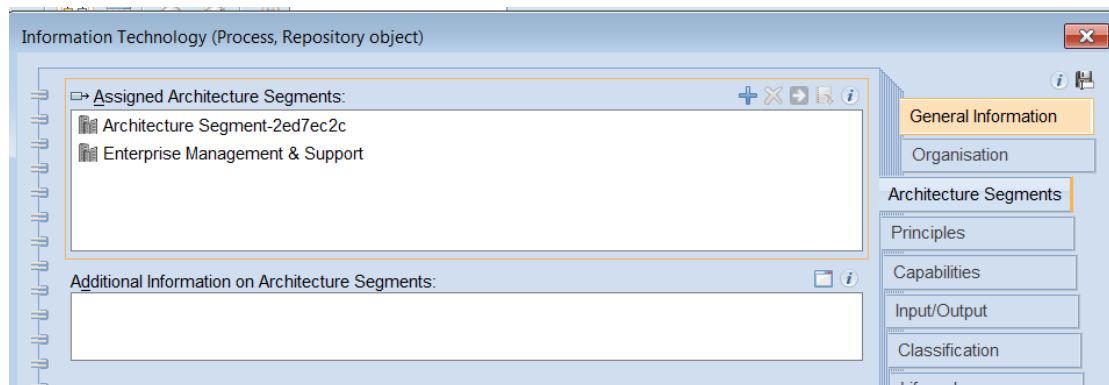


Рисунок 5.8 - Список связей с архитектурными сегментами в блокноте

Процессы могут связываться с возможностями. На графической модели эта связь отражается с помощью прямой линии с ромбиком - "Supported Capabilities", как показано на рис. 5.9.

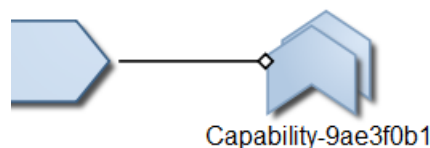


Рисунок 5.9 - Связь процесса и возможности

Одновременно эта связь отражается в списке связей закладки "Capabilities", как показано на рис. 5.10.

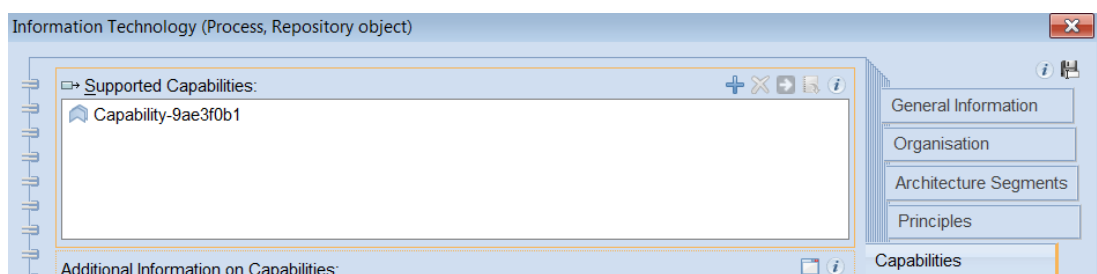


Рисунок 5.10 - Список связей с возможностями в блокноте для процесса

На рис. 5.11 показана вкладка "Classification" для процесса. Здесь кроме уже известных полей "Strategic Importance" - важность для стратегии и "IT Support" - поддержка со стороны информационных технологий, добавлены два новых поля.

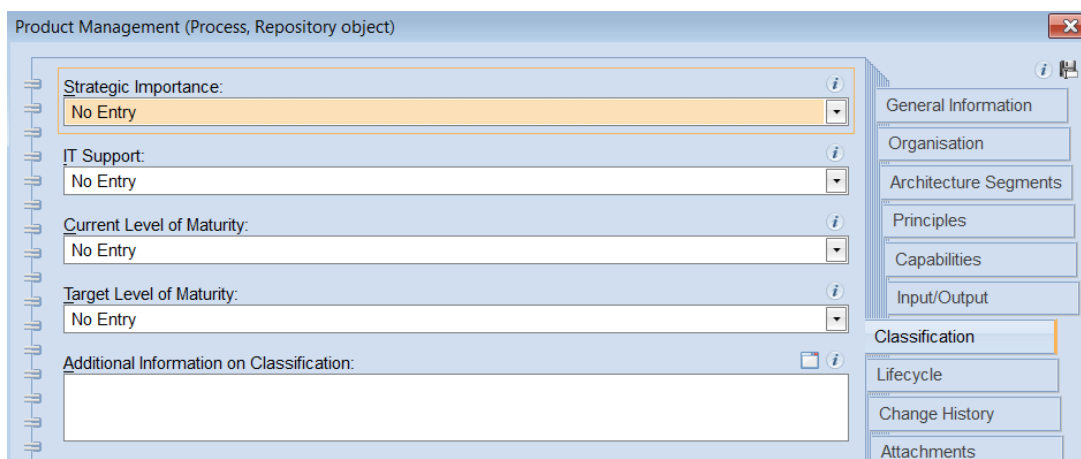


Рисунок 5.11 - Вкладка "Classification" для процесса

Значения поля "Current Level of Maturity" - текущий уровень зрелости, показаны на рис. 5.12.

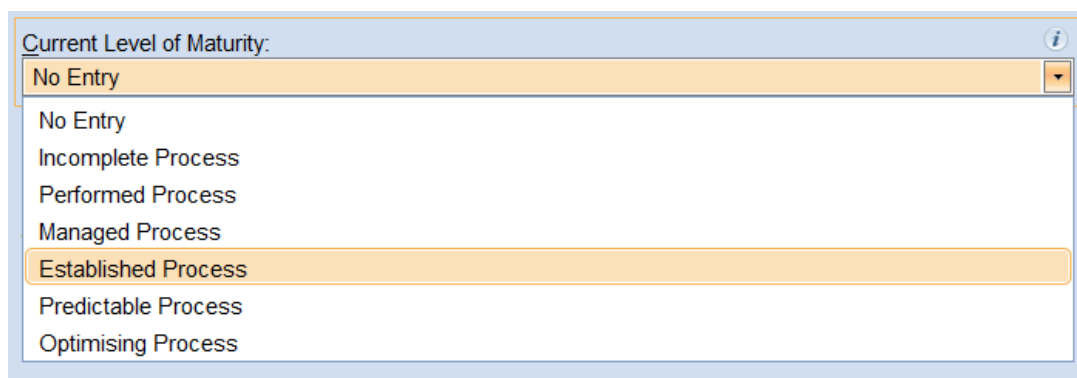


Рисунок 5.12 - Возможные значения параметра "Current Level of Maturity"

Рассмотрим значения полей.

"No Entry" – означает, что уровень зрелости процесса не определен.

"Incomplete process" - процесс не оформлен полностью.

"Performed process" - процесс достаточно подробно промоделирован и подготовлены регламенты процесса.

"Managed process" - процесс внедрен в работу. Он отрегулирован, управляется с использованием функций планирования и мониторинга. Продукты процесса отлажены, контролируются и нормально производятся.

"Established process " - процесс вошел в стационарную устойчивую фазу.

"Predictable process" - процесс четко работает в условиях ограничений и возможных изменений окружающих условий.

"Optimizing process" - процесс постоянно улучшается и оптимизируется.

Второе поле "Target Level of Maturity" - целевое значение уровня процесса используется для стратегического планирования развития архитектуры предприятия. Значения поля показаны на рис. 5.13. Они совпадают со значениями для текущего состояния процесса.



Рисунок 5.13 - Возможные значения параметра "Target Level of Maturity"

В системе ADOit для объекта "Процесс" имеется возможность построить графическое представление классификации объектов этого класса. Для получения такого представления необходимо отметить мышью нужные объекты (Как обычно удерживая клавишу Ctrl), как показано на рис. 5.14.



Рисунок 5.14 - Выбор объектов для классификации

Затем выбрать пункты меню Generate view/Portfolio/Process[Strategic Importance - IT Support], как показано на рис. 5.15.

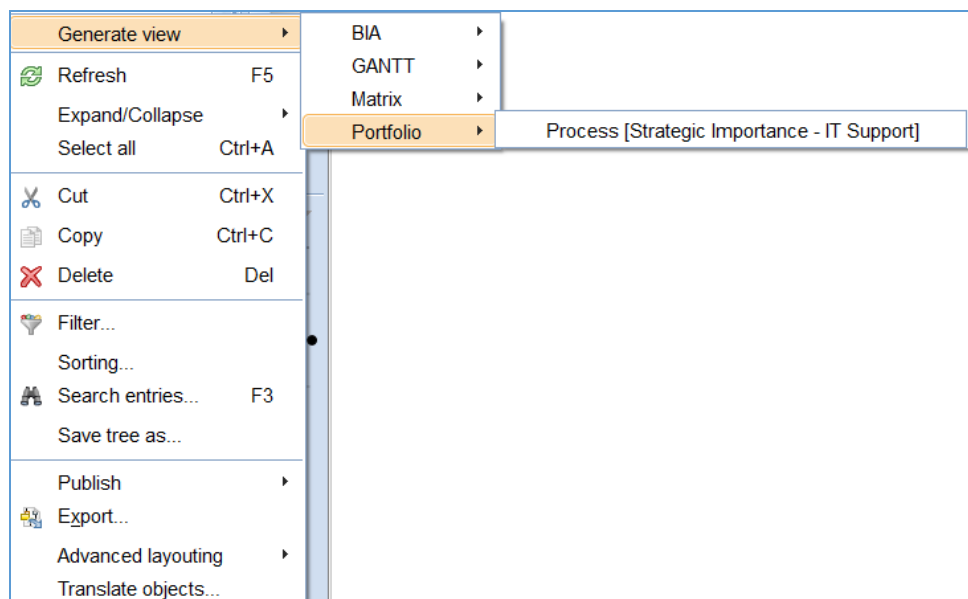


Рисунок 5.15 - Выбор представления в виде пузырьковой диаграммы

В результате будет сгенерирована пузырьковая диаграмма, показанная на рис. 5.16.

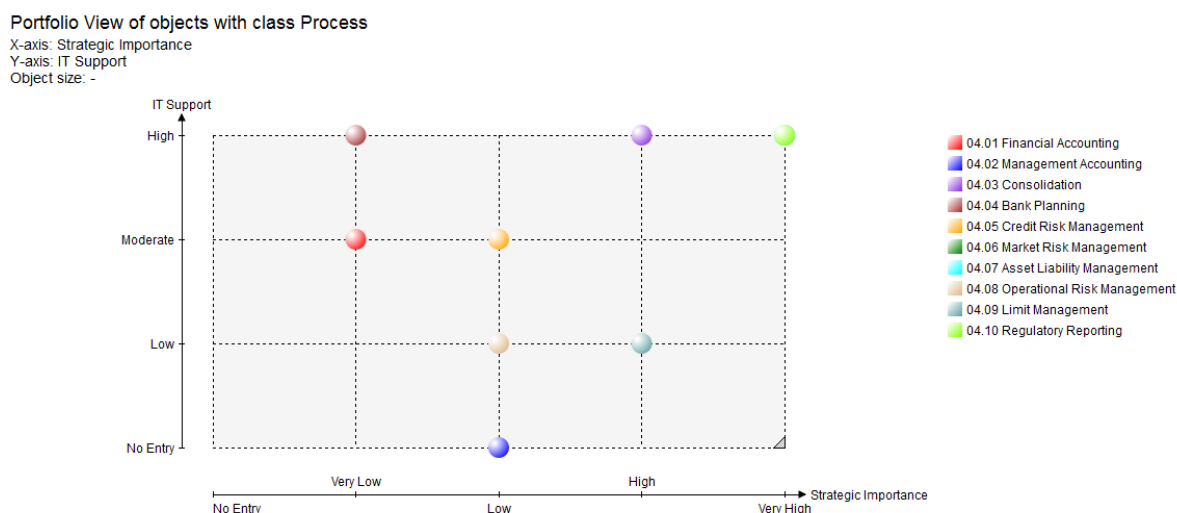


Рисунок 5.16 - Пузырьковая диаграмма для процессов

Как видно из рис. 5.15 для представления процессов существуют кроме пузырьковой диаграммы еще три вида представлений. Диаграмма Ганта будет рассмотрена позже, после изучения архитектуры информационных технологий.

При выборе пункта меню BIA (Business Impact Analysis) откроется меню для автоматического построения графических моделей, как показано на рис. 5.17. В данном практикуме можно воспользоваться последним пунктом - "Process Hierarchy" (Иерархия процессов). Процессы для кото-

рых строится представление должны быть отмечены, как показано на рис. 5.14.

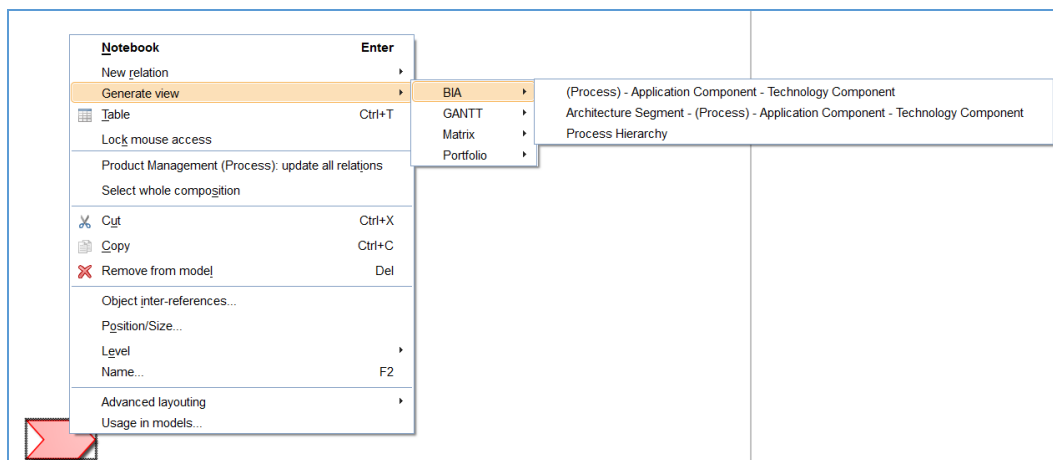


Рисунок 5.17 - Меню для выбора представления

В результате система автоматически сгенерирует графическую модель, в которой будут отражены связи между процессами, как показано на рис. 5.18.

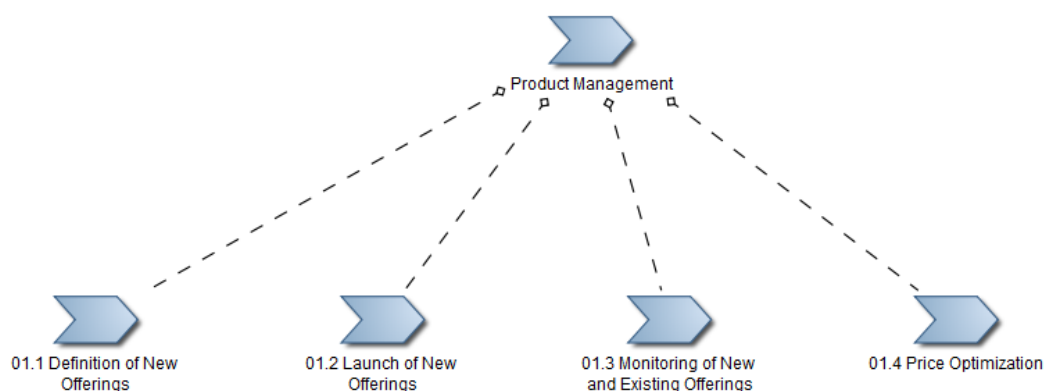


Рисунок 5.18 - Сгенерированная модель бизнес-процессов

5.2. Связь процессов с функциями

Модель архитектуры бизнеса может поддерживать связи между функциями бизнеса и бизнес-процессами. Графически связь показывается простой линией с именем "Supported processes", как показано на рис. 5.19.

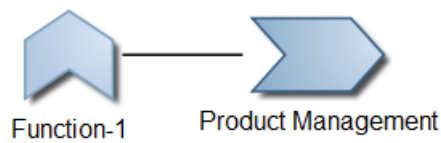


Рисунок 5.19 - Связь между функцией и бизнес-процессом

Список связей одновременно отражается во вкладке "Processes" блокнота для функции, как показано на рис. 5.20.

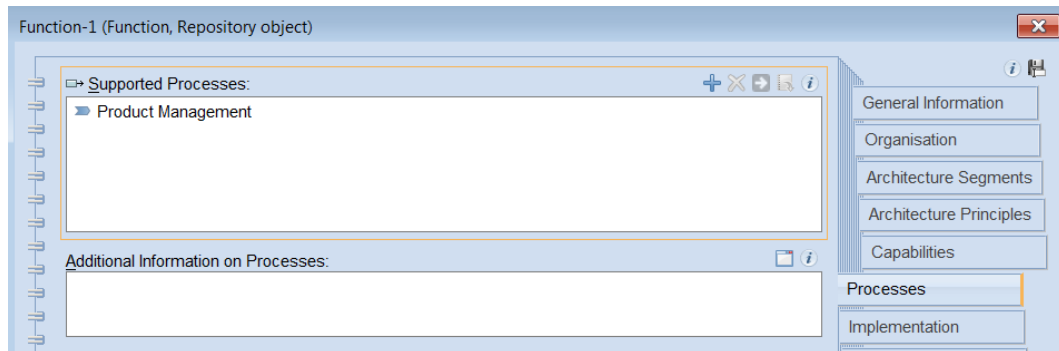


Рисунок 5.20 - Список связей функции и процессов

В системе есть возможность для функций строить автоматически графические модели зависимости, таким же образом, как было описано для процессов. Пример представления VIA для функций показан на рис. 5.21.

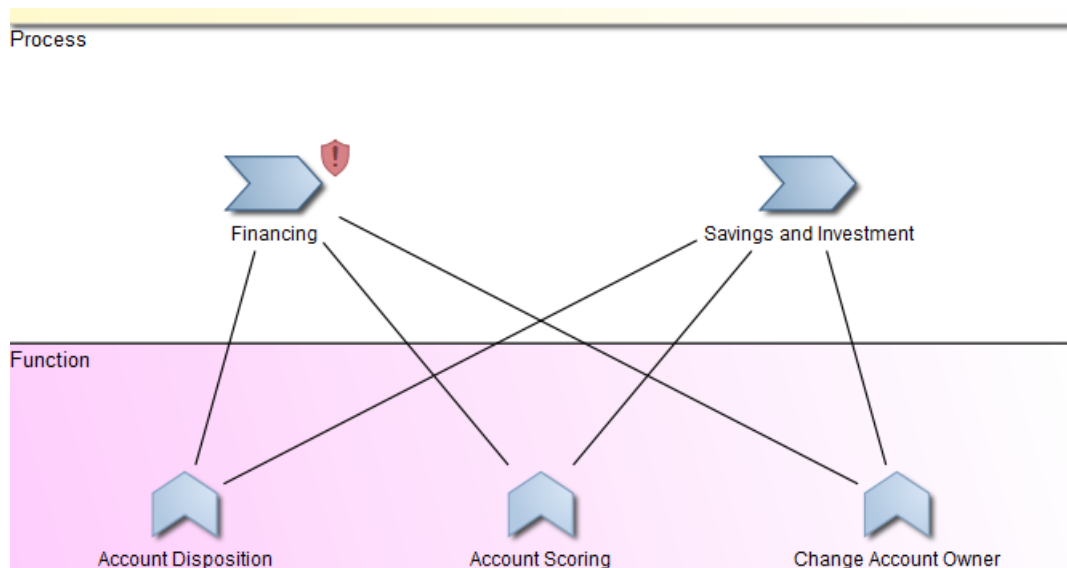


Рисунок 5.21 - Пример анализа зависимостей (VIA)

5.3. Модель данных

Объект "Данные" ('Data Entity') в бизнес архитектуре предназначен для представления определенной группы информации. Графическое представление объекта показано на рис. 5.22.



Рисунок 5.22 - Графическое изображение объекта "Данные"

Для объекта "Данные" блокнот имеет вид, показанный на рис. 5.23.

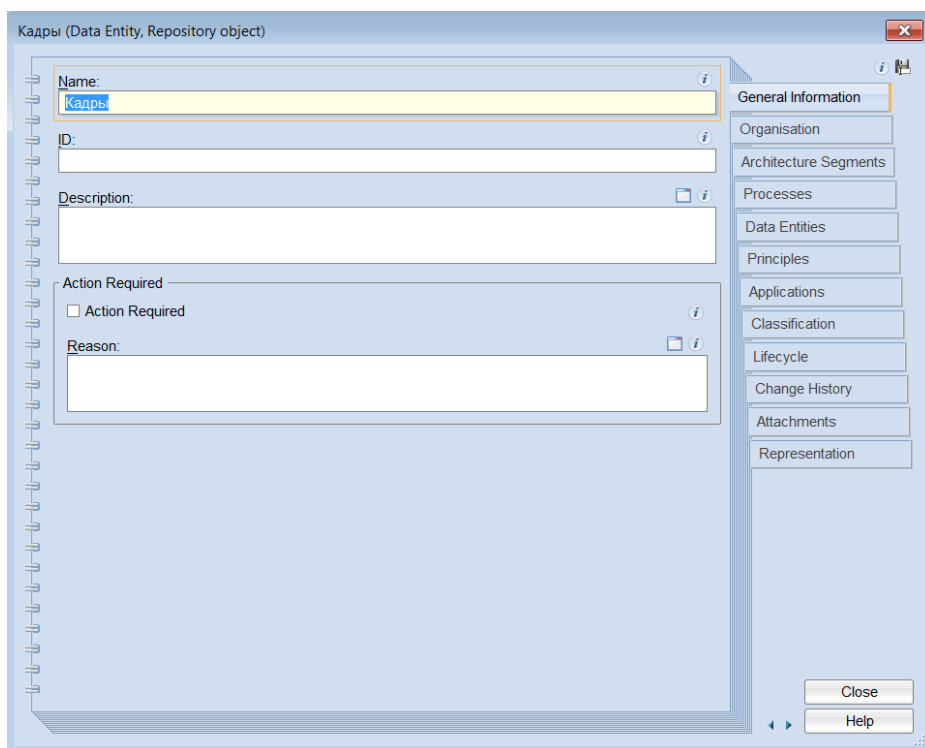
A screenshot of a software window titled 'Кадры (Data Entity, Repository object)'. The window has a light blue background and a sidebar on the right with a list of tabs: 'General Information', 'Organisation', 'Architecture Segments', 'Processes', 'Data Entities', 'Principles', 'Applications', 'Classification', 'Lifecycle', 'Change History', 'Attachments', and 'Representation'. The 'General Information' tab is selected. The main area contains several input fields: 'Name:' with the value 'Кадрь', 'ID:', 'Description:', 'Action Required' (a checkbox), and 'Reason:'. Each field has an information icon (i) to its right. At the bottom right, there are 'Close' and 'Help' buttons.

Рисунок 5.23 - Блокнот для объекта "Данные"

Многие вкладки данного блокнота уже описаны ранее.

Рассмотрим вкладку 'Data Entities'. Эта вкладка задает связи между данными. Вкладка показана на рис. 5.24.

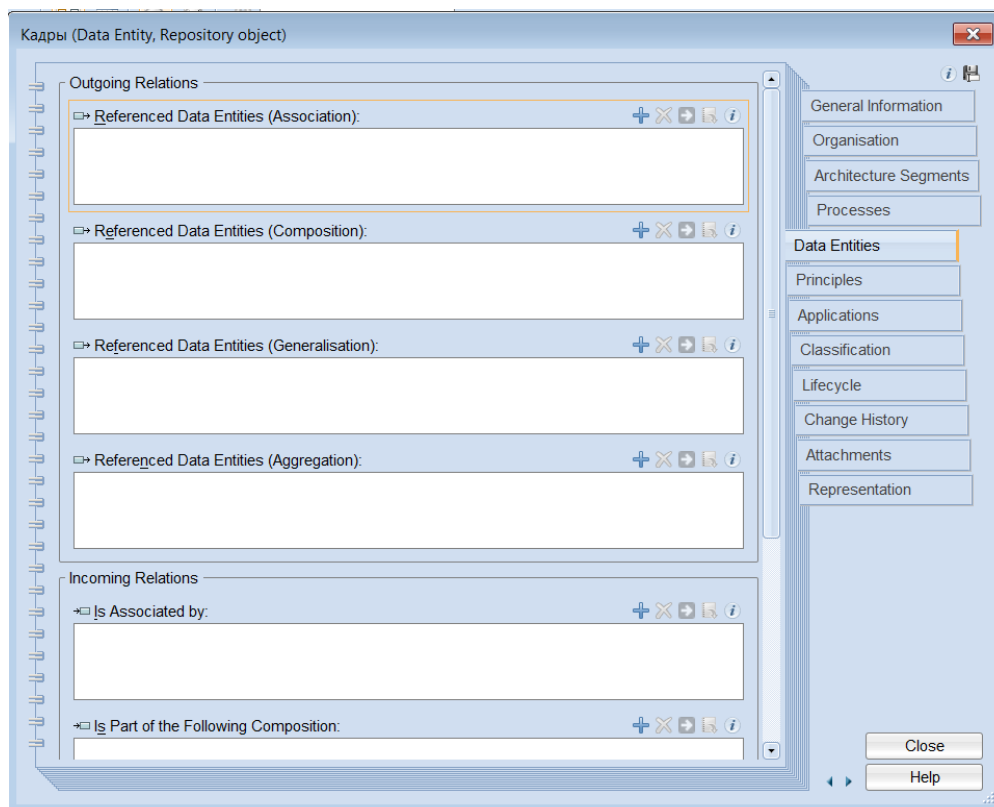


Рисунок 5.24 - Вкладка для задания связей между данными

На вкладке имеются следующие группы полей:

'Outgoing Relations' - исходящие связи;

'Incoming Relations' - входящие связи.

Группа полей 'Outgoing Relations' состоит из следующих полей:

'Referenced Data Entities (Association)' - связи типа ассоциации. На графической модели такие связи отображаются символом . Обозначает смысловое объединение данных. Например, должности и профессии связаны между собой ассоциативной связью.

'Referenced Data Entities (Composition)' - связи типа композиции. На графической модели такие связи отображаются символом . Данные об объектах, которые состоят из других объектов. Связь односторонняя.

'Referenced Data Entities (Generalization)' - связи типа обобщения. На графической модели такие связи отображаются символом . Данные об объектах, которые по смыслу являются обобщением других объектов.

'Referenced Data Entities (Aggregation)' - связи типа агрегирования. На графической модели такие связи отображаются символом . Данные об объектах, которые являются группой других, близких по свойствам объектов.

Группа полей 'Incoming Relations' состоит из следующих двух полей:

'Is Associated by' - связи типа ассоциации, но здесь объект является членом другого ассоциированного объекта.

'Is Part of the Following Composition' - здесь объект входит в данные, представляющие собой композицию.

На рис. 5.25 показан пример семантической модели данных.

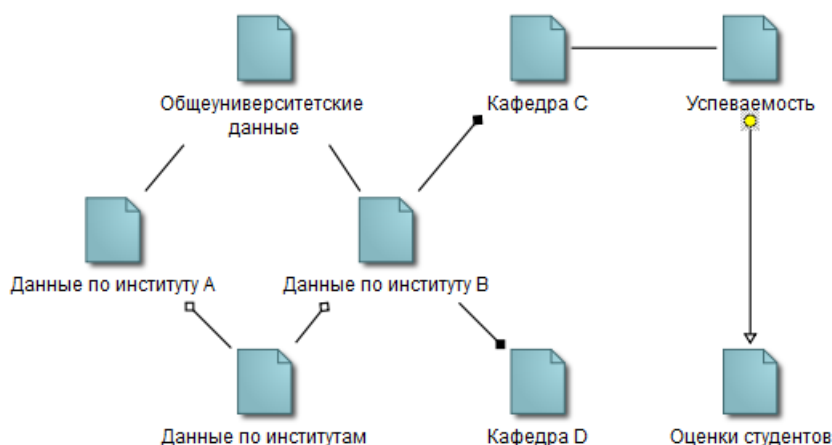


Рисунок 5.25 - Пример семантической модели данных

На рис. 5.26 показана вкладка "Classification" для данных.

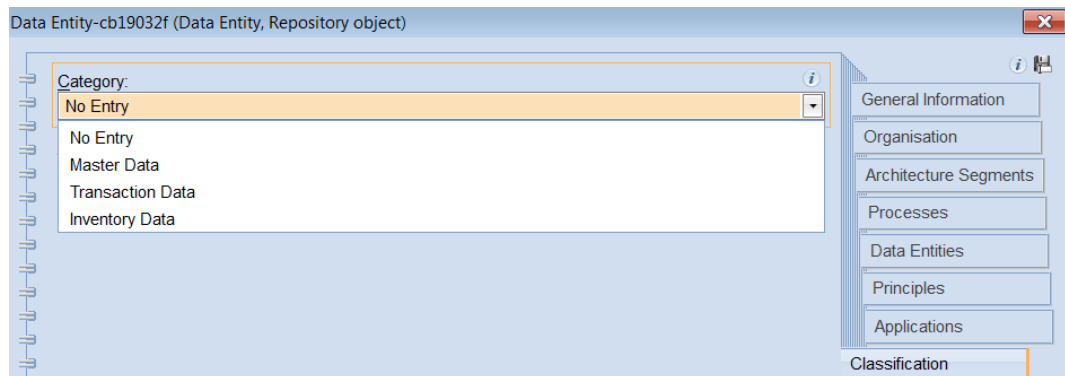


Рисунок 5.26 - Вкладка "Classification" для данных

Имеется возможность указать следующие типы данных:

"Master data" - в этом случае данные играют ключевую роль в операциях бизнеса. Данные могут использоваться несколькими функциями бизнеса, использоваться и храниться в разных информационных системах. Как правило, это информация о продуктах, клиентах, кадрах, партнерах и т.д.

"Transaction data" - в этом случае данные носят динамический характер и являются входами и выходами соответствующих бизнес-процессов.

" Inventory data (or analytical data)" - в этом случае это оцифрованная и структурированная информация, которая используется в системах бизнес-аналитики и системах поддержки принятия решений.

5.4. Связь процессов и данных

Объекты модели типа "Данные" связаны с бизнес-процессами. Так в вкладке "Processes", для объекта "Данные", показанной на рис. 5.27, задаются два типа связи.

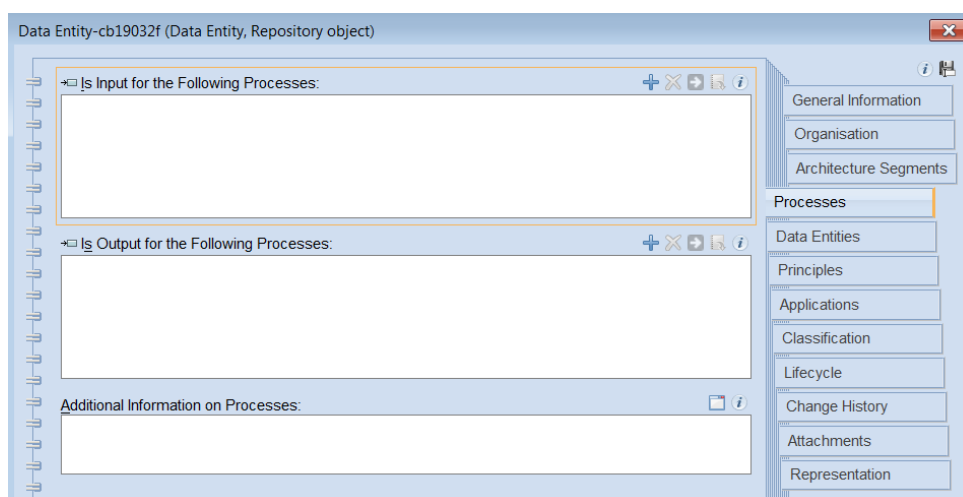


Рисунок 5.27 - Вкладка для связи данных с бизнес-процессами

Графическое представление (Простая линия) для связи типа "Input" - вход, показано на рис. 5.28.

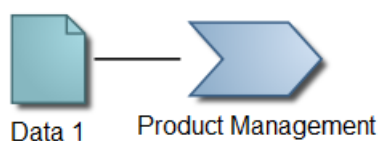


Рисунок 5.28 - Связь типа "Input"

Графическое представление (Простая линия) для связи типа " Output" - выход, показано на рис. 5.29.

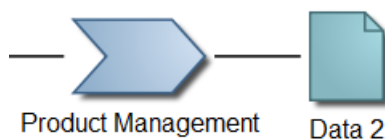


Рисунок 5.29 - Связь типа " Output"

5.5. Матричное представление связей между объектами

В системе есть возможность представлять связи между объектами архитектуры в виде матриц.

На рис. 5.30 показан фрагмент главного меню, содержащий иконку для вызова инструмента работы с матрицами.

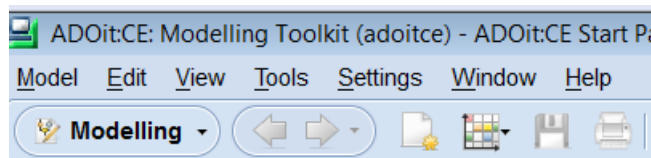


Рисунок 5.30 - Фрагмент главного меню

Если нажать мышкой на иконку в меню быстрых кнопок , то откроется меню, показанное на рис. 5.31, со списком возможных матриц, которые допустимы в системе.

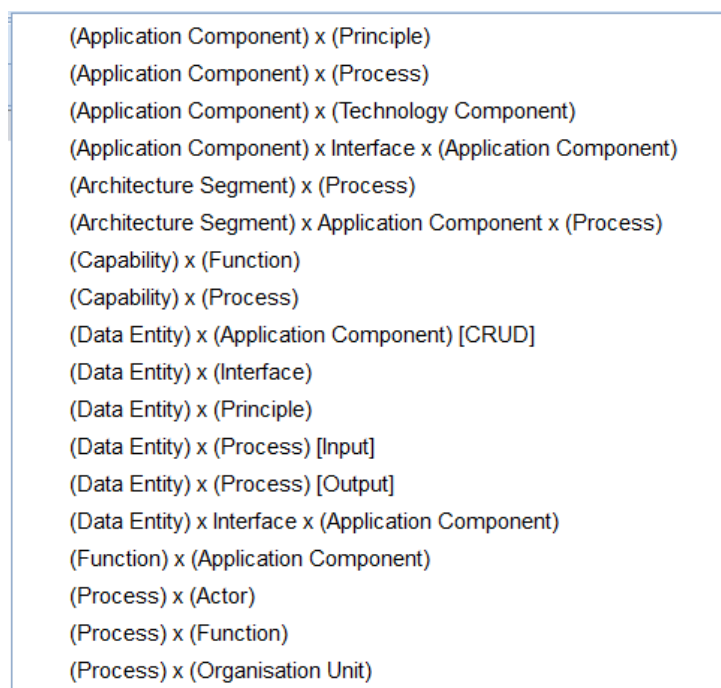


Рисунок 5.31 - Список возможных матриц

Если выбрать соответствующую матрицу, например, (Architecture Segment)x(Process), то будет создана заготовка матрицы, как показано на рис. 5.32.

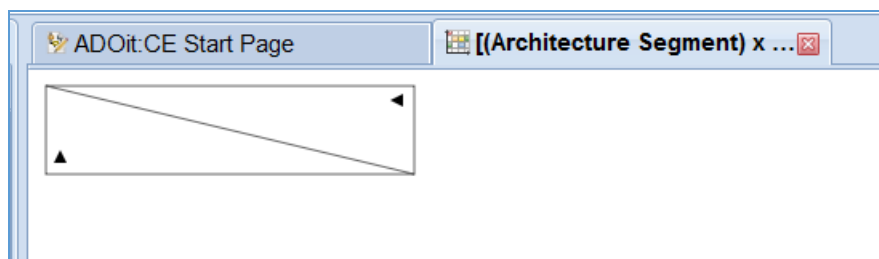


Рисунок 5.32 - Начальная фаза формирования матрицы

Чтобы сформировать строки и столбцы матрицы, надо соответствующие элементы из хранилища "перетащить" с помощью мыши на соответствующие поля. Если объект перетаскивается на поле со стрелочкой вверх, то объект добавляется в столбцы матрицы. Если объект перетаскивается на поле со стрелочкой влево, то объект добавляется в строки матрицы.

Упражнение. Создать три процесса для примера с архитектурой малого предприятия, как показано на рис. 5.33.

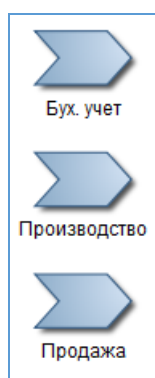


Рисунок 5.33 – Пример бизнес-процессов для упражнения

Установить связи бизнес-процесса «Бух. учет» с функциями, как показано на рис. 5.34.

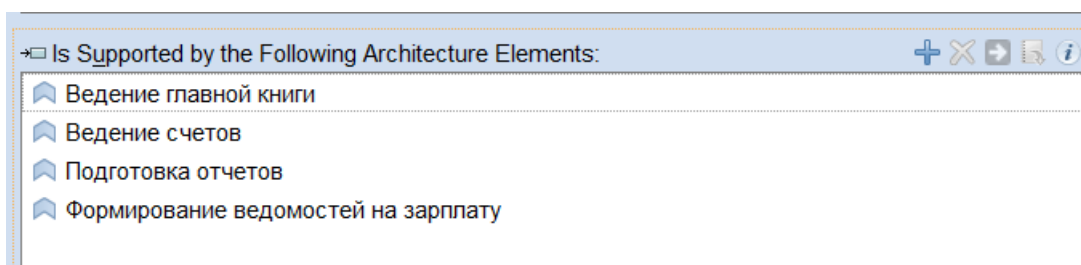


Рисунок 5.34 – Связи бизнес процесса «Бух. учет» с функциями»

Установить связи бизнес-процесса «Бух. учет» с бизнес-единицами, как показано на рис. 5.35.

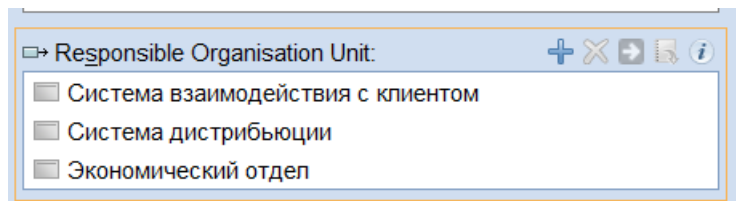


Рисунок 5.35 – Связи бизнес процесса «Бух. учет» с бизнес-единицами

Установить связи бизнес-процесса «Бух. учет» с сегментами бизнеса, как показано на рис. 5.36.

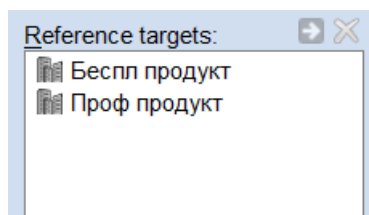


Рисунок 5.36 – Связи бизнес процесса «Бух. учет» с сегментами бизнеса

Установить связи бизнес-процесса «Бух. учет» с принципами, как показано на рис. 5.37.

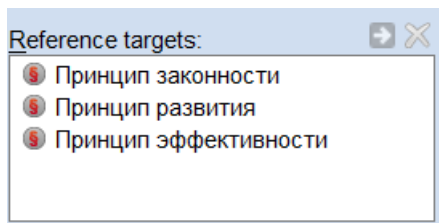


Рисунок 5.37 – Связи бизнес процесса «Бух. учет» с принципами

Установить связи бизнес-процесса «Бух. учет» с возможностями, как показано на рис. 5.38.

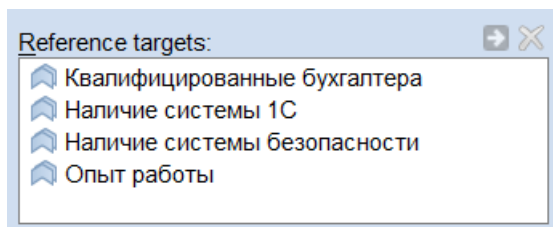


Рисунок 5.38 – Связи бизнес процесса «Бух. учет» с возможностями

Задать параметры бизнес-процесса, как показано на рис. 5.39.

Рисунок 5.39 – Стратегические параметры бизнес-процесса «Бух. учет»

Задать временные параметры для бизнес-процесса, как показано на рис. 5.40.

Рисунок 5.40 – Временные параметры бизнес процесса «Бух. учет»

Установить связи бизнес-процесса «Производство» с функциями, как показано на рис. 5.41.

Рисунок 5.41 – Связи бизнес процесса «Производство» с функциями

Установить связи бизнес-процесса «Производство» с принципами, как показано на рис. 5.42.

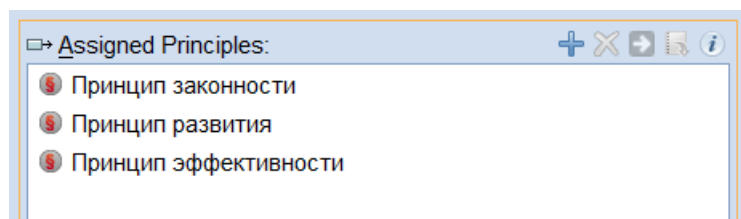


Рисунок 5.42 – Связи бизнес процесса «Производство» с принципами

Установить связи бизнес-процесса «Производство» с возможностями, как показано на рис. 5.43.

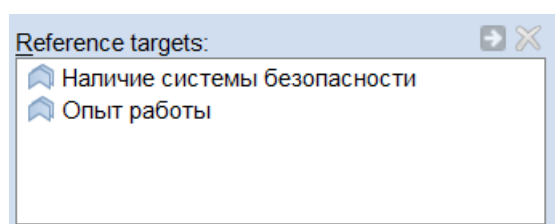


Рисунок 5.43 – Связи бизнес процесса «Производство» с возможностями

Создать матрицу связей между бизнес-единицами и бизнес-процессами. Возможный результат показан на рис. 5.44.

	Бух. учет	Продажа	Производство
Производственный отдел			
Система взаимодействия с клиентом			
Система дистрибьюции			
Экономический отдел			

Рисунок 5.44 – Матрица связей между бизнес-единицами и бизнес-процессами

На рисунке имеющиеся связи показаны заполненными ячейками. Ячейки матрицы будут пустыми, если в блокнотах соответствующих объектов не задать связи.

Чтобы удалить лишний столбец или строку, надо щелкнуть правой кнопкой мыши на имени столбца или строки. При этом откроется меню, показанное на рис. 5.45.

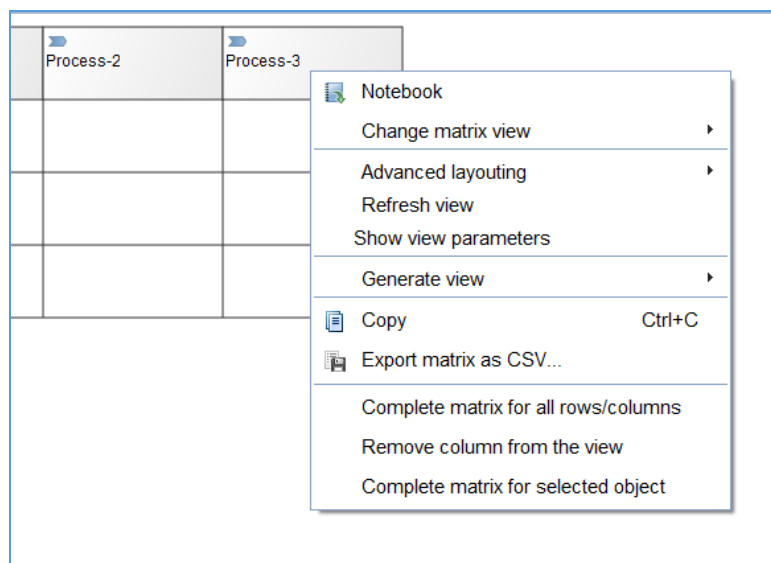


Рисунок 5.45 - Удаление столбца

Для удаления столбца нужно выбрать пункт меню 'Remove column from the view'.

При открытии блокнота для ячейки (дважды щелкнуть левой кнопкой мыши) можно добавить описание связи и временные характеристики связи, как показано на рис. 5.46.

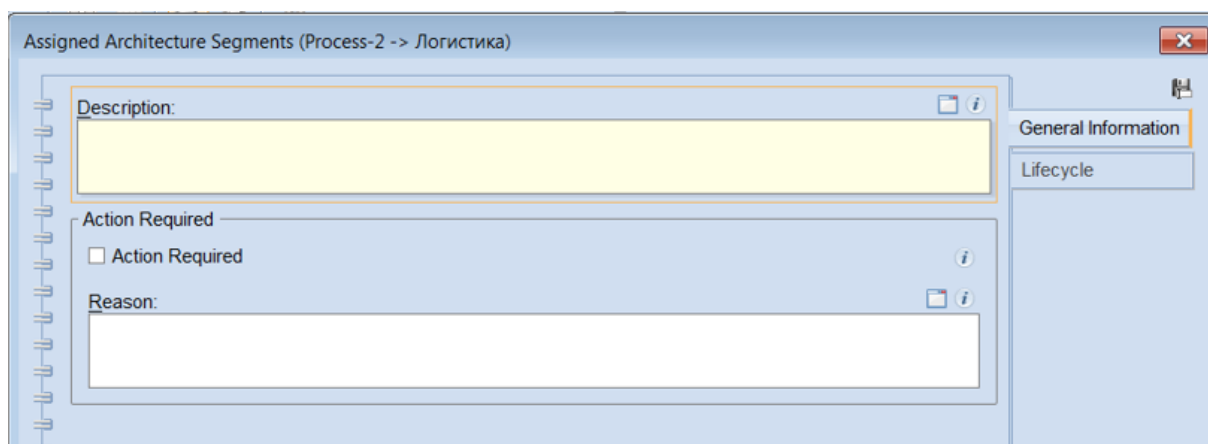


Рисунок 5.46 - Блокнот с описанием связи двух объектов

Если открыть контекстное меню, связанное со столбцом, то можно выбрать другое представление связи между объектами, как показано на рис. 5.47.

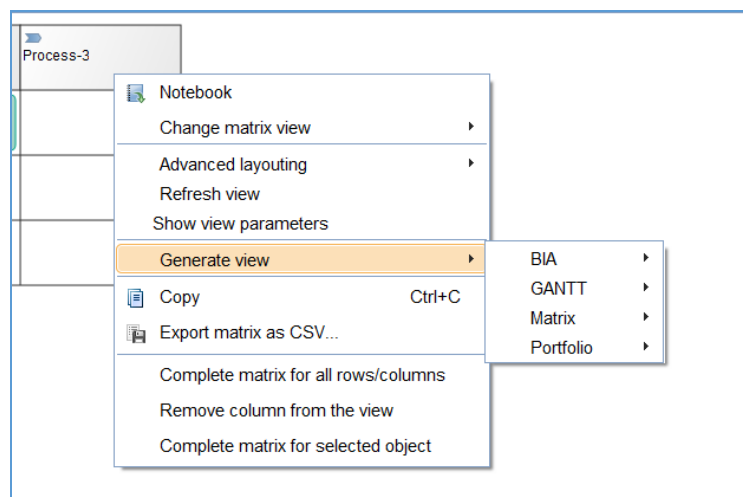


Рисунок 5.47 - Выбор нового представления взаимосвязи

5.6. Заинтересованные лица (Stakeholders)

На архитектуру предприятия значительное влияние оказывают, так называемые «заинтересованные лица». Мы уже рассмотрели две группы таких лиц: клиенты предприятия и работники предприятия. Также к заинтересованным лицам можно отнести следующие категории: поставщики ресурсов предприятия, партнеры по бизнесу, власти и другие группы людей и организации. В системе ADOITCE отсутствуют графические образы для заинтересованных лиц, поэтому для их обозначения будем использовать объект «Бизнес-единица». На рис. 5.48 показан пример модели с заинтересованными лицами.

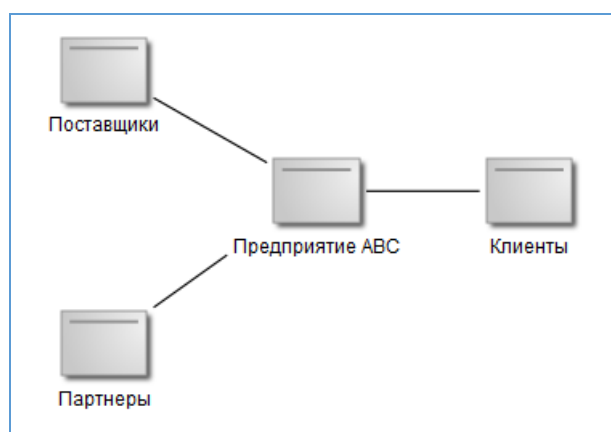


Рисунок 5.48 – Организационная схема с партнерами и поставщиками

5.7. Ресурсы бизнес-процессов

Важной частью архитектуры бизнес-процессов является источник ресурсов для их выполнения. На графической схеме можно указать ресурсы в виде блока, как показано на рис. 5.49.

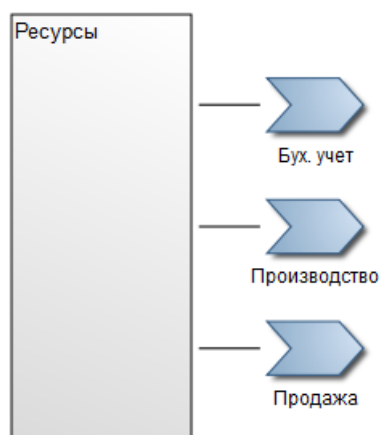


Рисунок 5.49 – Связь ресурсов и бизнес-процессов

В бесплатной версии системы не предусмотрены объекты для описания ресурсов. С этой целью можно использовать объекты типа блока, как показано на рис. 5.49

5.8. Задание по выполнению практикума

- Выполнить упражнение, описанное в методичке.
- Используя бизнес-процессы, приведенные в параграфе 5.12, составить фрагмент карты процессов банка, содержащий процессы согласно вариантам табл.1.
- Соединить полученные процессы с функциями и возможностями, созданными в предыдущем практикуме.
- Построить пузырьковую диаграмму для бизнес-процессов по параметру обеспеченностью ИТ.
- Ввести в модель данные, используемые в процессе 1.1.1.1.1. Связать эти данные с процессом.

5.9. Содержание отчета

Отчет должен содержать следующую информацию: задание на выполнение практикума, копии окон экрана с выполненными упражнениями, копии окон экрана с выполненным индивидуальным заданием, выводы.

5.10. Варианты заданий

Таблица 1 – Варианты бизнес-процессов

Вар.	Процессы	Вар.	Процессы	Вар.	Процессы
1	1.1.1.1.1. 1.2.1.1.1. 3.3.4.1.1.1.	11	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.2.	21	1.1.1.1.3. 1.2.1.1.4. 3.3.4.1.1.3.
2	1.1.1.1.2. 1.2.1.1.1. 3.3.4.1.1.2.	12	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.3.	22	1.1.1.1.3. 1.2.1.1.4. 3.3.4.1.1.4.
3	1.1.1.1.3. 1.2.1.1.1. 3.3.4.1.1.3.	13	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.4.	23	1.1.1.1.3. 1.2.1.1.4. 3.3.4.1.1.5.
4	1.1.1.1.4. 1.2.1.1.1. 3.3.4.1.1.4.	14	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.5.	24	1.1.1.1.3. 1.2.1.1.4. 3.3.4.1.1.6.
5	1.1.1.1.5. 1.2.1.1.1. 3.3.4.1.1.5.	15	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.6.	25	1.1.1.1.3. 1.2.1.1.4. 3.3.4.1.1.7.
6	1.1.1.1.1. 1.2.1.1.2. 3.3.4.1.1.6.	16	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.7.	26	1.1.1.1.3. 1.2.1.1.5. 3.3.4.1.1.1.
7	1.1.1.1.1. 1.2.1.1.3. 3.3.4.1.1.7.	17	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.8.	27	1.1.1.1.3. 1.2.1.1.5. 3.3.4.1.1.2.
8	1.1.1.1.1. 1.2.1.1.4. 3.3.4.1.1.8.	18	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.9.	28	1.1.1.1.2. 1.2.1.1.5. 3.3.4.1.1.1.
9	1.1.1.1.1. 1.2.1.1.5. 3.3.4.1.1.9.	19	1.1.1.1.3. 1.2.1.1.4. 3.3.4.1.1.1.	29	1.1.1.1.5. 1.2.1.1.3. 3.3.4.1.1.1.
10	1.1.1.1.2. 1.2.1.1.3. 3.3.4.1.1.1.	20	1.1.1.1.3. 1.2.1.1.4. 3.3.4.1.1.2.	30	1.1.1.1.5. 1.2.1.1.3. 3.3.4.1.1.3.

5.11. Контрольные вопросы

1. Что такое бизнес-процесс?
2. Что такое архитектура бизнес-процесса?
3. Как бизнес-процесс связан с технологией?
4. Как бизнес-процесс связан с исполнителями?
5. Как бизнес-процесс связан с документами?
6. Кто такие заинтересованные лица?
7. Как построить матрицу связей?
8. Как построить пузырьковую диаграмму?

5.12. Фрагмент процессной модели деятельности банка

1. Основные процессы.
 - 1.1. Обслуживание физических лиц.
 - 1.1.1. Расчетно-кассовое обслуживание физических лиц.
 - 1.1.1.1. Текущие счета физических лиц.
 - 1.1.1.1.1. Открытие текущего счета для физического лица.
 - 1.1.1.1.2. Прием взносов на счет.
 - 1.1.1.1.3. Проведение выплат со счета.
 - 1.1.1.1.4. Взимание комиссии со счета.
 - 1.1.1.1.5. Оформление доверенности на распоряжение счетом.
 - 1.1.1.2. Расчетное обслуживание.
 - 1.1.1.3. Кассовое обслуживание.
 - 1.1.1.4. Валютный контроль и валютные операции.
 - 1.1.1.5. Денежные переводы по системам.
 - 1.1.2. Вклады.
 - 1.1.3. Кредитование физических лиц.
 - 1.1.4. Банковские карты для физических лиц.
 - 1.1.5. Индивидуальные банковские ячейки (сейфы) для физических лиц.
 - 1.2. Обслуживание юридических лиц.
 - 1.2.1. Расчетно-кассовое обслуживание юридических лиц.
 - 1.2.1.1. Текущие счета юридических лиц.
 - 1.2.1.1.1. Открытие текущего счета для юридического лица.
 - 1.2.1.1.2. Прием средств на счет.
 - 1.2.1.1.3. Проведение выплат со счета.
 - 1.2.1.1.4. Взимание комиссии со счета.
 - 1.2.1.1.5. Оформление доверенности на распоряжение счетом.
 - 1.2.1.2. Расчетное обслуживание.
 - 1.2.1.3. Кассовое обслуживание.
 - 1.2.1.4. Валютный контроль и валютные операции.
 - 1.2.1.5. Денежные переводы по системам.
 - 1.3. Работа на финансовых и межбанковских рынках.
 2. Вспомогательные процессы.
 3. Управленческие процессы
 - 3.1. Управление финансами.
 - 3.2. Управление персоналом.
 - 3.3. Управление маркетингом
 - 3.3.1. Маркетинговые исследования
 - 3.3.2. Планирование ценообразования
 - 3.3.3. Реклама
 - 3.3.4. Управление продуктами.
 - 3.3.4.1. Разработка и внедрение продуктов и услуг.

- 3.3.4.1.1. Разработка новых и изменение текущих продуктов.
- 3.3.4.1.1.1. Управление разработкой новых и изменением текущих продуктов.
- 3.3.4.1.1.2. Поиск и отбор идей по новым продуктам.
- 3.3.4.1.1.3. Управление портфелем проектов по новым продуктам/ изменениям текущих продуктов.
- 3.3.4.1.1.4. Управление проектом по разработке нового продукта/изменению текущего продукта.
- 3.3.4.1.1.5. Выполнение исследований по новому продукту/изменениям текущего продукта.
- 3.3.4.1.1.6. Проработка идеи нового продукта/изменения текущего продукта.
- 3.3.4.1.1.7. Разработка нового продукта/измененного текущего продукта.
- 3.3.4.1.1.8. Подготовка запуска продукта.
- 3.3.4.1.1.9. Запуск продукта.

Глава 6. Лабораторный практикум «Моделирование архитектуры информационной системы»

6.1. Состав архитектуры информационной системы

Вторая основная часть архитектуры предприятия - это архитектура информационной системы. Архитектура информационной системы состоит из архитектуры приложений, архитектуры данных и архитектуры технологических компонентов.

Приложения – это программы, которые позволяют пользователю обрабатывать информацию и работать в компьютерных сетях, не владея программированием. Приложения делятся на приложения общего назначения и специальные приложения.

К приложениям общего назначения можно отнести следующие приложения:

- Электронные калькуляторы для обработки числовой информации.
- Текстовые редакторы для создания и редактирования текстовых документов.
- Электронные таблицы для хранения данных в табличной форме и работы с этими данными.
- Графические редакторы для создания и обработки графических изображений.
- Программы разработки презентаций.
- Звуковые и мультимедиа редакторы и многие другие.

К приложениям специального назначения можно отнести следующие приложения:

- Системы автоматизированного проектирования.
- Бухгалтерские программы.
- Системы управления бизнес-процессами.
- Системы моделирования.
- Система планирования ресурсов 1С.
- Система моделирования бизнес-процессов ADONIS.
- Система электронного банка.
- Система бизнес аналитики Pentaho.
- Система управления проектами MS Project.
- Система стратегического управления ADOscore.
- Системы автоматизированного управления предприятием и многие другие.

Технологические компоненты информационной системы предприятия – это программные и аппаратные средства, обеспечивающие работу системы.

К таким компонентам можно отнести следующие программные продукты:

- Операционные системы.
- Системы управления базами данных.
- Системы обеспечения коммуникаций.
- Антивирусы и многие другие.

6.2. Создание модели архитектуры информационной системы

Модель архитектуры информационной системы может храниться в одной из первоначально созданных папок или для нее можно создать новую папку, как показано на рис. 6.1.

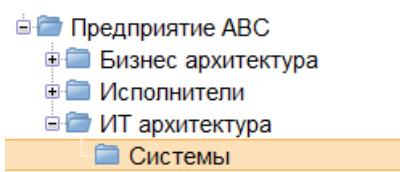


Рисунок 6.1 - Создание папки для информации об информационной системе

При выборе типа новой модели, как это уже рассматривалось в предыдущих практикумах, будет открыт контекстный список видов моделей, как показано на рис. 6.2.

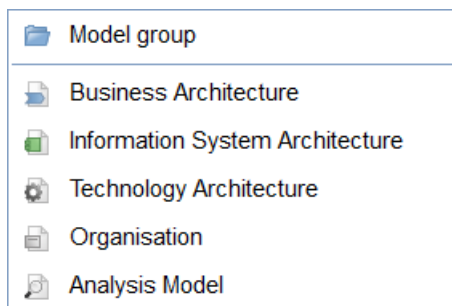


Рисунок 6.2 – Список с видами моделей

При выборе модели “Information System Architecture” (Архитектура информационной системы) открывается новое рабочее поле с палитрой объектов, показанной на рис. 6.3.

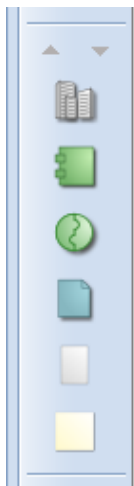


Рисунок 6.3 – Палитра объектов для модели информационной системы

6.3. Объект «Приложение» (Application Component)

Для представления в модели прикладной компьютерной системы, которая решает бизнес задачи предприятия служит объект "Приложение".

В качестве примеров программных приложений можно привести следующие системы:

- Система планирования ресурсов 1С.
- Система моделирования бизнес-процессов ADONIS.
- Система электронного банка.
- Система бизнес аналитики Pentaho.
- Система управления проектами MS Project.
- Система стратегического управления ADOscore.

Графическое изображение объекта "Приложение, показано на рис. 6.4.



Приложение

Рисунок 6.4 – Графическое изображение объекта "Приложение"

Блокнот для приложения с именем «ADOit» показан на рис. 6.5.

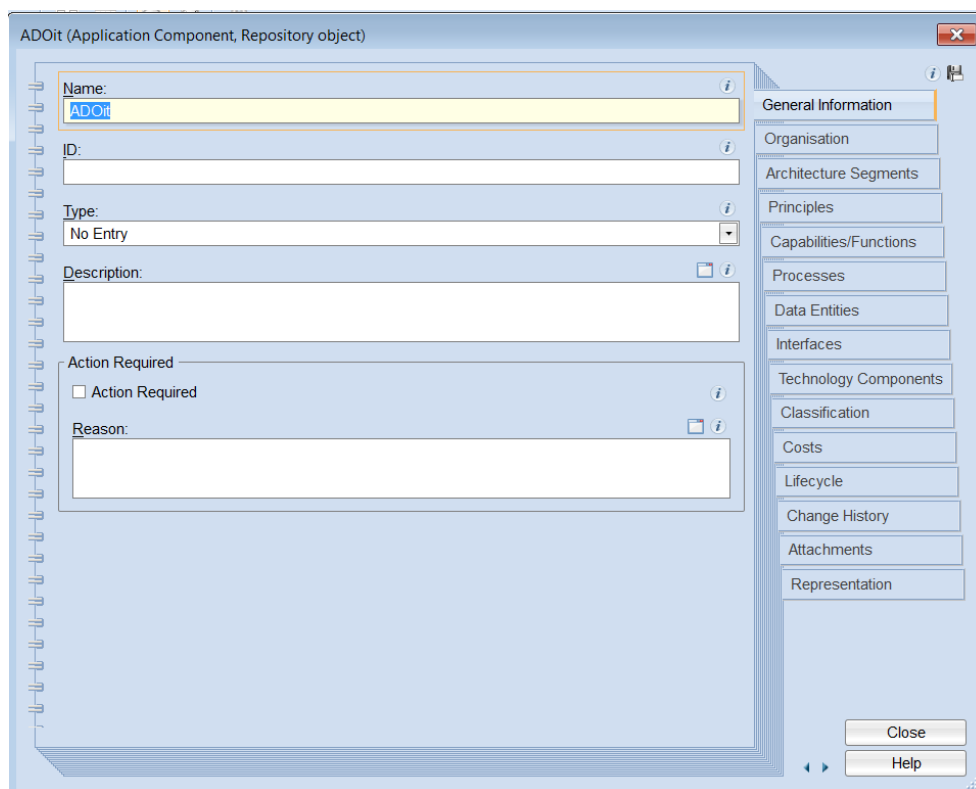


Рисунок 6.5 - Блокнот для элемента "Приложение"

Надо отметить, что блокнот для приложения имеет наиболее развитую систему вкладок. На вкладке общей информации появилось новое поле “Type” (Тип приложения). При раскрытии контекстного списка открывается список возможных типов приложений, как показано на рис. 5.6.

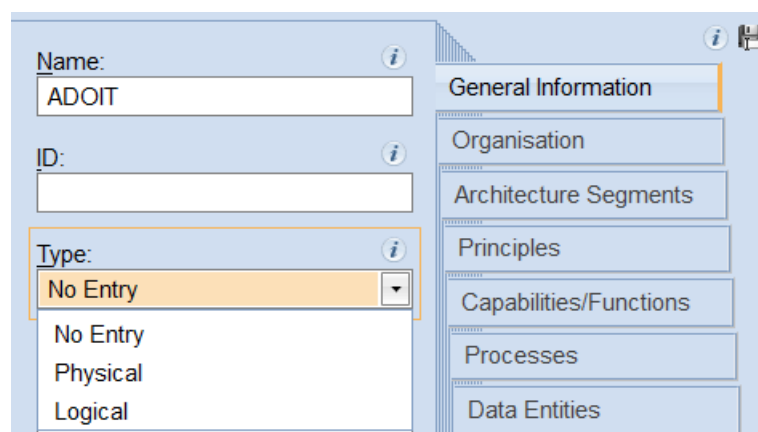


Рисунок 5.6 – Типы приложений

Это поле может принимать следующие значения:
 “No Entry” - тип приложения не определен;

“Physical” - система для своего выполнения имеет специфические физические (материальные) элементы. Например, сеть терминалов самообслуживания банка, терминалы отображения показателей предприятия в ситуационной комнате и другие;

“Logical” - приложение носит логический характер, то есть реализовано как виртуальная часть в какой-то программной системе или может использовать другие стандартные программные системы. Например, операционные системы, системы виртуализации и т.д.

Вкладка “Organization” имеет стандартные поля для связи с элементами организационной модели, которые были рассмотрены в предыдущих практикумах.

На вкладках “Architecture Segments” и “Principles” задаются связи приложения с сегментами бизнеса и принципами.

На вкладках “Capabilities/Functions” и “Processes” задаются возможности, функции и бизнес-процессы, которые обеспечивает данное приложение.

Вкладка “Data Entities” задает связь с архитектурой данных. Более подробно эта вкладка будет рассмотрена ниже. Вкладки “Interfaces” и “Technology Components” будут рассмотрены после изучения соответствующих архитектурных объектов.

На рис. 6.7 показана вкладка для задания параметров приложения (Classification).

1С Предприятие (Application Component, Repository object)

Category: Standard Software

Strategic Importance: High

Business Fit: Sufficient

IT Fit: Moderate

Cost Fit: Good

Number of Users: 20

☐ External Application Component

Security

Availability: Standard

Confidentiality: High

Integrity: Standard

Additional Information on Classification:

General Information

Organisation

Architecture Segments

Principles

Capabilities/Functions

Processes

Data Entities

Interfaces

Technology Components

Classification

Costs

Lifecycle

Change History

Attachments

Representation

Close

Help

Рисунок 6.7 - Вкладка для параметров приложения (Classification)

Все поля имеют раскрывающиеся списки стандартных значений.

Поле “Category” задает категорию/тип программного продукта. На рис. 6.8 показаны возможные значения этого параметра.

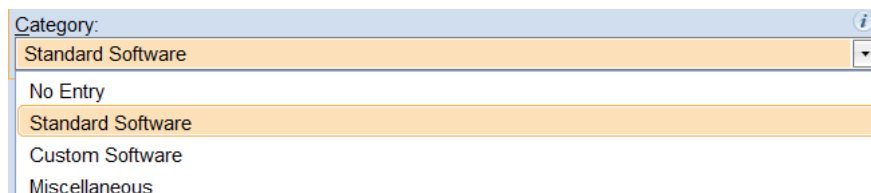


Рисунок 6.8 - Категории программного продукта

Параметр может принимать следующие значения:

“No Entry” - категория приложения не определена;

“Standard Software” - стандартное программное обеспечение;

“Custom Software” - собственное программное обеспечение;

“Miscellaneous” - смешанное программное обеспечение.

Поле “Strategic Importance” задает степень важности для стратегии предприятия. На рис. 6.9 показаны возможные значения этого параметра.

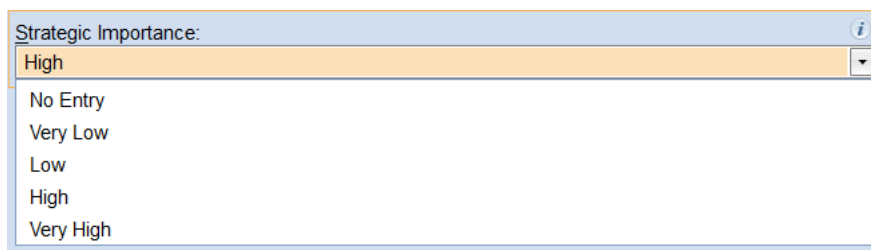


Рисунок 6.9 - Уровни стратегической важности

Параметр может принимать следующие значения:

“No Entry” - уровень важности не определен;

“Very Low” - очень низкий уровень;

“Low” – низкий уровень;

“High” – высокий уровень;

“Very High” - очень высокий уровень.

Поле “Business Fit” задает степень соответствия приложения задачам бизнеса. На рис. 6.10 показаны возможные значения этого параметра.

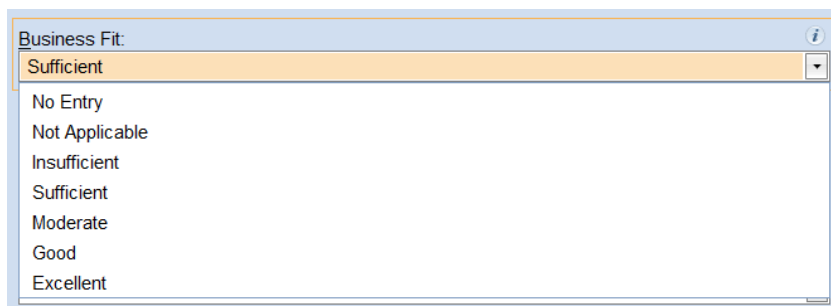


Рисунок 6.10 - Уровни соответствия приложения для бизнеса

Параметр может принимать следующие значения:

- “No Entry” - уровень не определен;
- “Not Applicable” – приложение не применяется;
- “Insufficient” – неудовлетворительный уровень;
- “Sufficient” – удовлетворительный уровень;
- “Moderate” – средний уровень;
- “Good” – хороший уровень;
- “Excellent” – отличный уровень.

Поле “Number of Users” задает количество пользователей приложения.

Маркер “External Application Component” используется для обозначения, что приложение является внешним.

Группа полей “Security” задает параметры приложения, связанные с безопасностью.

Поле “Availability” задает степень доступности приложения. На рис. 6.11 показаны возможные значения этого параметра.

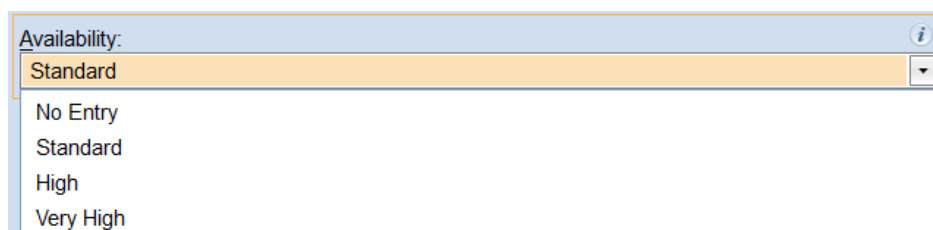


Рисунок 6.11 - Уровни доступности приложения

Параметр может принимать следующие значения:

- “No Entry” - уровень не определен;
- “Standard” – стандартный уровень;
- “High” – высокий уровень;
- “Very High” - очень высокий уровень.

Поле “Confidentiality” задает уровень конфиденциальности приложения. Значения уровней параметра аналогичны уровням предыдущего параметра.

Поле “Integrity” задает степень интегрированности приложения. Значения уровней также аналогичны значениям уровней двух предыдущих параметров.

Вкладка “Costs”, показанная на рис. 6.12 задает стоимостные параметры приложения.

1С Предприятие (Application Component, Repository object)

Investment Costs: 100000

Operating Costs: 10000

Additional Information on Costs:

General Information
Organisation
Architecture Segments
Principles
Capabilities/Functions
Processes
Data Entities
Interfaces
Technology Components
Classification
Costs
Lifecycle
Change History
Attachments
Representation

Close
Help

Рисунок 6.12 - Вкладка со стоимостными параметрами приложения

Вкладка имеет два следующих дополнительных поля:

“Investment Costs” - стоимость приобретения;

“Operating Costs” - стоимость владения.

Оба поля имеют численные значения. Значения стоимости задается в условных единицах.

На рис. 6.13 показана вкладка, задающая жизненный цикл приложения.

“In Development” – приложение в процессе проектирования;

“In Production” – приложение в использовании;

“Retired” – приложение выведено из эксплуатации.

Поле “Production Date” задает конкретную дату внедрения приложения, то есть ту дату, когда приложение начнет использоваться на предприятии для решения текущих задач бизнеса.

Поле “Decommission Date” задает дату, когда приложение планируется или уже выведено из оперативного использования.

Поле “Investment Strategy” задает тип стратегии финансирования создания приложения. Поле принимает значения, показанные на рис. 6.15.

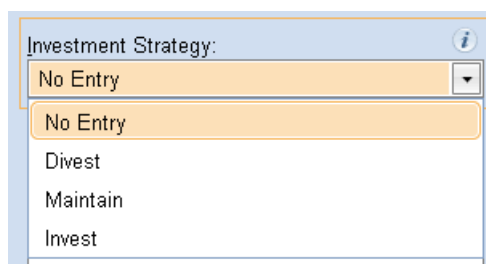


Рисунок 6.15 – Инвестиционные стратегии

Поле может принимать следующие значения:

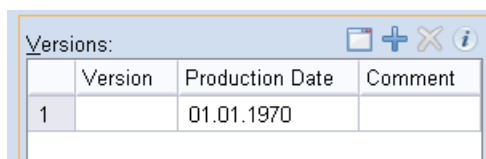
“No Entry” - уровень не определен;

“Divest” – продать приложение, прекратить финансирование;

“Maintain” – финансировать поддержку приложения в рабочем состоянии;

“Invest” – вкладывать дополнительные средства в поддержку или развитие приложения.

Поле “Version”, показанное на рис. 6.16 содержит таблицу истории версий приложения.



	Version	Production Date	Comment
1		01.01.1970	

Рисунок 6.16 – Таблица с историей версий приложения

Окно “Replaced Application Components” содержит список компонентов приложения, которые были заменены, заменяются или будут заменены в данном приложении. На рис. 6.17 показаны приложения в хранилище для выбора приложения, которое было заменено в данном приложении. В качестве такого приложения было выбрано приложение с именем “New”.

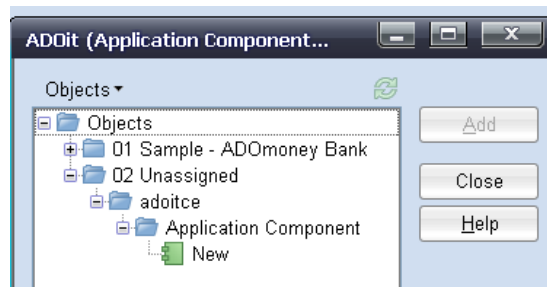


Рисунок 6.17 – Выбор замененного приложения для вставки в блокнот

В результате в окне будет отображена иконка замененного компонента приложения, как показано на рис. 6.18.

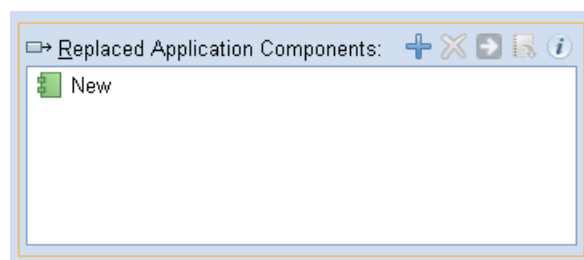


Рисунок 6.18 – Список замененных компонентов приложения

Окно “Replaced by” аналогично предыдущему окну, но в нем отражаются компоненты, на которые заменены, заменяются или будут заменены компоненты данного приложения.

Упражнение 1. Составить показатели жизненного цикла приложения ADOit, как показано на рис. 6.19.

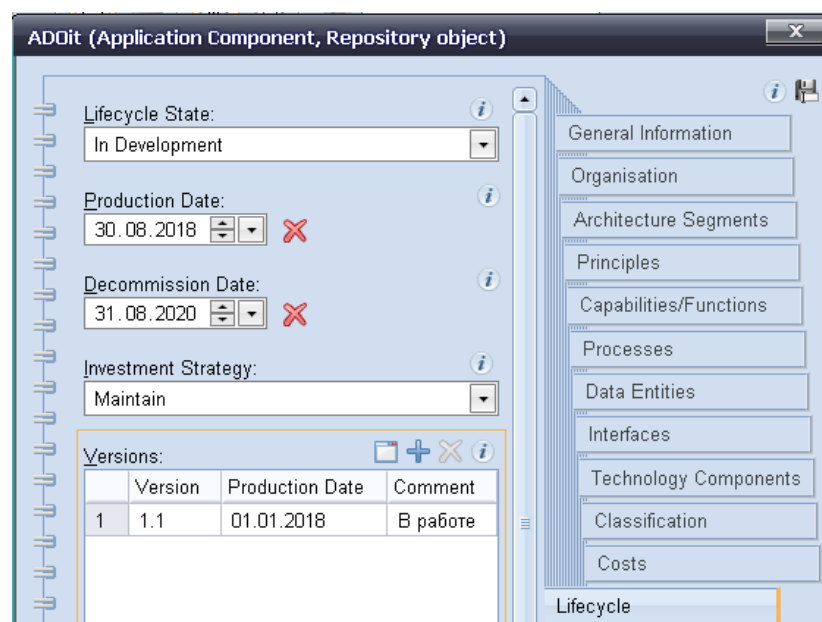


Рисунок 6.19 – Параметры приложения для упражнения

Остальные вкладки носят стандартный характер и были рассмотрены в предыдущих практиках.

6.4. Построение пузырьковых диаграмм для приложений

В системе ADOit имеется возможность построения определенных графических средств для анализа архитектуры. Одним из этих средств является пузырьковая диаграмма.

Как рассматривалось ранее для того, чтобы построить пузырьковую диаграмму, необходимо удерживая клавишу Ctrl на клавиатуре отметить нужные объекты типа "Приложение". Затем выбрать пункт меню 'Generate view' (Создать вид), как показано на рис. 6.20.

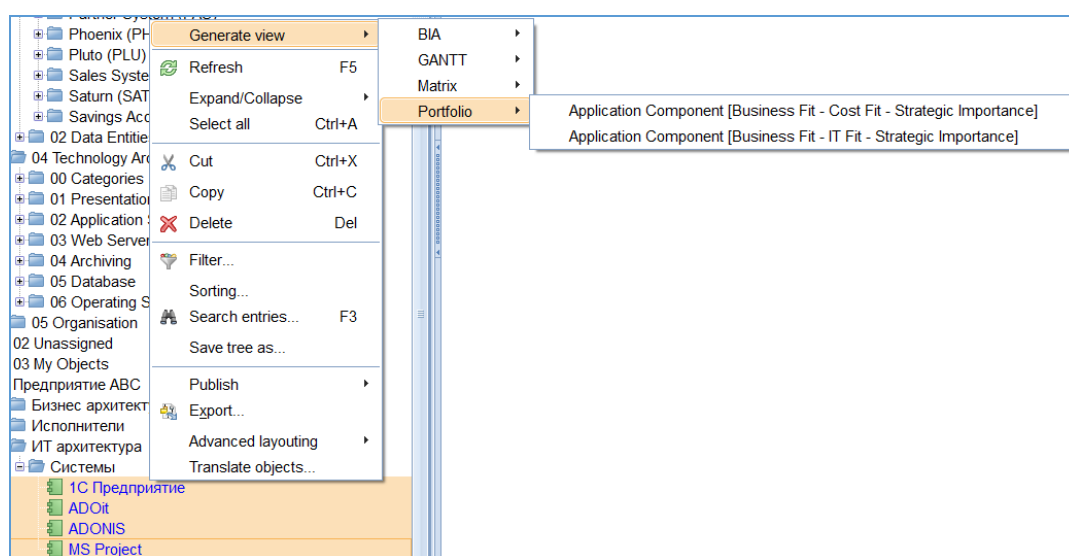


Рисунок 6.20 - Выбор вида пузырьковой диаграммы

В отличие от рассматриваемого ранее вида пузырьковой диаграммы для приложения строится трехпараметрическая диаграмма. На рис. 6.20 показано, что предлагается построить два варианта диаграммы. В первом варианте по осям координат показываются уровни двух параметров 'Business Fit' и 'Cost Fit', а во втором 'Business Fit' и 'IT Fit'. В обоих случаях параметр 'Strategic Importance' представляется в виде пузырька соответствующего размера.

На рис. 6.21 показан пример построения пузырьковой диаграммы первого типа для четырех приложений.

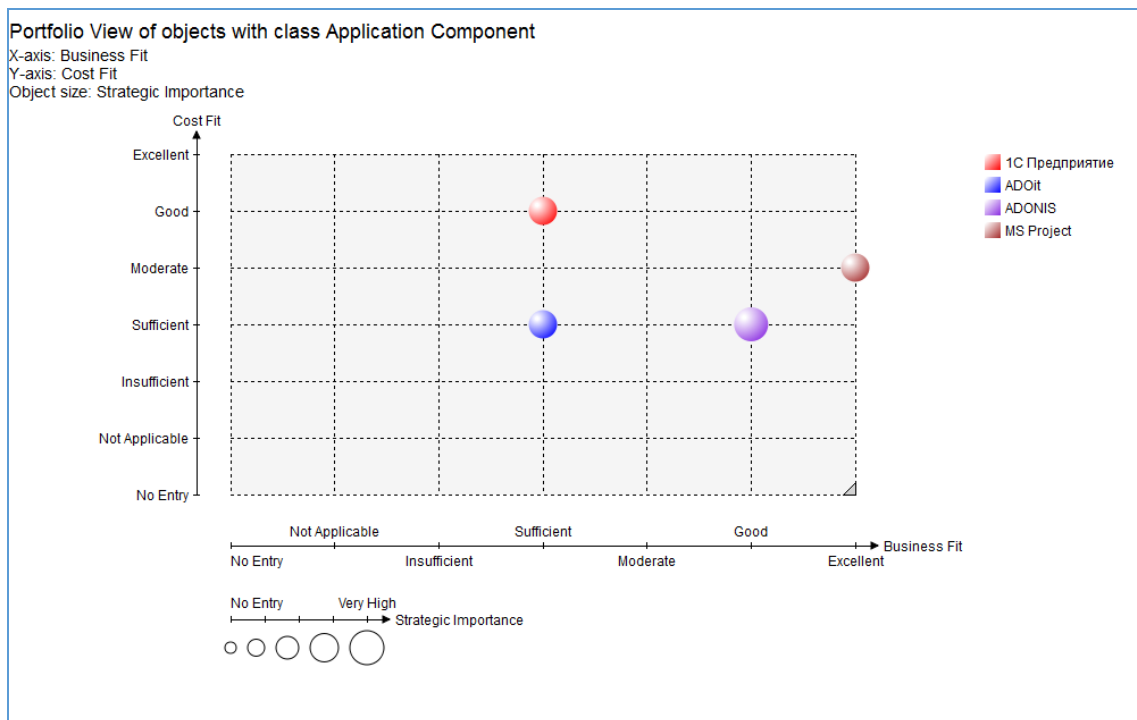


Рисунок 6.21 - Диаграмма с осями для параметров 'Business Fit' и 'Cost Fit'

На рис. 6.22 показан пример построения пузырьковой диаграммы второго типа для тех же четырех приложений.

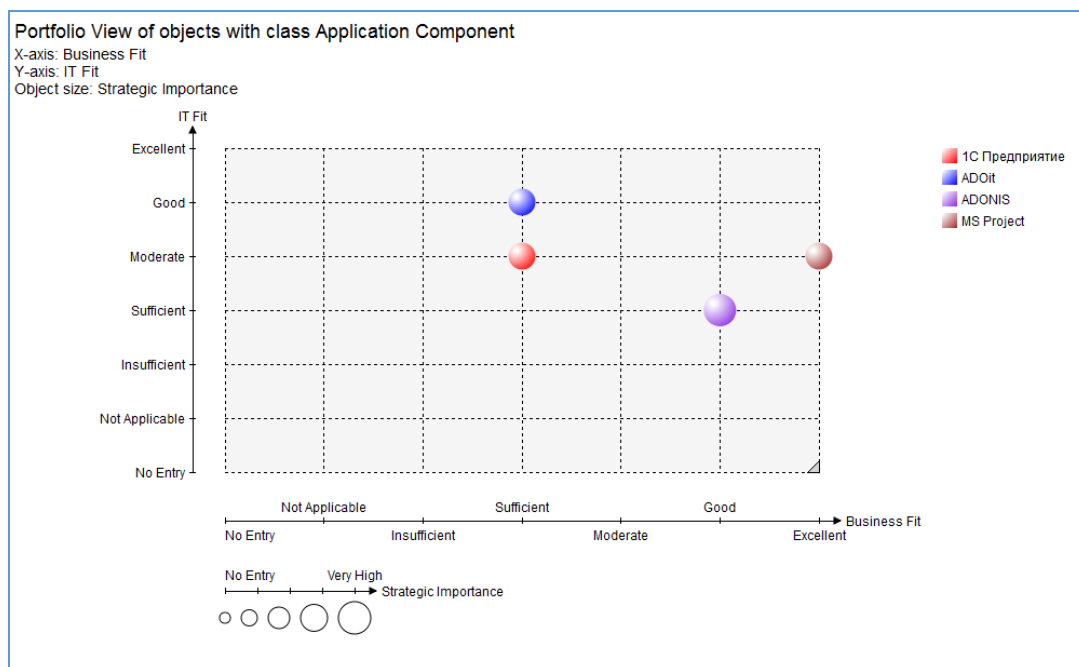


Рисунок 6.22 - Диаграмма с осями для параметров 'Business Fit' и 'IT Fit'

Упражнение 2. Требуется создать четыре объекта типа приложения, как показано на рис. 6.21: MS Project, 1C Предприятие, ADONIS и ADOit.

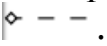
Самостоятельно для каждого объекта заполнить вкладку 'Classification'.
Получить два типа пузырьковой диаграммы.

6.5. Связи между приложениями и данными

На графической модели передача информации от приложения к приложению показывается стрелкой . На рис. 6.23 показан фрагмент модели информационной системы с указанием потока передачи информации.



Рисунок 6.23 - Поток передачи информации

На графической схеме можно также указать связь между заменяемыми приложениями с помощью пунктирной линии с пустым ромбиком .

Для задания связи данных с приложением как было уже сказано ранее проще воспользоваться контекстным меню и пунктом “New relation”, как показано на рис. 6.24.

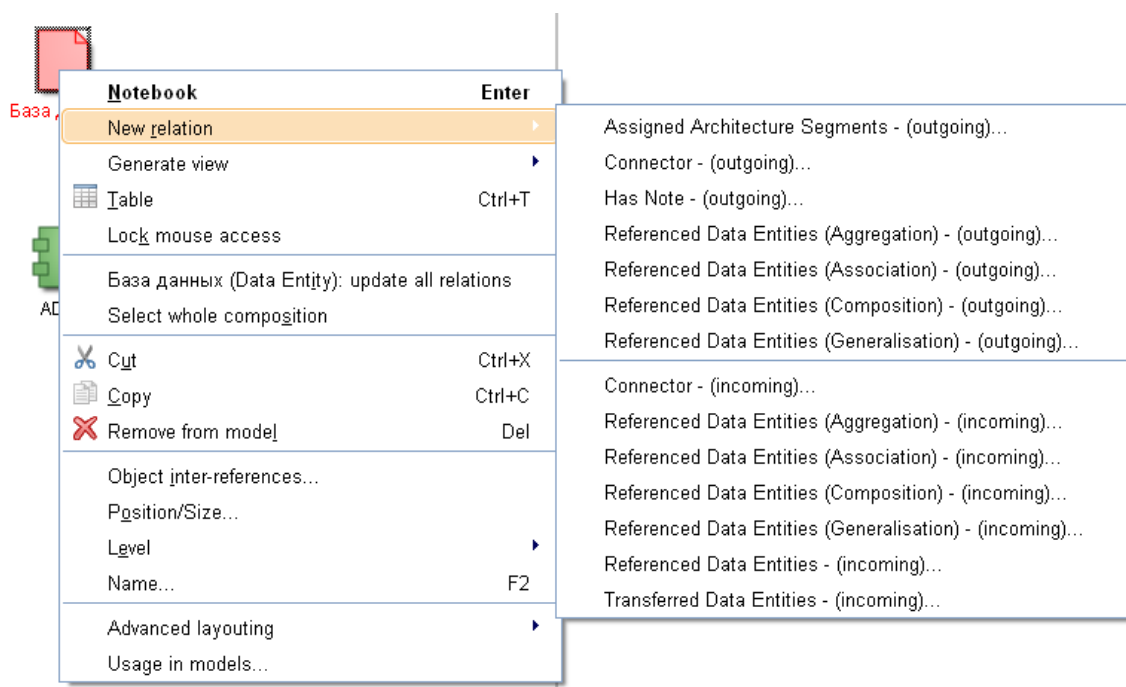


Рисунок 6.24 – Меню для выбора связи для объекта типа «Данные»

В свою очередь связи можно задавать и для объекта «Приложение», как показано на рис. 6.25.

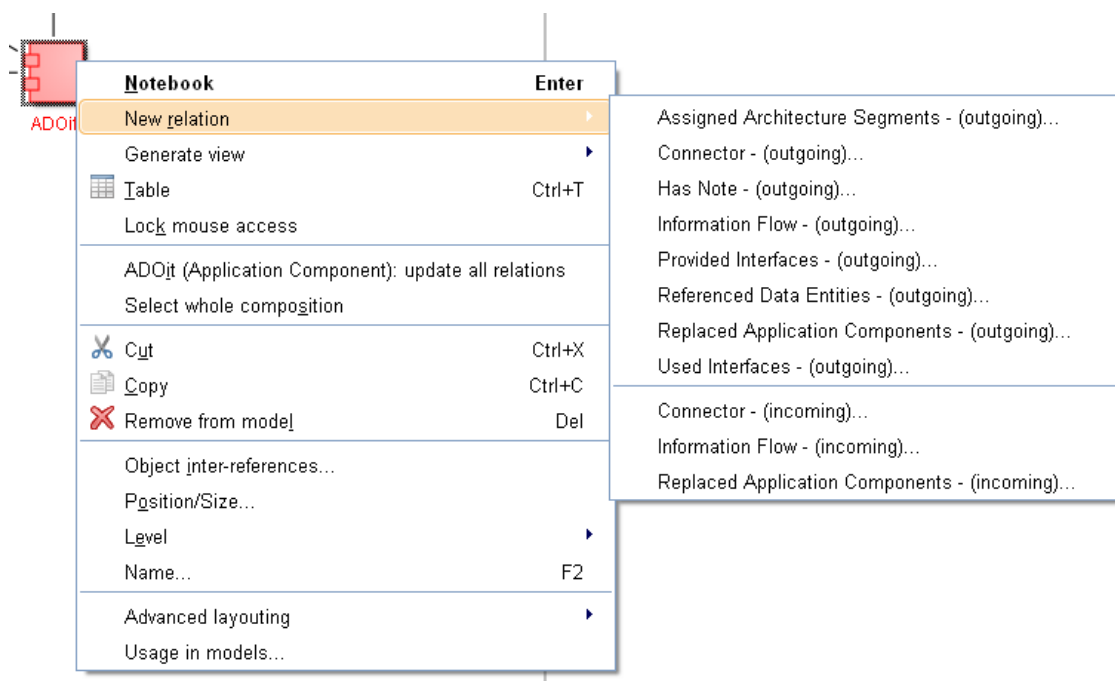


Рисунок 6.25 - Меню для выбора связи для объекта типа «Приложение»

На рис. 6.26 показана простейшая модель информационной системы с одним приложением и тремя хранилищами данных.

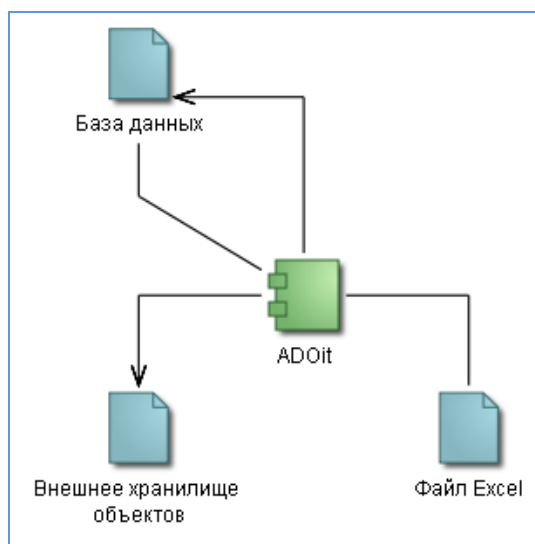


Рисунок 6.26 – Модель архитектуры информационной системы с данными

К сожалению, бесплатная версия системы имеет ряд недостатков. В частности, не все виды связей имеют специальное обозначение. В этом случае рекомендуется использовать универсальную связь, как показано на рис. 6.26. Так исходящая связь со стрелкой “Referenced Data Entities – (out-

going)» для приложения не имеет обратной входящей связи от базы данных и приходится использовать универсальный коннектор.

На рис. 6.27 показана ссылка на связанное приложение в блокноте для данных.

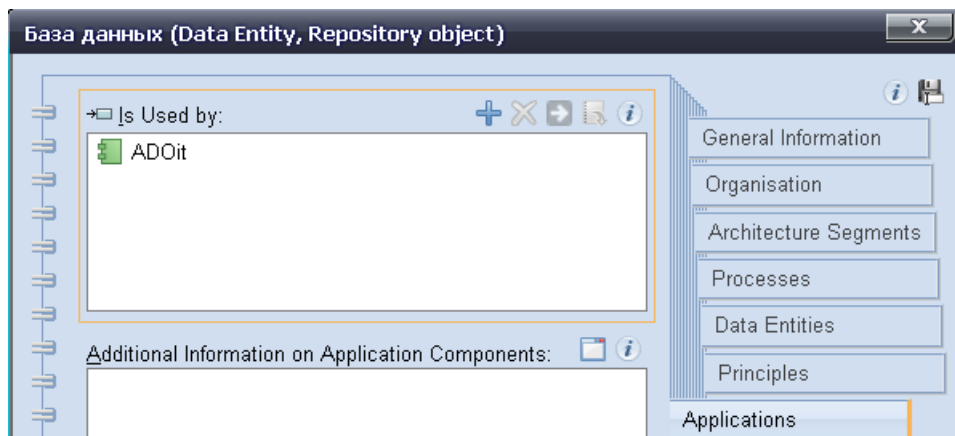


Рисунок 6.27 – Отражение связей с приложениями для объекта «Данные»

На рис. 6.28 показана созданная автоматически связь с данными в блокноте для приложения.

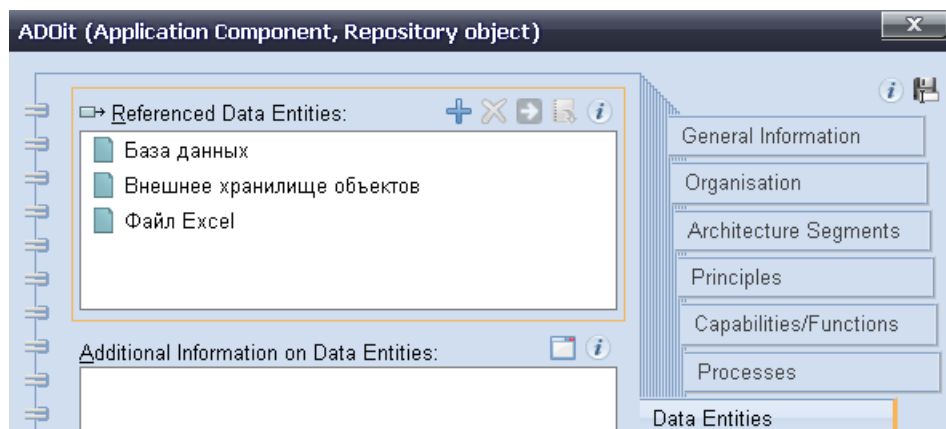


Рисунок 6.28 - Отражение связей с данными для объекта «Приложение»

Для данных существует целое множество видов связей, которые обычно рассматриваются в разделе архитектуры данных.

Упражнение 3. Требуется соединить введенные приложения в предыдущем упражнении связями типа «Информационный поток». Добавить, как минимум произвольно три типа данных. Соединить данные с приложениями.

6.6. Матричные взгляды на приложение

Как в предыдущем практикуме для выбора вида матрицы надо воспользоваться последовательностью меню “Generate view” и “Matrix”, как показано на рис. 6.29.

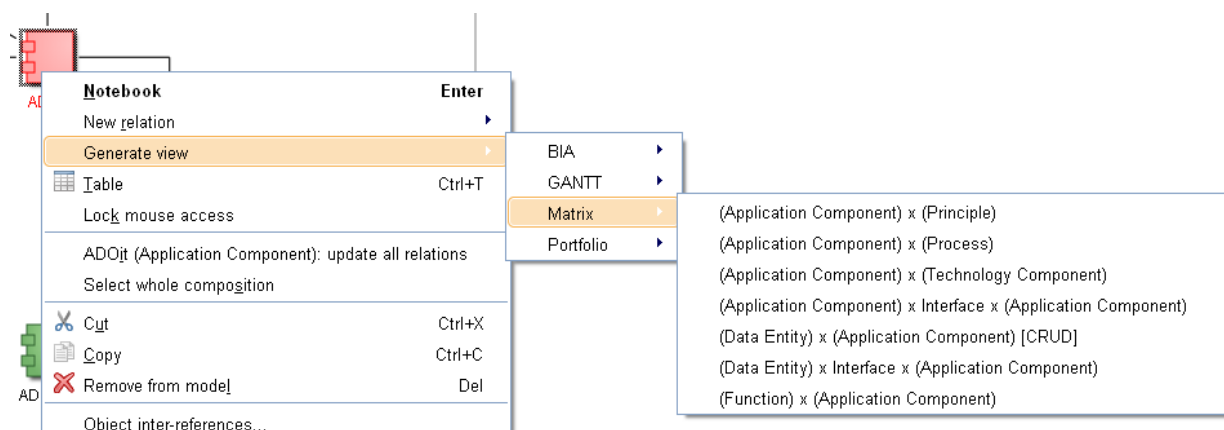


Рисунок 6.29 – Последовательность меню для выбора типа матрицы

Новым инструментом здесь является матрица типа “(Data Entity)x(Application Component)[CRUD]”. Вид матрицы для одного приложения ADOit и трех мест хранения данных, показан на рис. 6.30.



Рисунок 6.30 – Вид матрицы в режиме CRUD

Для такого представления матрицы надо щелкнуть правой кнопкой мыши на любом месте матрицы. При этом откроется контекстное меню, показанное на рис. 6.31. Аббревиатура CRUD означает «C» - Create (Создать), «R» - Read (Читать), «U» - Update - (Изменять) и «D» – Delete (Удалять).

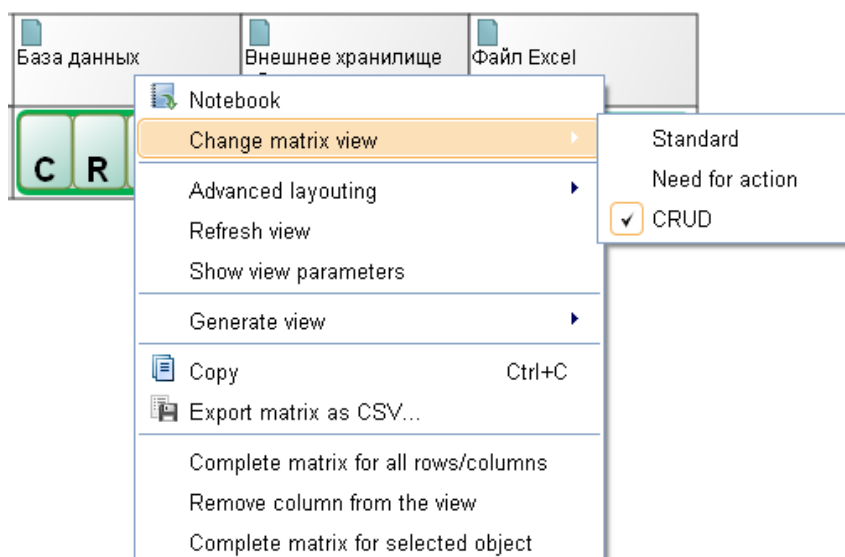


Рисунок 6.31 – Последовательность меню для выбора режима представления матрицы

При выборе режима CRUD в ячейках матрицы будут показываться одна или несколько букв CRUD для индикации возможностей работы с данными.

Возможность режима “Advanced layouting” для построения кластерных моделей будет рассмотрена позднее. Матрицу можно сохранить в виде текстового файла (CSV) для дальнейшего анализа в пункте “Export matrix as CSV”.

При двойном щелчке левой кнопкой мыши на соответствующей ячейке матрицы открывается блокнот, показанный на рис. 6.32. Также можно вызвать блокнот с помощью пункта “Notebook” контекстного меню.

Referenced Data Entities (ADOit -> База данных)

Description:

General Information

Lifecycle

CRUD

☒ Create

☒ Read

☒ Update

☒ Delete

☒ Master

Action Required

☒ Action Required

Reason:

Cardinality:

n...m

☒ Display name and cardinalities

Рисунок 6.32 – Блокнот для ячейки матрицы

В блокноте задается режим работы с данными. Маркер “Master” служит для обозначения главного или корневого приложения.

На рис. 6.33 показан список возможных ассоциаций (Cardinality) объектов.

Cardinality:

n...m

No Entry

0...n

1...n

1...1

n...m

Рисунок 6.33 – Типы ассоциаций между объектами

Пункт “No Entry” как обычно означает, что ассоциация не определена, ассоциация “0...n” не имеет смысла так как отсутствие одного объекта связи означает, что связи просто нет, ассоциация “1...n” – «один ко многим», ассоциация “1...1” – «один к одному» и ассоциация “n...m” – «многие ко многим».

Упражнение 4. Требуется составить матричный взгляд на введенные в предыдущем упражнении приложения и данные в режиме CRUD.

При матричном моделировании в нижнем правом углу рабочего поля имеется кнопка “Show legend”, как показано на рис. 6.34.

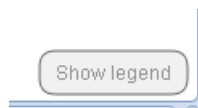


Рисунок 6.34 – Кнопка для выбора обозначений

При нажатии на эту кнопку открывается окно, показанное на рис. 35.

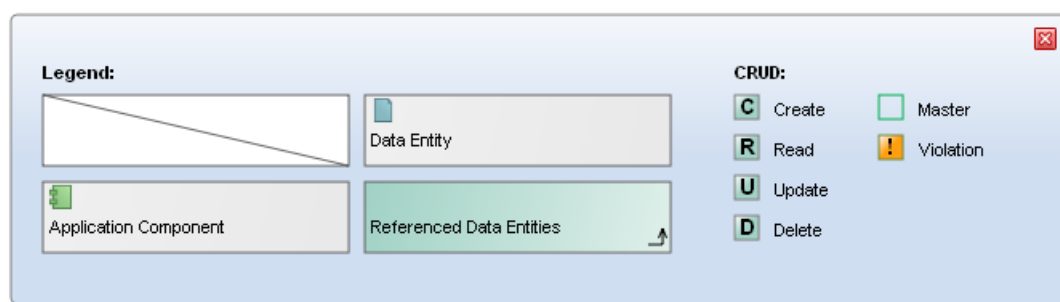


Рисунок 35 – Расшифровка обозначений в матрице

Если выбрать пункты меню “Change matrix view” и “Need for action” на рис. 6.31, то в матрице будут отмечены другим цветом связи данных и приложений, требующих дополнительных действий. Пример представления такой матрицы, приведен на рис. 6.36.

	База данных	Внешнее хранилище объектов	Файл Excel
ADOit			

Рисунок 6.36 – Матрица с отмеченными цветом ячейками, требующими дополнительного действия

Рассмотрим построение матрицы для приложений и принципов. Для примера введем следующие два принципа архитектуры информационной системы: «Принцип безопасности» и «Принцип доступности». Также рассмотрим следующие два приложения: «1С предприятие» и «CRM».

Установим связи каждого приложения с введенными двумя принципами. В результате в блокноте для каждого принципа будут отражены приложения, с которыми они связаны, как показано на рис. 6.37.

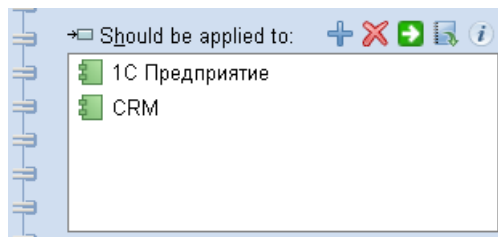


Рисунок 6.37 – Список связанных приложений в блокноте для принципа

При нажатии правой кнопкой мыши на иконке приложения в показанном списке будет открыто меню, показанное на рис. 6.38.

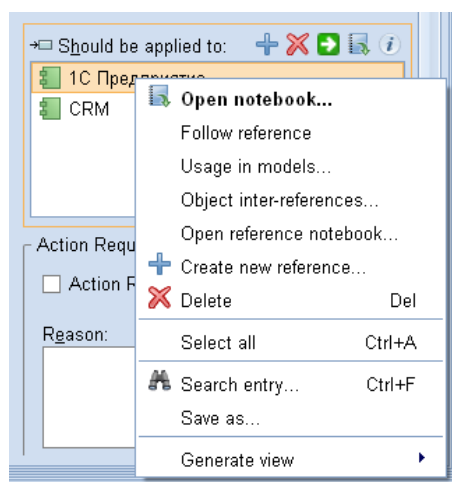


Рисунок 6.38 – Меню для действий со связанным объектом

При выборе пункта меню “Open reference notebook” (Открыть блокнот для связи) будет открыт блокнот, показанный на рис. 6.39.

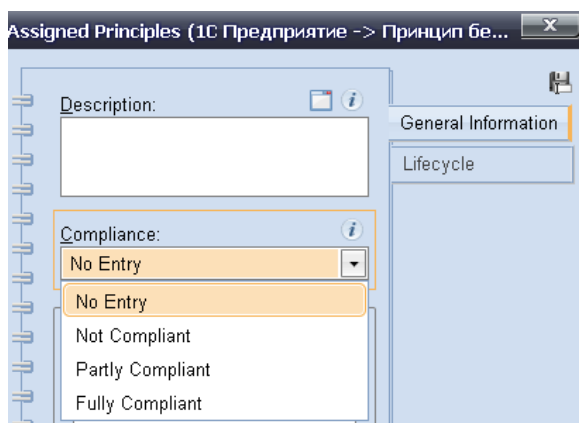


Рисунок 6.39 – Блокнот для параметров связи с принципом

Поле “Compliance” (Соответствие) может принимать следующие значения:

- “Not Entry” – неопределенное значение;
- “Not Compliant” – не соответствует;
- “Partly Compliant” – частично соответствует;
- “Fully Compliant” – полностью соответствует.

Для построения матрицы выберем пункт меню на рис. 6.29 “(Application Component)x(Principle)” для любого из двух рассматриваемых приложений. Затем на макете матрицы с помощью правой кнопки мыши откроем меню, показанное на рис. 6.31 и выберем пункт меню “Compete matrix for all rows/columns”. В результате будет сформирован стандартный вид матрицы, показанный на рис. 6.40.

	Принцип безопасности	Принцип доступности
1С Предприятие		
CRM		

Рисунок 6.40 – Стандартный вид матрицы для приложений и принципов

При выборе пункта меню на рис. 6.31 “Change matrix view”, будет открыт список, показанный на рис. 6.41.

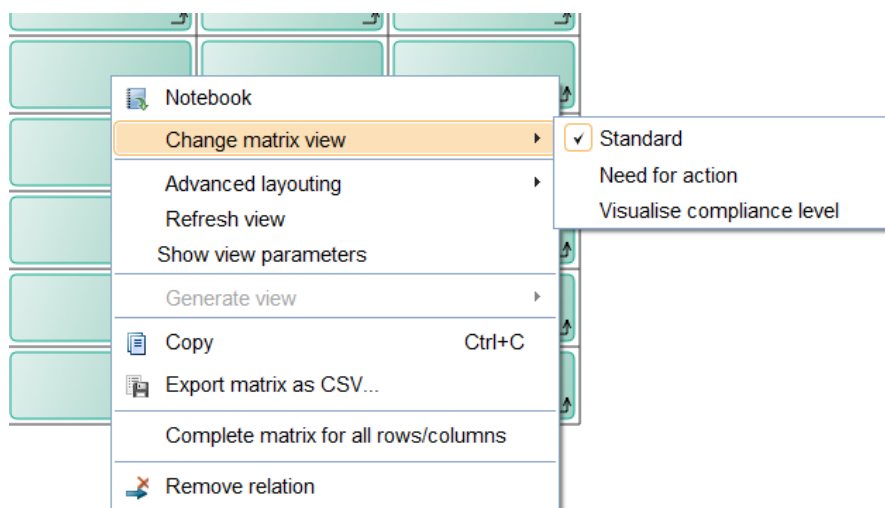


Рисунок 6.41 – Выбор формы представления матрицы

При выборе вида матрицы “Visualize compliance level” матрица будет представлена в форме, показанной на рис. 6.42.

	Принцип безопасности	Принцип доступности
1С Предприятие		
CRM		

Рисунок 6.42 – Матрица с отображенными уровнями согласованности

При этом окно обозначений принимает вид, показанный на рис. 6.43.

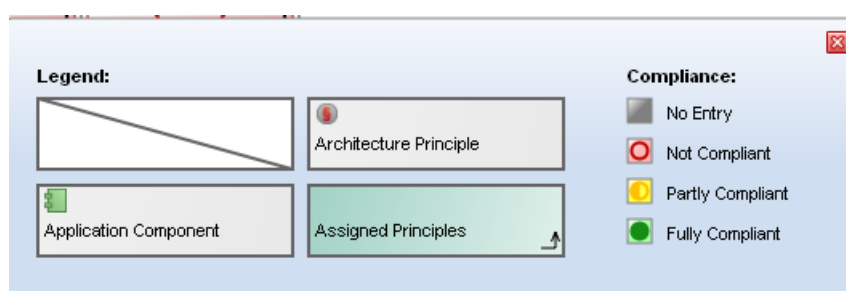


Рисунок 6.43 – Окно обозначений для матрицы приложений и принципов

Упражнение 5. Требуется построить матрицу связей для рассмотренного примера приложений и принципов.

Рассмотрим еще одно представление матрицы для приложений. Пусть имеется три приложения: 1С Предприятие, ADONIS и CRM. Добавим три бизнес-процесса для двух бизнес-сегментов. Пусть предприятие имеет два следующих сегментов бизнеса: «Бесплатный продукт» и «Коммерческий продукт». Пусть также рассматривается три бизнес-процесса: «Бухучет», «Продажи» и «Производство».

Упражнение 6. Создать новые объекты архитектуры, связать их друг с другом. Построить модель ВИА для одного из процессов, как показано на рис. 6.44 для процесса «Производство».

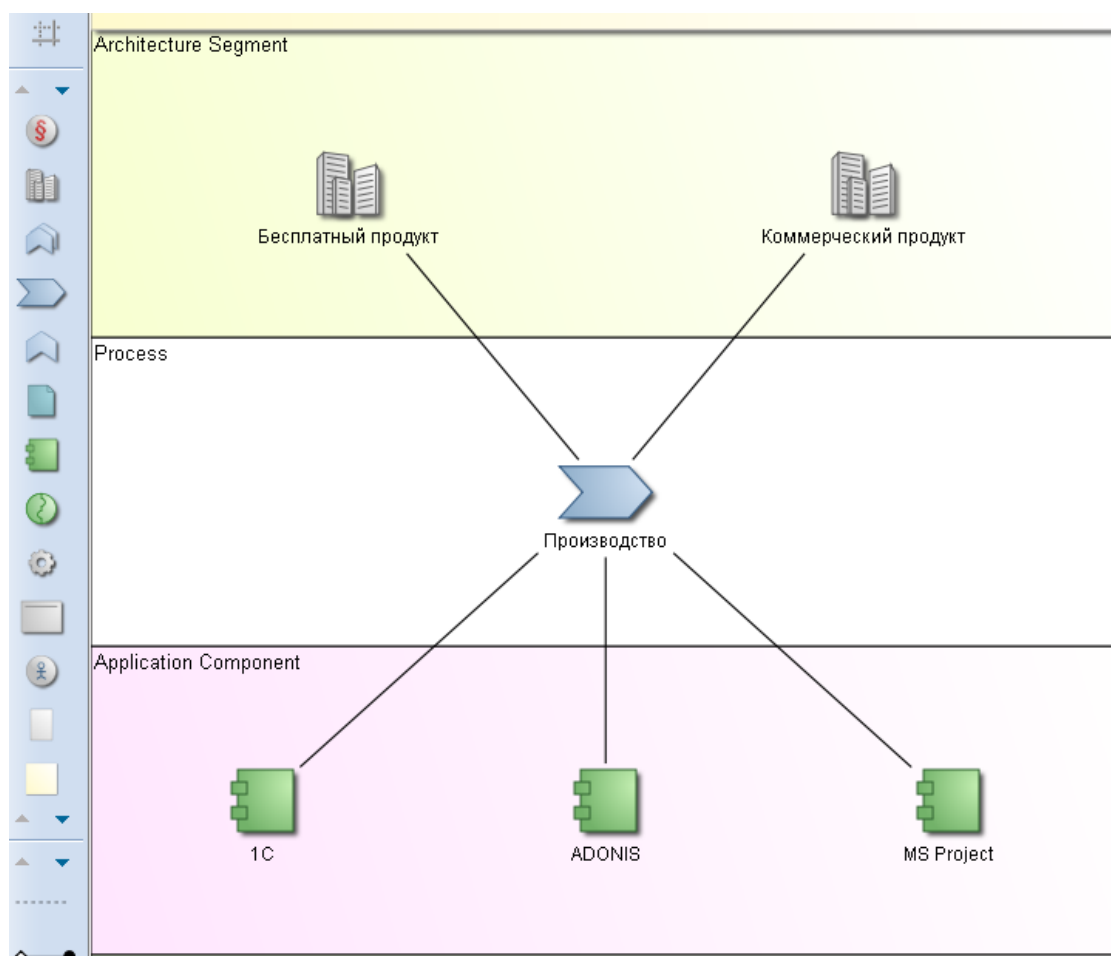


Рисунок 6.44 – Пример модели BIA для процесса «Производство»

Для каждого из приложений в блокнотах произвольно зададим параметры “Investment Strategy” и “Lifecycle State” во вкладке “Lifecycle”.

Для построения матричной модели воспользуемся главным меню, в котором перечислены все возможные типы матричных моделей, как показано на рис. 6.45.

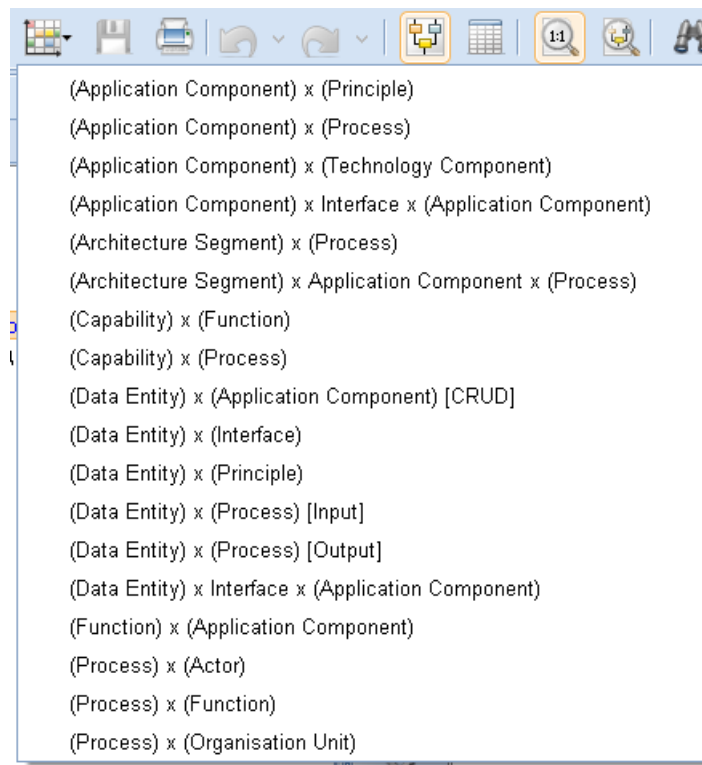


Рисунок 6.45 – Меню для выбора всех видов матричных моделей

Выберем пункт меню “(Architecture Segment) x Application Component x (Process)”. Также, как в предыдущих примерах создадим классическую матрицу, показанную на рис. 6.46.

	Бухучет	Продажи	Производство
Бесплатный продукт	1C MS Project	1C ADONIS CRM MS Project	1C ADONIS MS Project
Коммерческий продукт	1C MS Project	1C ADONIS CRM MS Project	1C ADONIS MS Project

Рисунок 6. 46 – Классический вид матрицы

Для того, чтоб убрать пустые строки и столбцы надо вновь вызвать правой кнопкой мыши меню управления матрицей, как показано на рис. 6.47.

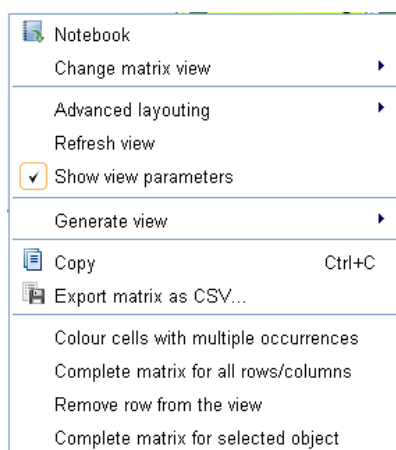


Рисунок 6.47 – Меню для выбора пункта “Show view parameters”

В результате откроется окно с маркерами, показанное на рис. 6.48.

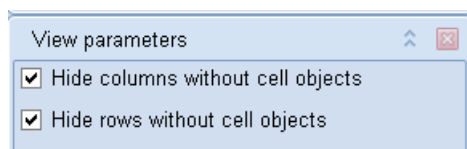


Рисунок 6.48 – Маркеры для удаления пустых строк и столбцов

Если отметить маркер “Hide columns without cell objects” (Скрыть столбцы с пустыми ячейками) или “Hide rows without cell objects” (Скрыть строки с пустыми ячейками), то соответствующие столбцы и строки не будут показаны в матрице.

Изменим вид матрицы. Для этого вызовем меню, показанное на рис. 6.49.

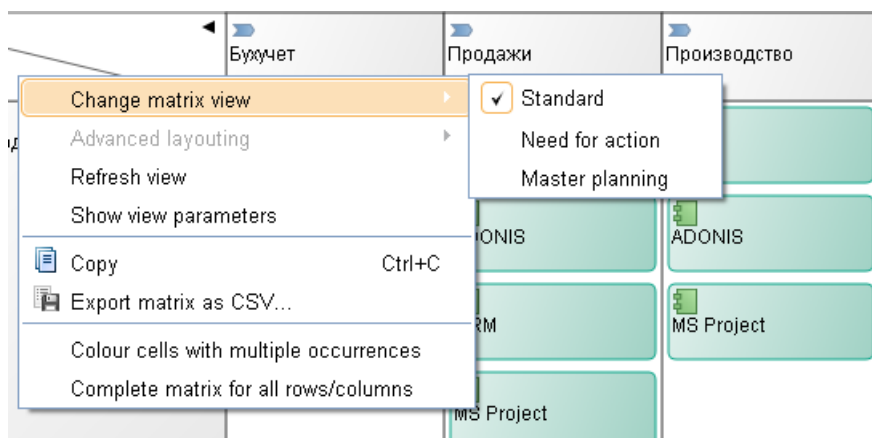


Рисунок 6.49 – Меню для выбора вида матрицы

Выберем вид матрицы “Master planning”. В результате матрица примет вид, показанный на рис. 6.50.

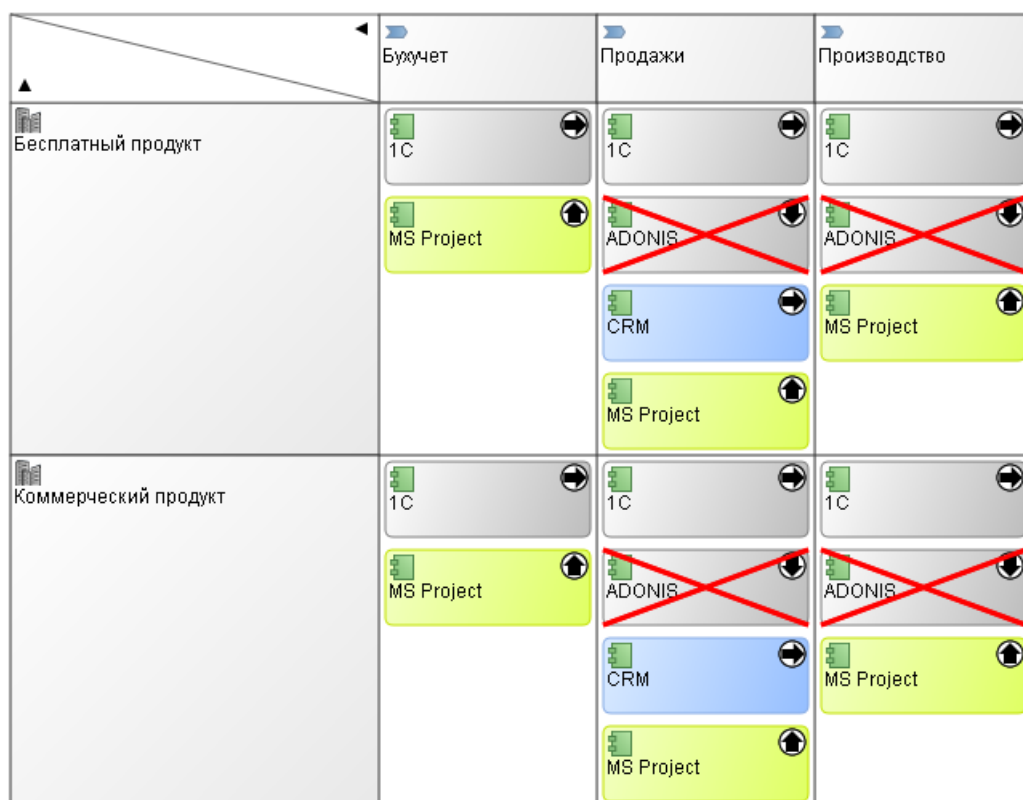


Рисунок 6.50 – Вид матрицы с максимальной информацией

При этом окно расшифровки обозначений принимает вид, показанный на рис. 6.51.

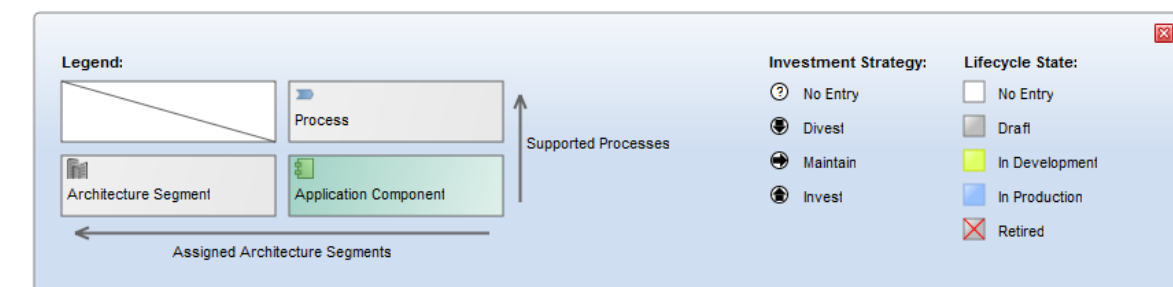


Рисунок 6.51 – Окно расшифровки обозначений

Пункт меню “ Colour cells with multiple occurrences” на рис. 6.49 позволяет отражать ячейки с множественными связями различными цветами, как показано на рис. 6.52.

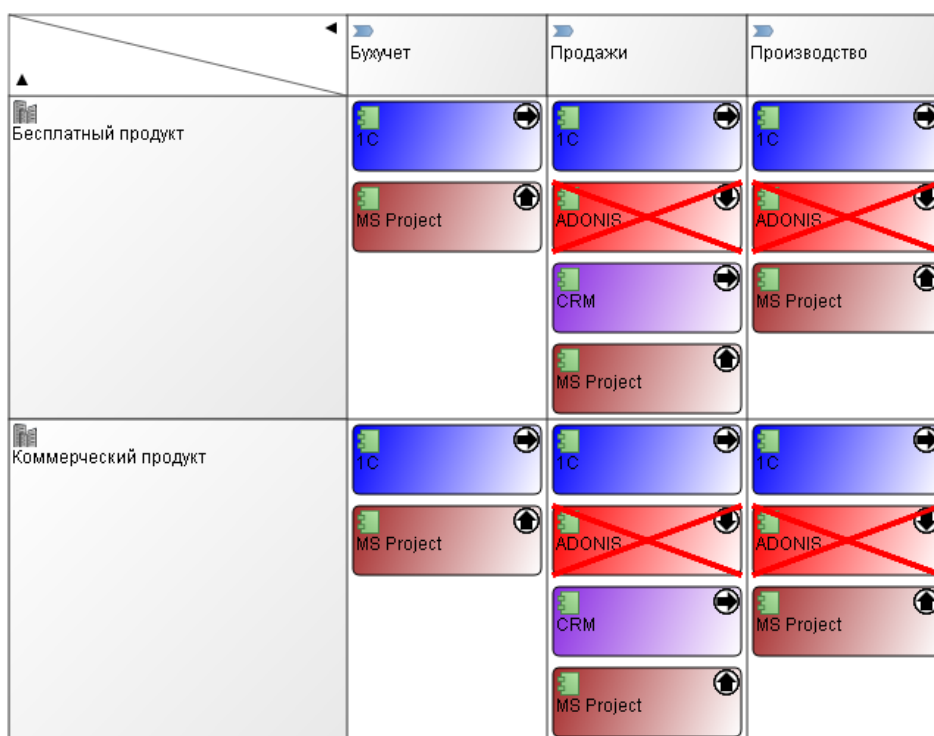


Рисунок 6.52 – Отражение цветом множественных связей

Упражнение 6. Требуется построить матрицу связей для рассмотренного примера с сегментами бизнеса и процессами.

6.7. Объект «Интерфейс» (Interface)

Вспользуемся определением из Википедии: «Интерфейсом в теории систем называется «общая граница» между отдельными системами, через которую они взаимодействуют; совокупность средств и правил, обеспечивающих взаимодействие отдельных систем (например, человека, программного обеспечения, аппаратного обеспечения и т. п.)». При разработке архитектуры информационной системы обычно под интерфейсом понимают программное или аппаратное обеспечение, предназначенное для передачи информации между приложениями или между приложением и человеком.

В системе ADOit объект «Интерфейс» обозначается иконкой, показанной на рис. 6.53.



Рисунок 6.53 – Изображение интерфейса на схеме архитектуры

На рис. 6.54 показан пример блокнота для объекта типа «Интерфейс».

Рисунок 6.54 – Пример блокнота для объекта «Интерфейс»

Новой вкладкой здесь является вкладка «Application Components», показанная на рис. 6.55.

Рисунок 6.55 – Вкладка «Application Components»

На этой вкладке отражаются связи интерфейса с приложениями. Здесь имеется два окна. В первом “Is Provided by” отражается список приложений, поставляющих информацию через интерфейс. Во втором “Is Used by” отражается список приложений, получающих информацию через интерфейс.

На рис. 6.56 показана вкладка для отображения связей интерфейса с данными. Приводится список объектов типа «Данные», которые участвуют в передаче информации через этот интерфейс.

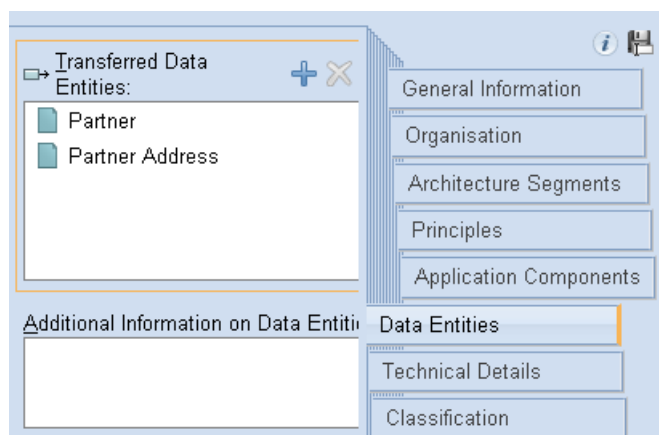


Рисунок 6.56 – Вкладка связи интерфейса с данными

На рис. 6.57 показана вкладка для технических параметров интерфейса.

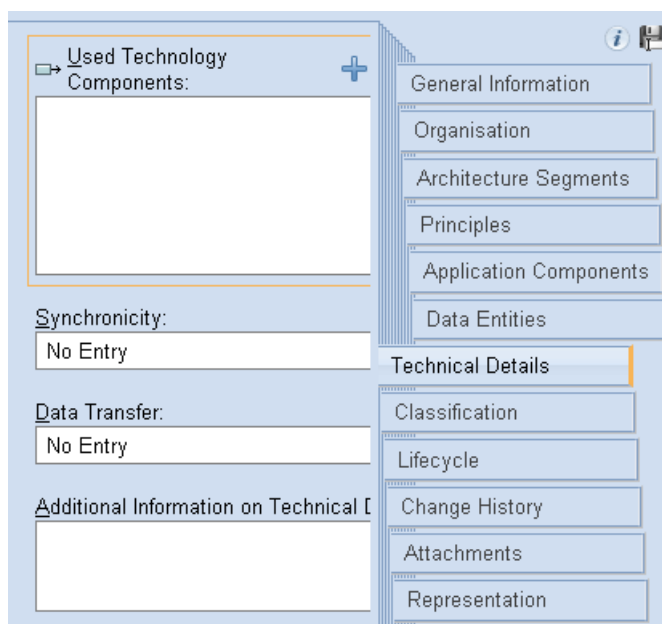


Рисунок 6.57 – Технические параметры интерфейса

Окно “Used Technology Components” служит для указания связей с технологическими компонентами архитектуры, которые будут рассмотрены позже.

Поле “Synchronicity” задает режим работы интерфейса. Возможные значения параметра показаны на рис. 6.58.

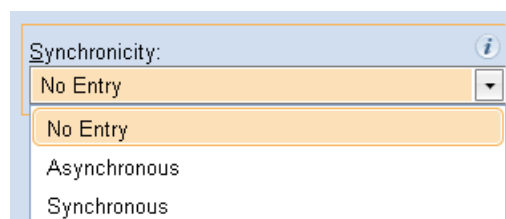


Рисунок 6.58 – Режимы работы интерфейса

Асинхронный режим “Asynchronous” предполагает, что приложение и интерфейс работают по времени независимо друг от друга. То есть приложение ждет, когда придет информация от интерфейса. При этом приложение выполняет свои функции автономно.

Синхронный режим “Synchronous” предполагает, что существует определенный алгоритм синхронизации работы двух устройств. Это может задаваться специальным таймером, когда обе программы осуществляют связь или более сложными методами.

На рис. 6.59 показаны возможные параметры поля “Data Transfer”, которые задают различные способы передачи информации через интерфейс.

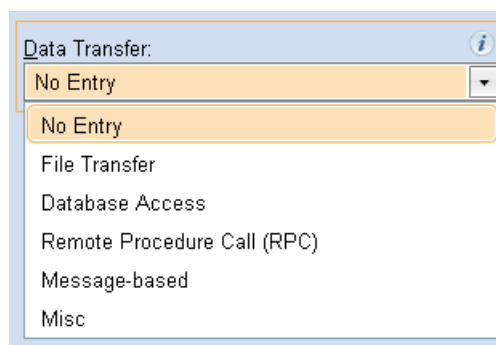


Рисунок 6.59 – Пути передачи информации через интерфейс

Возможны следующие способы передачи информации:

- 'No Entry' – способ не определен;
- 'File Transfer' – информация передается с помощью файла;
- 'Database Access' – информация передается путем доступа к базе данных;

'Remote Procedure Call' – информация передается с помощью вызова удаленной процедуры;

'Message-based' - передается сообщение;

'Misc' - смешанный способ передачи информации.

На рис. 6.60 показана вкладка “Classification”. Все значения параметров полностью совпадают с соответствующими параметрами для приложения.

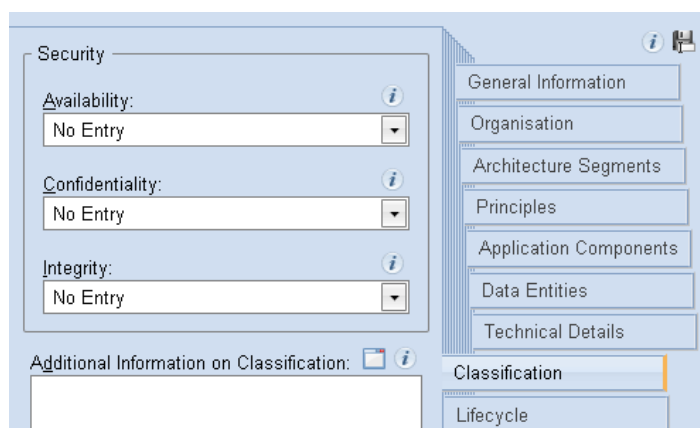


Рисунок 6.60 – Вкладка “Classification”

Остальные вкладки также идентичны соответствующим вкладкам для приложения.

Связь с интерфейсами можно указывать также и в блокноте для приложения, как показано на рис. 6.61.

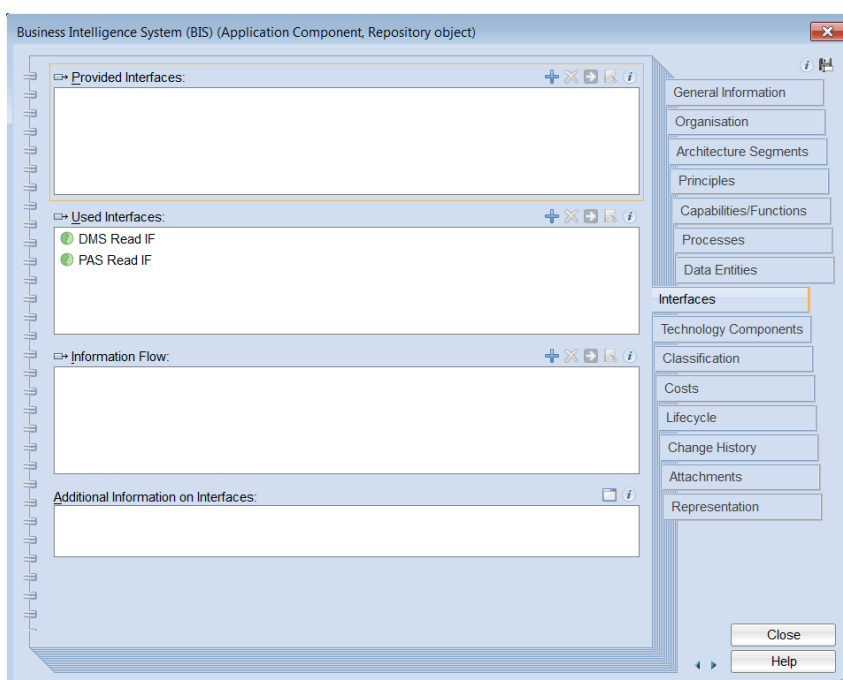


Рисунок 6.61 - Вкладка 'Interfaces' для приложения

На вкладке можно указать связи со следующими типами интерфейсов:
'Provided Interfaces' – интерфейсы, передающие информацию приложению. На графической схеме связь приложения с этим интерфейсом показывается линией с пустым кружочком ○—;

'Used Interfaces' – интерфейсы передающие информацию от приложения. На графической схеме связь приложения с этим интерфейсом показывается линией с залитым кружочком и скобкой ●—(.

На рис. 6.62 показан пример модели информационной системы для банка.

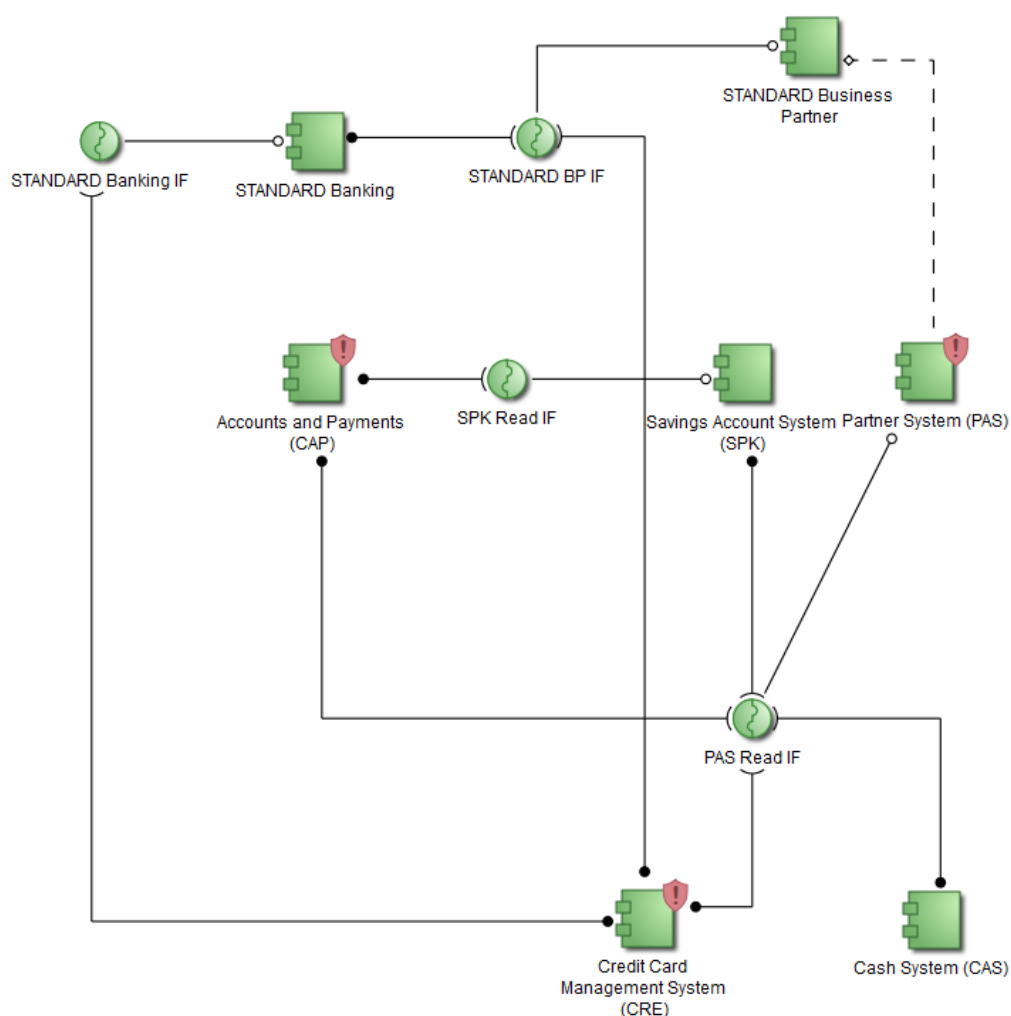


Рисунок 6.62 - Пример модели информационной системы банка

6.8. Модель технологической архитектуры

Модель технологической архитектуры создается выбором типа новой модели "Technology Architecture" Из списка создаваемых новых моделей,

показанного раньше. В результате открывается новое рабочее поле с палитрой инструментов, показанной на рис. 6.63.



Рисунок 6.63 – Палитра инструментов для создания модели технологической архитектуры

При построении технологической архитектуры новым объектом является объект «Технологический инструмент» (Technology Component). Для графического представления этого объекта в модели используется иконка, показанная на рис. 6.64.



Рисунок 6.64 – Графическое представление объекта «Технологический инструмент» (Technology Component)

В хранилище данных элементы этого класса можно разбить по видам деятельности или по сегментам бизнеса и поместить в папки с соответствующими именами. Пример структуры папок с технологическими инструментами в области информационных технологий показан на рис. 6.65.



Рисунок 6.65 – Пример классификации технологических инструментов

Блокнот для объекта типа «Технологический инструмент» имеет вид, показанный на рис. 6.66.

Рисунок 6.66 – Блокнот для объекта ‘Technology Component’

Новая вкладка здесь это вкладка “Usage” (Применение), вид которой показан на рис. 6.67.

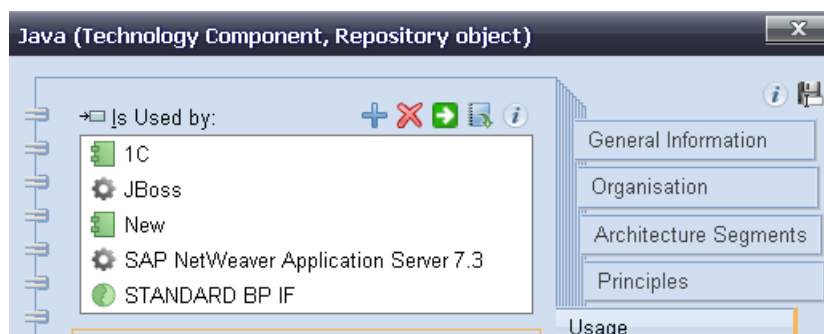


Рисунок 6.67 – Вкладка “Usage”

Здесь указываются объекты типа «Приложение», «Интерфейс» и другие технологические элементы, которые используют данный технологический элемент.

Рассмотрим вкладку блокнота ‘Lifecycle’ (Жизненный цикл), показанную на рис. 6.68.

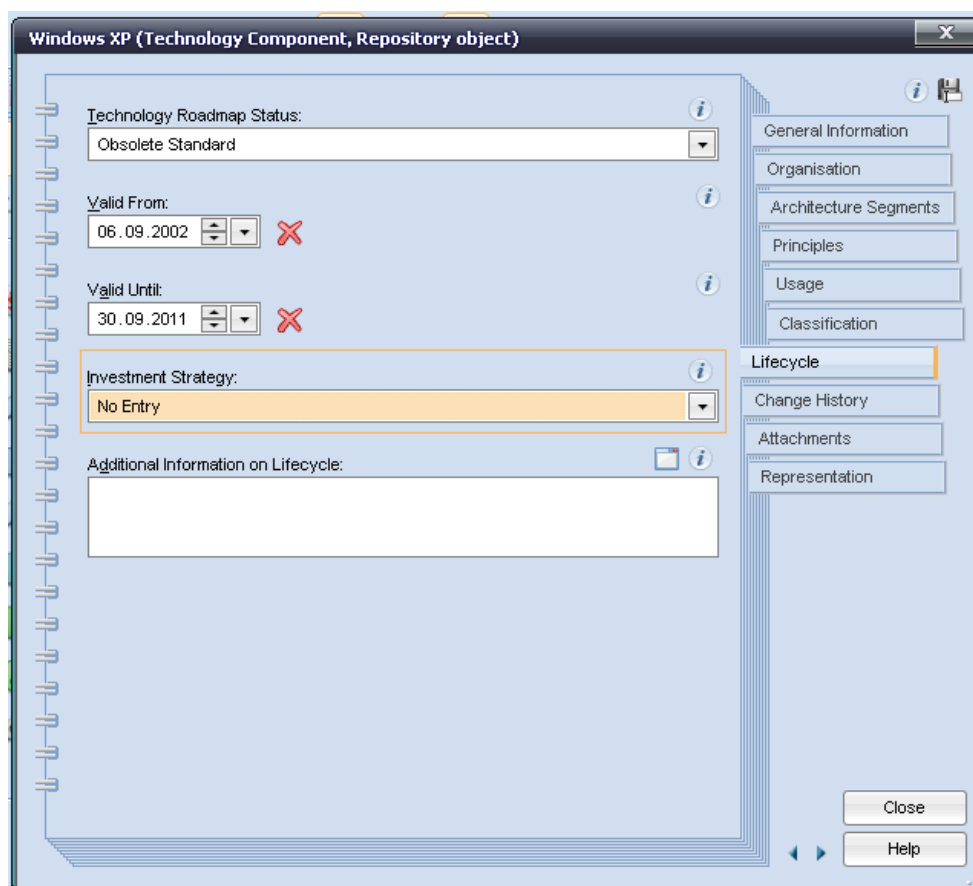


Рисунок 6.68 - Вкладка с параметрами жизненного цикла технологического элемента

Вкладка имеет некоторое отличие от предыдущих рассмотренных вкладок этого типа. Поле 'Technology Roadmap Status' (Статус технологии на дорожной карте) служит для определения состояния стандарта технологии. Возможные значения этого поля показаны на рис. 69.

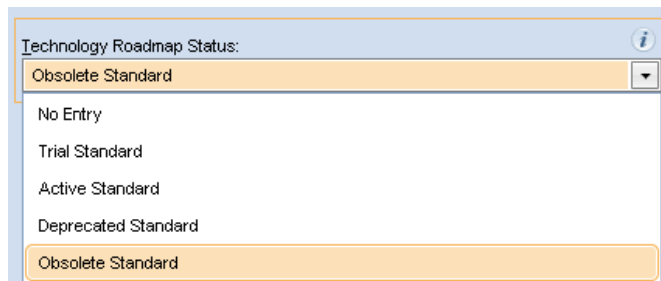


Рисунок 6.69 - Меню для выбора значения статуса технологии

Поле может принимать следующие значения:

- “No Entry” – значение не задано;
- “Trial Standard” – стандарт в состоянии испытания;
- “Active Standard” – действующий стандарт;
- “Deprecated Standard” – не рекомендуемый стандарт;
- “Obsolete Standard” - устаревший стандарт.

Поле 'Investment Strategy' (Стратегия финансирования) имеет те же значения, что были рассмотрены ранее.

Для рассмотренных ранее объектов архитектуры можно устанавливать связи с технологическими инструментами, которые поддерживают эту функцию. Пример установки связи функции с приложением показан на рис. 6.70.

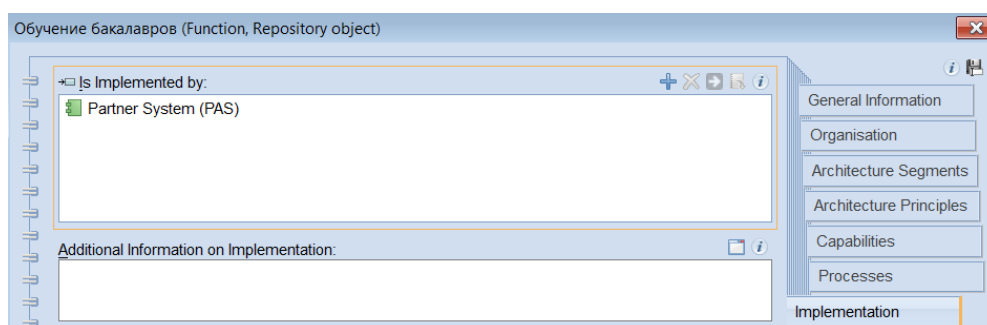


Рисунок 6.70 - Установка связи с приложением

Для объектов типа "Функция" можно формировать графические образы типа 'BIA - Business Impact Analysis' (Анализ взаимосвязей), 'Matrix' (Представление связи в виде матрицы) и 'Portfolio' (Представление связи в

виде пузырьковой диаграммы. Меню выбора типа представления показано на рис. 6.71.

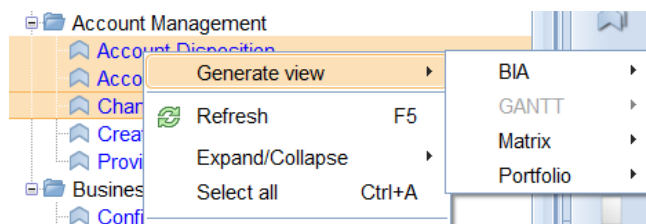


Рисунок 6.71 - Меню для выбора типа представления для функции

При выборе графического представления 'BIA' строится только одна схема взаимосвязи процессов, функций и приложений. Выбор показан на рис. 6.72, а сама диаграмма на рис. 6.73.

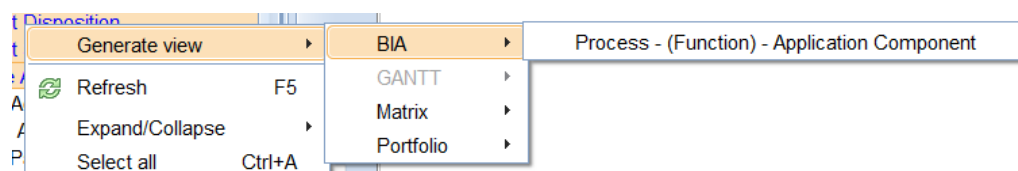


Рисунок 6.72 - Выбор типа диаграммы для функции

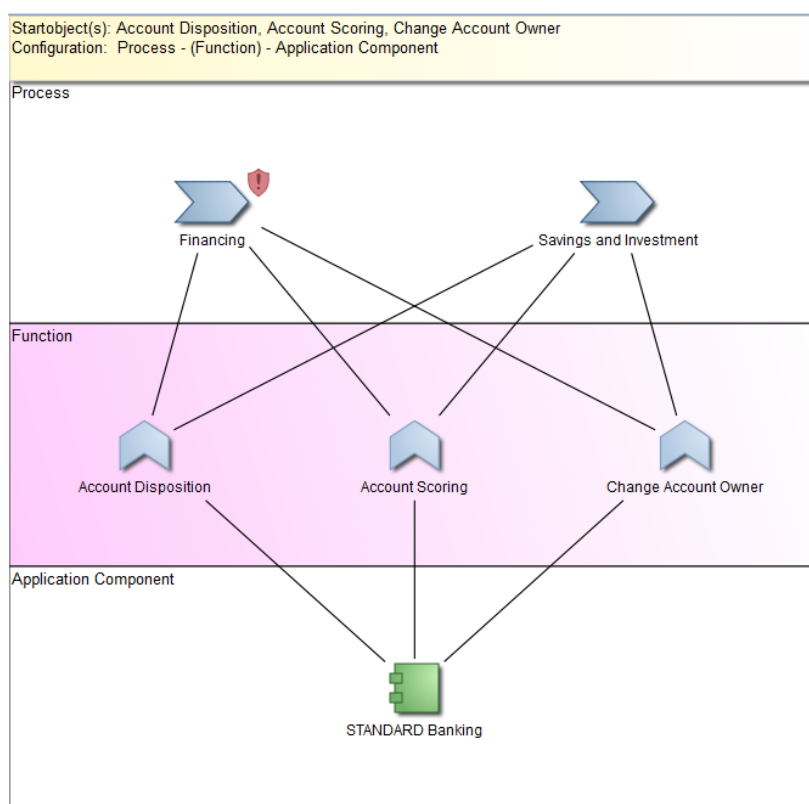


Рисунок 6.73 - Диаграмма взаимосвязи процессов, функций и приложений

В модели ВІА можно использовать все объекты архитектуры. Пример такой модели показан на рис. 6.74.

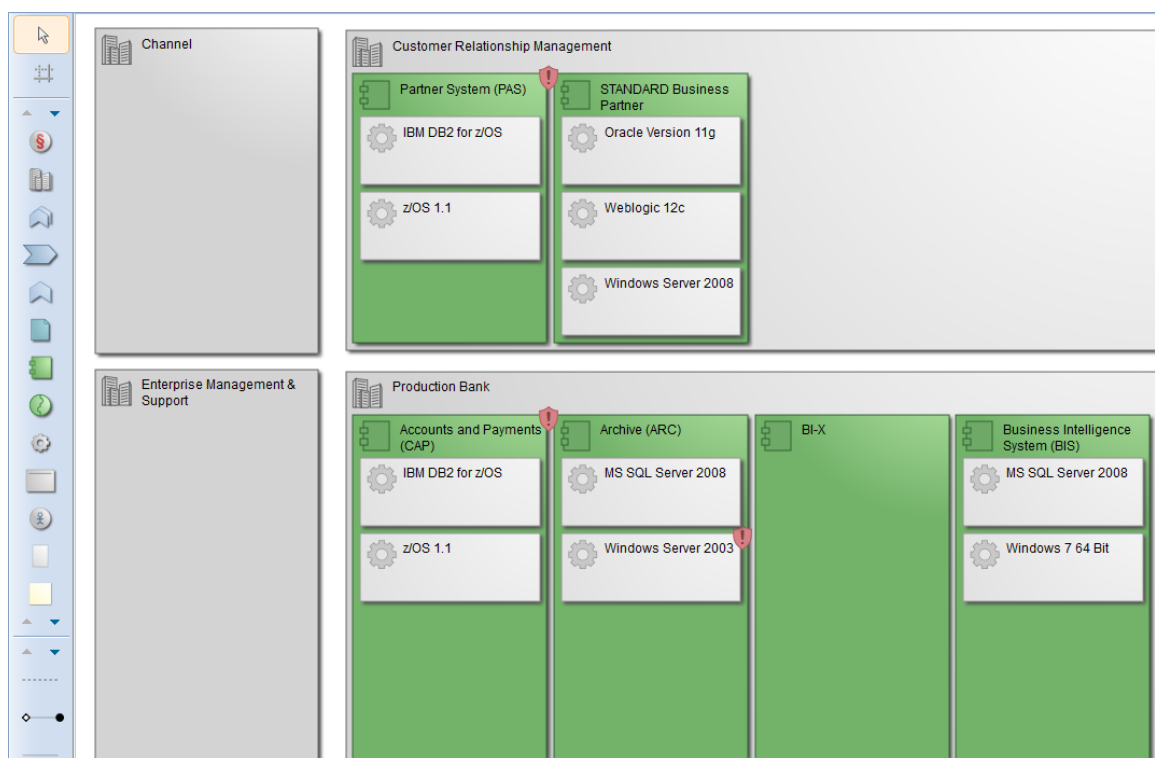


Рисунок 6.74 – Пример модели архитектуры в режиме ВІА

6.9. Построение графика Гантта

Для объектов, которые содержат информацию о жизненном цикле имеется возможность представлять эту информацию в виде графика Гантта. Для этого надо отметить нужные объекты, как показано на рис. 6.75, и выбрать соответствующий тип графика.

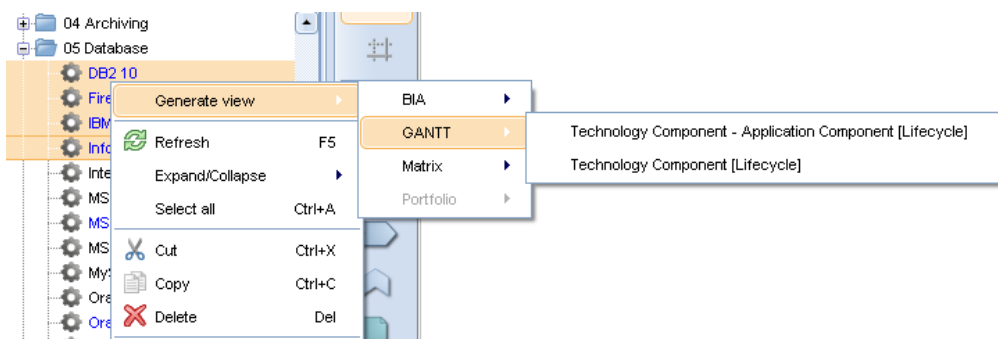


Рисунок 6.75 - Меню для выбора графического представления в виде диаграммы Гантта

Первый тип позволяет строить график Гантта для двух объектов: технологического элемента и приложения. Второй тип графика только для технологического элемента. На рис. 6.76 показан результат построения диаграммы Гантта только для одного элемента.

GANTT View: Technology Component [Lifecycle]

Configuration: Technology Component

Attributes: Valid From and Valid Until

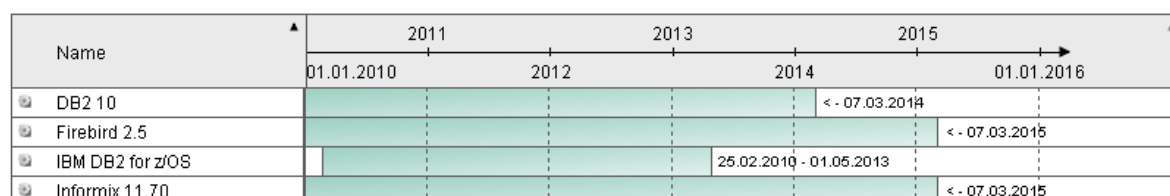


Рисунок 6.76 - Диаграмма Гантта для одного технологического элемента

На рис. 6.77 показан пример построения диаграммы Гантта для двух объектов: приложения и технологического элемента.

GANTT View: Application Component - Technology Component [Lifecycle]

Configuration: Application Component, Used Technology Components (outgoing), Technology Component

Attributes: Valid From and Valid Until

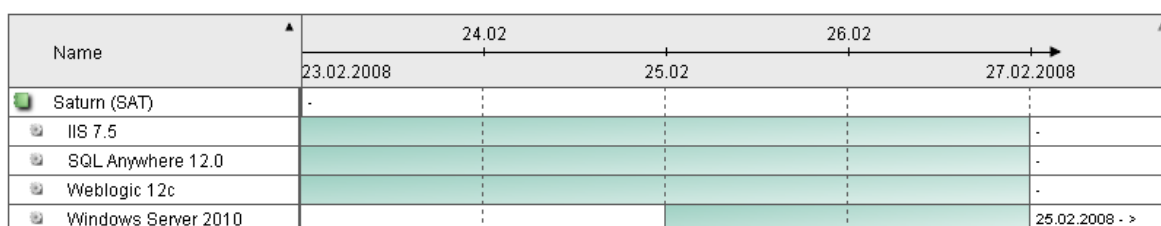


Рисунок 6.77 - Диаграмма Гантта для двух объектов

На рис. 6.78 показаны все виды моделей архитектуры предприятия, поддерживаемые системой ADOit-CE. Подробно не рассматривалась только табличная модель, но она создается автоматически из описания объектов, хранящихся в базе данных системы и носит более справочный характер, чем средство моделирования.

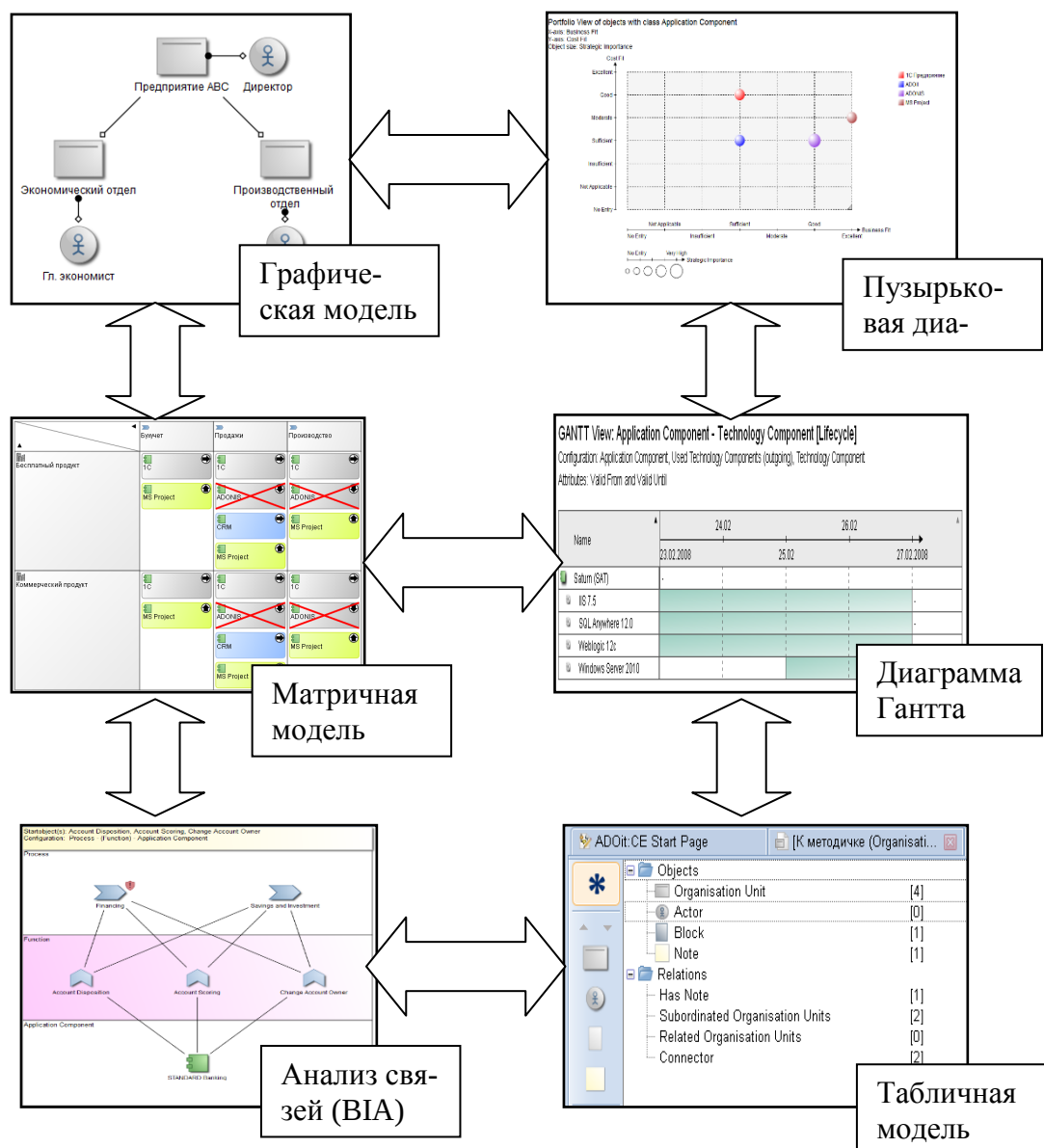


Рисунок 6.78 – Виды моделей в системе ADOit

6.10. Два упражнения для закрепления материала

Упражнение 7. Пусть требуется составить схему информационной системы, состоящей из следующих элементов:

Приложения: бухгалтерия, логистика, управление бизнес-процессами.

Программная система - 1С версии 8.3.

Интерфейс - с помощью СУБД MS SQL.

В начале создадим новые объекты в хранилище, а затем на основании этих объектов создадим графические модели.

Шаг 1. Введем технологический элемент - систему 1С версии 8.3. Для этого создадим папку "Упражнение 1", как показано на рис. 6.79 и 6.80.

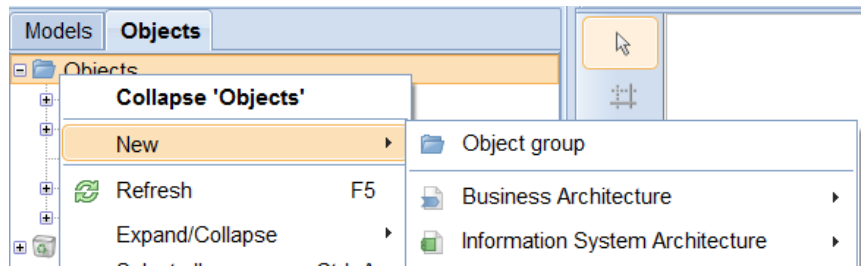


Рисунок 6.79 - Выбор пункта меню 'Model group' для создания новой папки

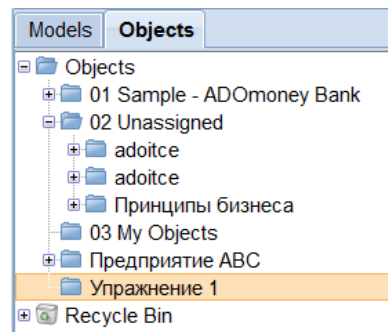


Рисунок 6.80 - Создание новой папки с именем "Упражнение 1"

Затем в созданной папке создадим новый объект, для чего выберем пункт меню 'Technology Component', как показано на рис. 6.81.

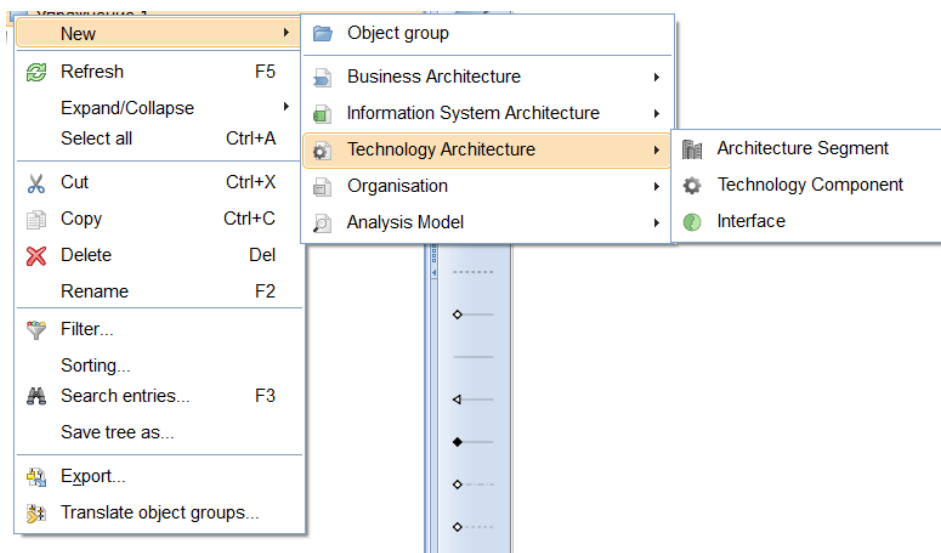


Рисунок 6.81 - Выбор пункта меню для создания технологического компонента

Дадим элементу новое имя - 1С вер 8-3. Результат показан на рис. 6.82.

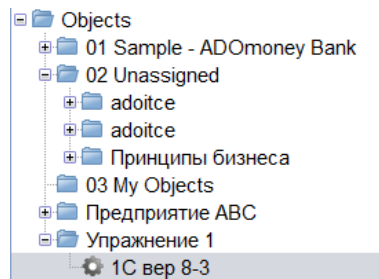


Рисунок 6.82 - Создан новый технологический компонент в хранилище

Шаг 2. Создадим объекты для приложений. Для этого выберем пункт меню 'Application Component', как показано на рис. 6.83. Выполним эти действия для всех трех приложений и переименуем элементы.

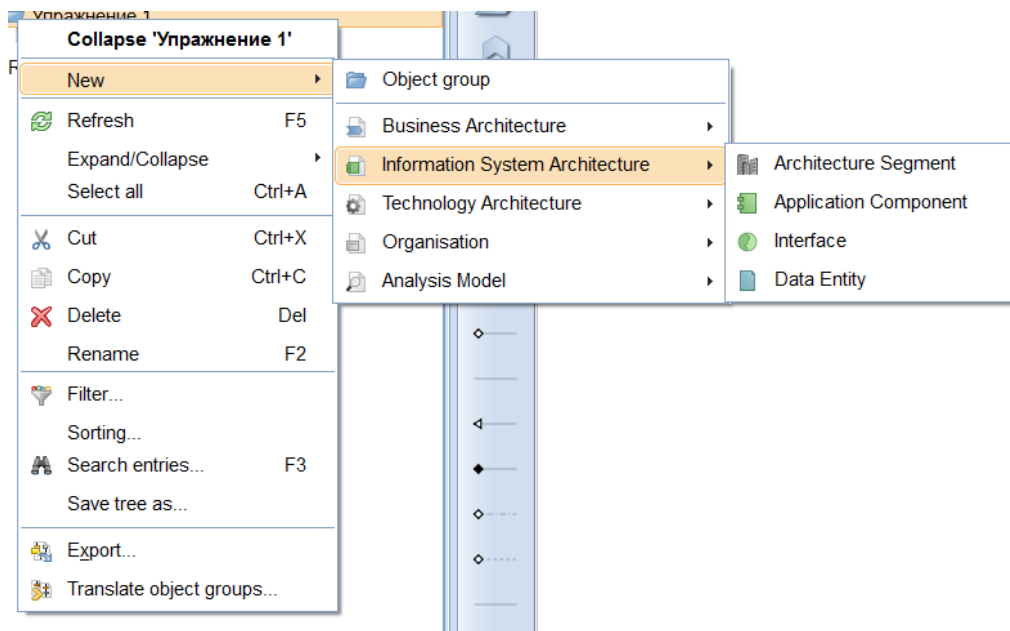


Рисунок 6.83 - Выбор пункта меню 'Application Component'

В результате в хранилище данных будут созданы новые объекты, как показано на рис. 6.84.

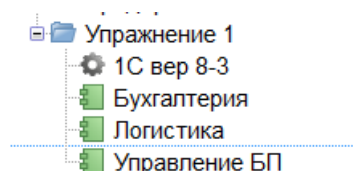


Рисунок 6.84 - Создано три новых объекта

Шаг 3. Создадим первоначальный вариант графической схемы модели информационной системы. Для этого вначале создадим папку "Упражнение1" в директории для моделей и в этой папке создадим новую модель типа 'Information System Architecture', как показано на рис. 6.85.

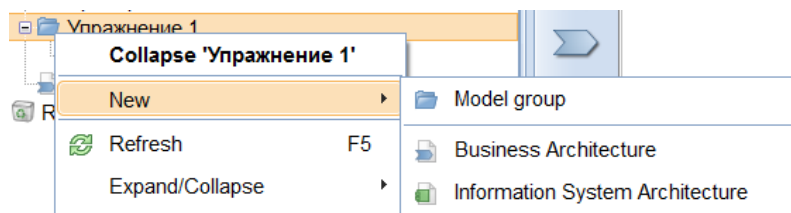


Рисунок 6.85 - Выбор пункта меню для создания модели

В результате будет открыто пустое рабочее поле, в котором можно создавать графическую схему модели. Переименуем модель, как показано на рис. 6.86.

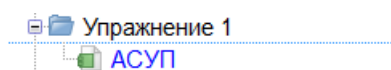


Рисунок 6.86 - Создание заготовка для новой модели

Перетащим приложения с помощью мыши из хранилища объектов на рабочее поле. В результате получим модель, показанную на рис. 6.87.

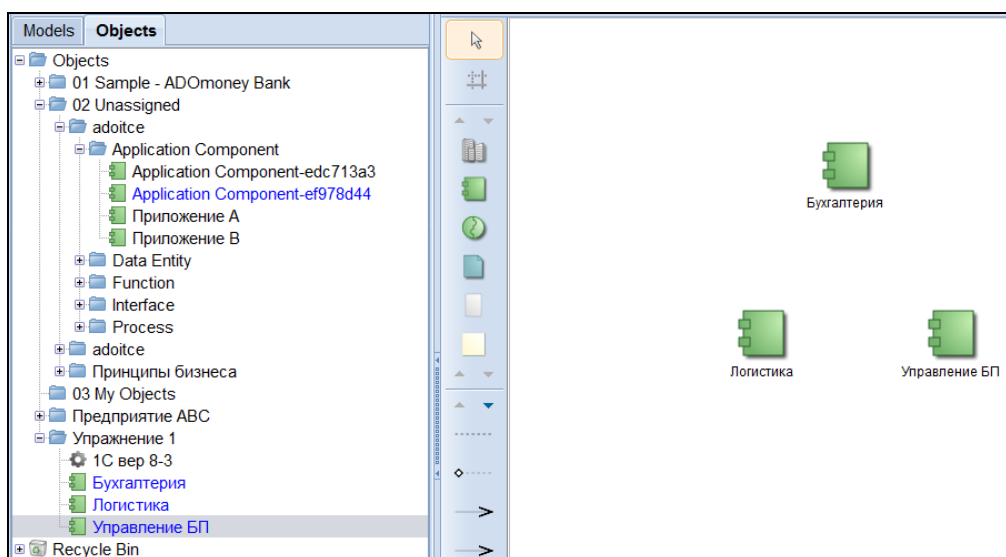


Рисунок 6.87 - Первоначальный вариант модели

Шаг 4. Добавим в модель интерфейсы. Для этого сделаем активным значок графического представления интерфейса и отметим место на схеме,

где его разместить. Каждый клик мыши размещает новый элемент на схеме. переименуем интерфейсы. Результат показан на рис. 6.88.

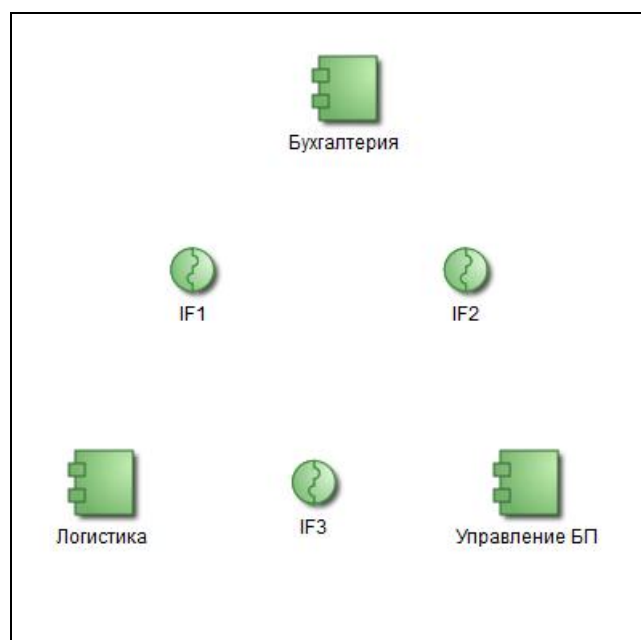


Рисунок 6.88 - Добавлены в схему интерфейсы

При выполнении этих действий в хранилище объектов будут добавлены новые объекты. При этом они добавляются в рабочую папку Unassigned/adoitce/interface, как показано на рис. 6.89.

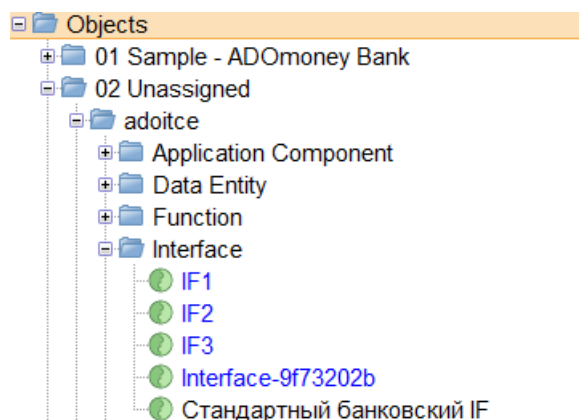


Рисунок 6.89 - Объекты добавлены в хранилище

Перенесем созданные объекты в папку "Упражнение 7". Проще всего это сделать с помощью мыши.

Шаг 5. Создадим графические связи с интерфейсами. Будем считать, что все интерфейсы используются в настоящее время. Результат показан на рис. 6.90.

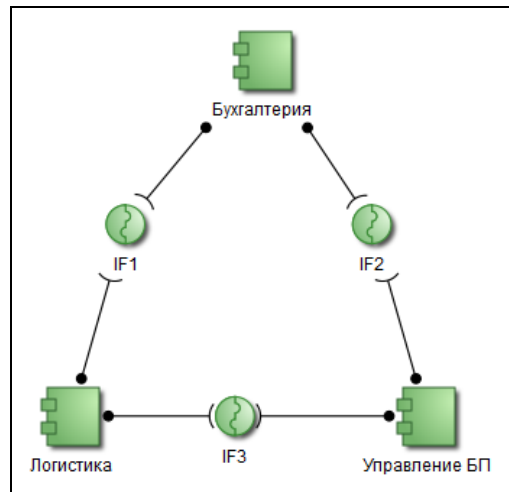


Рисунок 6.90 - Модель со связями приложений и интерфейсов

При выполнении этих действий система автоматически добавит связи в блокноты приложений и интерфейсов. Пример соответствующей вкладки для интерфейса показан на рис. 6.91.

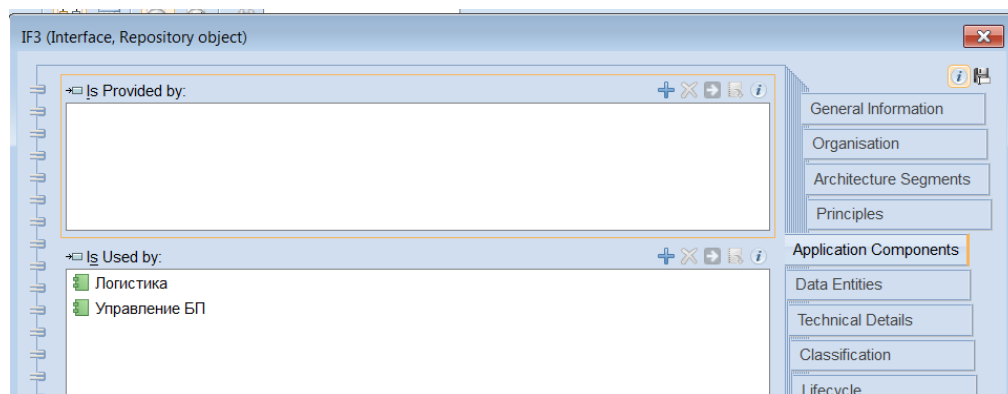


Рисунок 6.91 - Ссылки на объекты добавлены в блокнот для интерфейса

Шаг 6. Добавим ссылки на используемый программный продукт 1С в блокноты для созданных объектов модели. Пример ссылки приложения "Бухгалтерия" показан на рис. 6.92.

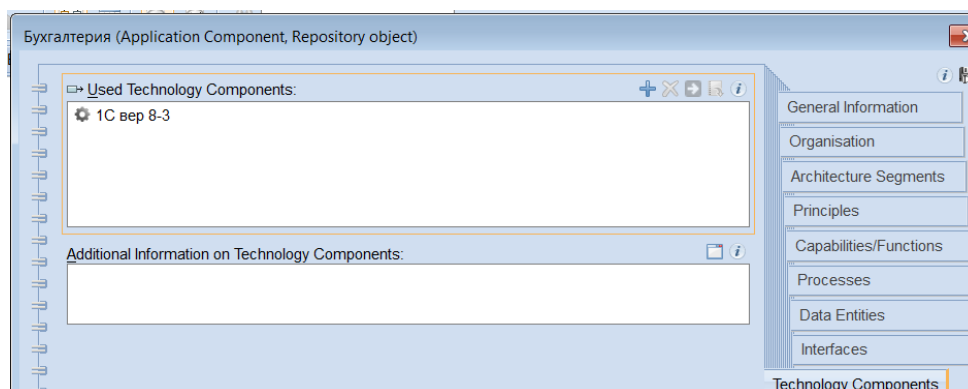


Рисунок 6.92 - Ссылка на систему 1С

Упражнение 8. Требуется провести анализ связей между созданными объектами.

Шаг 1. Создадим новую пустую модель типа 'Analysis Model'. Пункты меню для этого действия показаны на рис. 6.93.

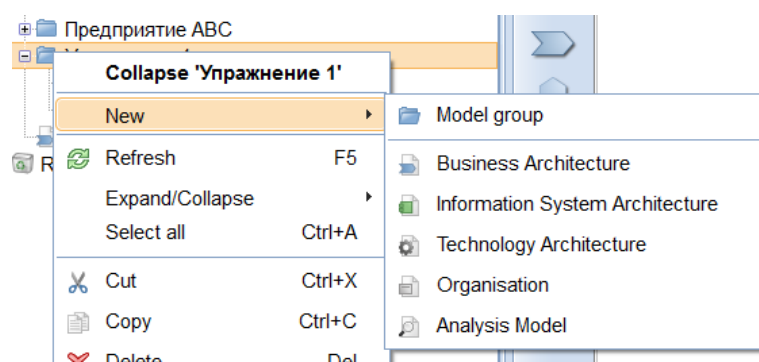


Рисунок 6.93 - Создание модели типа 'Analysis Model'

Шаг 2. Перетащим с помощью мыши на рабочее поле из хранилища следующие четыре объекта: приложение "Бухгалтерия", два интерфейса и программный продукт 1С. В результате получится схема, показанная на рис. 6.94. Все связи будут добавлены автоматически.

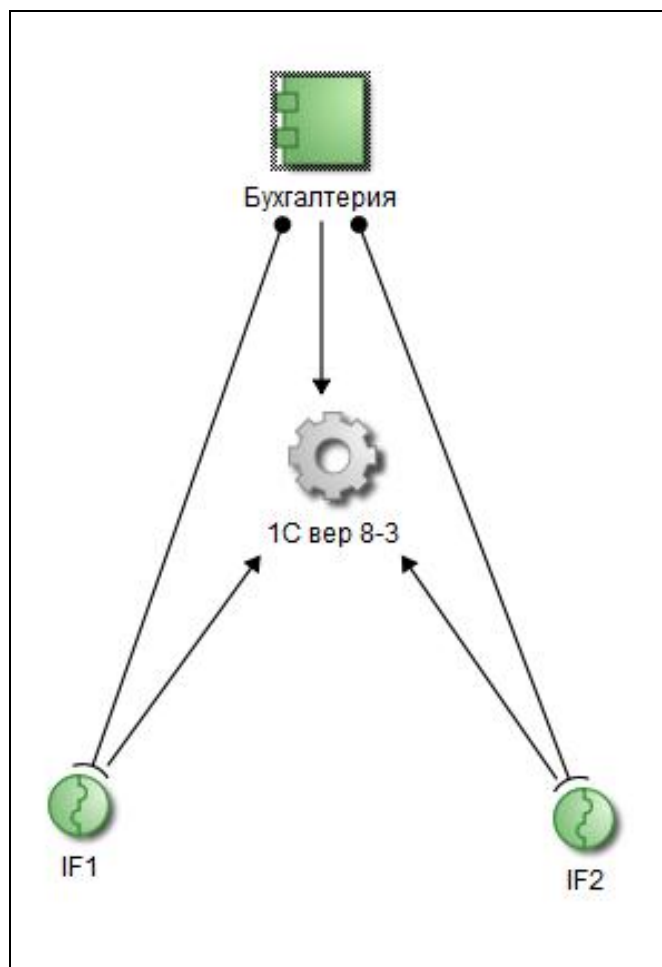


Рисунок 6.94 - Аналитическая модель связей между объектами

6.11. Задание по выполнению практикума

- Выполнить все упражнения, описанные в практикуме.
- Выполнить индивидуальное задание.
- Подготовить отчет.

6.12. Содержание отчета

Отчет должен содержать следующую информацию: задание на выполнение практикума, копии окон экрана с выполненными упражнениями, копии окон экрана с выполненным индивидуальным заданием, выводы.

6.13. Варианты заданий

В табл. 1 приведены списки возможных приложений.

Таблица 1 – Список возможных приложений

№	Приложение
1	Управление маркетингом и продажами
2	Управление заказами
3	Планирование цепочки поставок
4	Управление снабжением
5	Управление логистикой
6	Управление финансами
7	Управление персоналом
8	Управление обучением
9	Управление проектами
10	Управление ТО и ремонтами
11	Управление сервисом
12	Управление рисками
13	Управление продажами
14	Управление заказами
15	Планирование цепочки поставок
16	Управление снабжением
17	Управление производством
18	Управление финансами
19	Планирование капвложений
20	Стратегическое планирование и моделирование
21	Сбалансированная система показателей деятельности

В табл. 2 приведены варианты индивидуальных наборов приложений.

Таблица 2 Варианты приложений

№ варианта	Пакет приложений
1	1, 2, 3, 4, 5
2	6, 7, 8, 9, 10
3	11, 12, 13, 14, 15
4	16, 17, 18, 19, 20
5	21, 2, 4, 6, 8
6	10, 12, 14, 16, 18
7	20, 1, 3, 5, 7
8	9, 11, 13, 15, 17
9	19, 21, 1, 2, 3
10	4, 5, 6, 7, 8,
11	9, 10, 11, 12, 13
12	14, 15, 16, 17, 18

13	3, 4, 5, 6, 7
14	8, 9, 10, 11, 12
15	13, 14, 15, 16, 17
16	18, 19, 20, 21, 1
17	2, 3, 4, 5, 6
18	7, 8, 9, 10, 11
19	12, 13, 14, 15, 16
20	17, 18, 19, 20, 21
21	1, 4, 7, 10, 13
22	16, 19, 2, 5, 7
23	10, 13, 16, 19, 1
24	4, 7, 10, 15, 20
25	5, 9, 13, 17, 21

Требуется составить модель архитектуры приложений. Заполнить произвольно характеристики приложений. Связать приложения с функциями и бизнес-процессами, созданными в предыдущем практикуме. Построить пузырьковую диаграмму. Построить графическую модель полученной архитектуры.

В табл. 3 приведен список возможных технологических компонентов.

Таблица 3 – Список возможных технологических компонентов

№	Приложение
1	ОС Windows 7
2	ОС Windows 8
3	ОС Windows 10
4	ОС UNIX
5	ОС Linux
6	СУБД MS-SQL
7	СУБД DB-2
8	СУБД Oracle
9	СУБД Progress
10	Система Java
11	Система Pentaho
12	Система C++
13	Система JBoss
14	Система Vertica
15	Система AnyLogic
16	Система ADONIS
17	Система ADOScore
18	Система Piton
19	Система Eclipse
20	Система DMN
21	Система PMining

В табл. 4 приведены варианты индивидуальных наборов технологических компонентов.

Таблица 4 Варианты наборов технологических компонентов

№ варианта	Пакет технологических компонентов
1	1, 8,17,18,19
2	2, 6, 7, 8, 19
3	3,11, 13, 14, 15
4	4, 17, 18, 19, 20
5	5, 12, 4, 6,18
6	1, 12, 14, 16, 18
7	20, 1, 13, 5, 17
8	9, 11, 3, 15, 17
9	19, 21, 1, 12,13
10	4, 15,16, 7, 18,
11	1, 11, 12, 13, 21
12	4, 15, 16, 17, 18
13	3, 14,15, 21, 17
14	2, 9, 10, 11, 12
15	3, 14, 15, 16. 17
16	1, 19, 20, 21, 1
17	2,13, 14, 20, 6
18	3, 8, 9, 10, 11
19	4, 13, 14, 15, 16
20	1, 18, 19, 20, 21
21	2, 14, 7, 10, 13
22	1, 19, 12,15, 7
23	4, 13, 16, 19, 21
24	4, 7, 10, 15, 20
25	5, 9, 13, 17, 21

Требуется построить модель технологических компонентов информационной системы. Связать технологические компоненты с приложениями. Задать характеристики значимости и жизненного цикла компонентов. Построить матричную модель для бизнес-процессов, приложений и технологических компонентов. Построить график Ганта для технологических компонентов.

6.14. Контрольные вопросы

1. Что такое архитектура информационной системы?
2. Что такое приложение?
3. Приведите 10 примеров приложений общего назначения.
4. Приведите 10 примеров приложений специального назначения.
5. Что такое технологический компонент информационной системы?
6. Приведите 10 примеров технологических компонентов информационной системы.
7. Какие объекты можно построить в модели информационной системы?
8. Как связать приложение с бизнес-процессом?
9. Как связать приложение с бизнес-единицей?
10. Как связать приложение с возможностями?
11. Как связать приложение с функциями?
12. Какие объекты можно построить в модели технологических компонентов?
13. Что такое пузырьковая диаграмма?
14. Что такое график Гантта?
15. Что такое ВИА?
16. Что дает матричная модель архитектуры?

Заключение

В настоящее время моделирование архитектуры предприятия приобретает особое значение в связи с широким внедрением в практику менеджмента технологий цифровой экономики [4]. Цифровая экономика включает в себя такие разделы, как интеллектуальная автоматизация, социальные сети, интернет вещей, большие базы данных и искусственный интеллект. Все эти разделы цифровой экономики органически вписываются в современные архитектуры предприятия.

Настоящий практикум является основой для дальнейшего изучения систем управления предприятием. Модель архитектуры предприятия используется для создания систем стратегического управления, систем менеджмента, систем отношений с клиентами и других. На основе данного практикума будут подготовлены следующие учебные пособия по моделированию бизнеса.

Литература

1. Интеграция предприятия. Основа моделирования предприятия: ГОСТ Р ИСО 19439-2008.- Москва, 2010.- С. 33.
2. Промышленные автоматизированные системы. Требования к стандартным архитектурам и методологиям предприятия: ГОСТ Р ИСО 15704-2008. – Москва, 2010.- С.63.
3. Григорьев Л.Ю. Корпоративная архитектура и ее составляющие/ Л.Ю. Григорьев, Д.В. Кудрявцев, С.Л. Горелик [Электронный ресурс].- Режим доступа: http://bigc.ru/theory/innovations/corp_arch.php
4. Иванов В.В. Цифровая экономика: мифы, реальность, перспектива/ Иванов В.В., Малинецкий Г.Г.- М.: Российская академия наук, 2017.- 64 с.- [Электронный ресурс].- Режим доступа: <http://www.ras.ru/FStorage/Download.aspx?id=33cela09-9923-4a21c-ba39489dlc14>
5. Крицкая М. Организационная структура предприятия: виды и схемы [Электронный ресурс].- Режим доступа: <https://kontur.ru/articles/4197>
6. Кудрявцев Д. В. Технологии бизнес-инжиниринга: учеб. пособие /Д. В. Кудрявцев, М. Ю. Арзуманян, Л. Ю. Григорьев. — СПб. : Изд-во Политехн. ун-та, 2014. — 427 с.
7. Цуканов А.В. Моделирование бизнес-процессов/ лабораторный практикум.- Севастополь: СевГУ, 2018.- 204 с.
8. Цуканов А.В. Моделирование и анализ бизнес-процессов в системе ADONIS.- Saarbrücken: LAP Lambert Academic Publishing RU, 2018.- 246 с.
9. *Штейнгарт Е.А., Бурмистров А.Н. Обзор и сравнительная характеристика методологий разработки архитектуры предприятий // Научно-технические ведомости СПбГПУ. Экономические науки. 2016. № 3 (245). С. 111-129.*
10. **ADOIT** – The Enterprise Architecture Suite from the BOC Group [Электронный ресурс].- Режим доступа: <http://www.adoit-community.com>
11. Oracle VM VirtualBox [Электронный ресурс].- Режим доступа: <https://www.oracle.com/virtualization/virtualbox/index.html>

Учебное издание

Цуканов Александр Викторович

**МОДЕЛИРОВАНИЕ АРХИТЕКТУРЫ
ПРЕДПРИЯТИЙ:
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

Учебное пособие

В авторской редакции

Изд. № 9/19. Объем 11,25 п.л.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»