

Петрозаводский государственный университет
Институт Математики и информационных технологий
Кафедра Информатики и математического обеспечения

Основные диаграммы для анализа требований.

Лектор:

к.т.н., доцент Богоявленский Ю. А.
ybgv@cs.karelia.ru



Разделы

- Потоки данных, swimlane, варианты использования
- Диаграмма переходов состояний и таблица состояний.
- Карты диалоговых окон
- Таблицы и деревья решений. Таблицы событий и реакций
- Моделирование данных. Диаграмма сущность–связь
- Моделирование данных. Словарь данных

Потоки данных, swimlane, варианты использования.

Содержание

- Модели визуального представления.
- Диаграмма потоков данных
- Основные компоненты диаграммы потоков данных
- Нотации диаграмм потоков данных
- Правила и рекомендации построения модели DFD
- Основные термины диаграмм потоков данных
- Диаграмма swimlane
- Диаграммы вариантов использования.

Модели визуального представления

Основные модели визуального представления:

- диаграммы потоков данных (data flow diagrams, DFD);
- диаграммы рабочих потоков, такие как диаграммы swimlane;
- диаграммы переходов состояний (state-transition diagrams, STD) и таблицы состояний;
- карты диалоговых окон;
- таблицы и деревья решений;
- таблицы событий и реакций;
- деревья функций;
- диаграммы вариантов использования;
- диаграммы процессов;
- диаграммы «сущность-связь» (entity-relationship diagrams.

Диаграмма потоков данных

Основой данной методологии графического моделирования информационных систем является специальная технология построения диаграмм потоков данных DFD. В разработке методологии DFD приняли участие многие аналитики, среди которых следует отметить Э. Йордона (E. Yourdon). Он является автором одной из первых графических нотаций DFD. В настоящее время наиболее распространенной является так называемая нотация Гейна-Сарсона (Gene-Sarson), основные элементы которой будут рассмотрены в этом разделе.

Основные компоненты диаграммы потоков данных

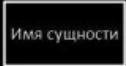
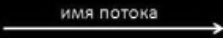
Модель системы в контексте DFD представляется в виде некоторой информационной модели, основными компонентами которой являются различные потоки данных, которые переносят информацию от одной подсистемы к другой. Каждая из подсистем выполняет определенные преобразования входного потока данных и передает результаты обработки информации в виде потоков данных для других подсистем.

Основными компонентами диаграмм потоков данных являются:

- внешние сущности
- накопители данных или хранилища
- процессы
- потоки данных
- системы/подсистемы

Нотации DFD

39

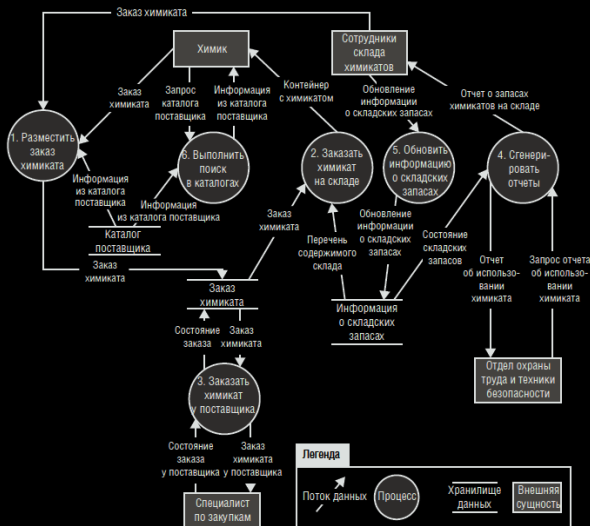
Объект	Нотация Йордана	Нотация Гейна-Сарсона
Процесс		
Внешняя сущность		
Поток данных		
Управляющий поток		
Хранилище данных		

Основные термины диаграмм потоков данных

- Поток данных определяет информацию (материальный объект), передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересылаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т. д.
- Процесс представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.
- Накопитель (хранилище) данных представляет собой абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.
- Внешняя сущность представляет собой материальный объект или физическое лицо, выступающие как источник или приемник информации (например, заказчики, персонал, программа, склад, инструкция).

Диаграмма потоков данных

Уровень 0 диаграммы потока данных для Chemical Tracking System (сокращенная версия)



Правила и рекомендации построения модели DFD

- У каждого процесса (подсистемы) на диаграмме потоков данных должен быть как минимум один входящий и один выходящий поток.
- Процесс должен запускаться на выполнение либо через обрабатываемый, либо через управляющий поток данных. Работа каждого процесса должна завершаться конкретным результатом.
- Каждый накопитель данных также должен иметь как минимум один входящий и один выходящий поток.
- Наличие только входящих потоков в накопитель означает, что информация накапливается, но не используется.
- Наличие только выходящих потоков из накопителя также является ошибкой. Прежде чем использовать данные из накопителя, они должны там появиться в результате работы какого-либо процесса (подсистемы, внешней сущности).
- Исключением из правил считается случай, когда накопитель является внешней сущностью. Тогда допускается наличие либо только входящих стрелок, либо только выходящих стрелок.

Диаграмма swimlane

Диаграммы swimlane — один из способов представления шагов, из которых состоит бизнес-процесс или операции предлагаемой программной системы. Это вариант блок-схемы, разделенной на видимые подкомпоненты, которые называются дорожками (lanes). Дорожки могут представлять разные системы или действующих лиц, которые выполняют шаги процесса. Диаграммы swimlane чаще всего применяются для отображения бизнес-процессов, последовательности процессов или взаимодействия системы и пользователя.

Диаграмма swimlane

Они могут показать, что происходит внутри кружка процесса в диаграмме потоков данных. Они помогают связать вместе функциональные требования, которые позволяют пользователям выполнять определенные задачи. Их можно также применять для выполнения детального анализа в процессе выявления требований, относящихся к каждому шагу процесса.

Диаграмма swimlane — одна из самых простых диаграмм для понимания заинтересованными лицами, потому что нотация проста и широко используется. Создание предварительной версии бизнес-процессов в виде диаграммы swimlane может стать хорошей отправной точкой дискуссий по выявлению требований.

Диаграмма swimlane

Swimlane-диаграммы являются инструментом, к которому часто прибегают руководители и менеджеры компаний. Эффективность этого вида диаграмм обусловлена рядом причин:

- применение «дорожек» способствует созданию точного представления о работе исполнителей — действиях должностных лиц или структурных подразделений, а также о функциях сотрудников;
- «дорожки» позволяют вовремя обнаружить неверные данные, которые противоречат объективному положению дел в компании;
- с помощью данных схем можно проследить динамику процесса, в частности, выявить ситуации, когда происходит перераспределение обязанностей между сотрудниками;
- они в простой и наглядной форме информируют руководство о задачах, привязанных к конкретной территории;
- использование «дорожек» помогает сократить затраты времени на принятие решений.

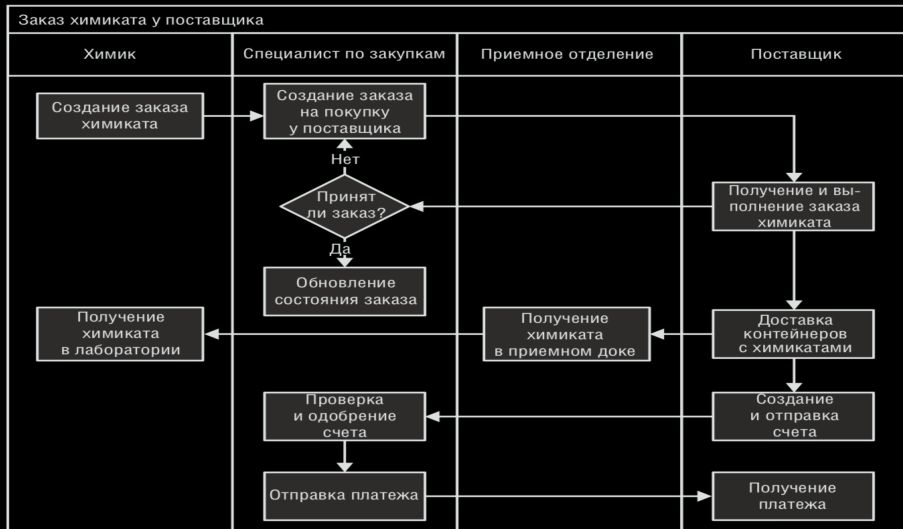
Диаграмма swimlane

Диаграммы swimlane могут содержать дополнительные фигуры, но самые популярные:

- шаги процесса в виде прямоугольников;
- переходы между шагами процесса, показанные как стрелки, соединяющие пары прямоугольников;
- решения — ромбы с несколькими исходящими ветками. Варианты решений показаны в виде текстовых надписей на стрелках, выходящих из ромба;
- дорожки (swimlane), разбивающие процесс, показанные как горизонтальные или вертикальные линии. Дорожки, как правило, отвечают ролям, подразделениям или системам. Они показывают, кто или что выполняет шаги на данной дорожке.

Диаграмма swimlane

Диаграмма swimlane процесса в системе Chemical Tracking System (частичная)



Дополнительные источники

- 1 https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3
- 2 <https://it.wikireading.ru/5086>

Диаграмма переходов состояний и таблица состояний.

Содержание

- Введение.
- Диаграмма переходов состояний.
- Элементы диаграммы переходов состояний.
- Вид диаграмм в зависимости от объекта.
- Пример. Заказ на химикат.
- Описание диаграммы из примера.
- Анализ диаграммы.
- Таблица переходов состояний.
- Взаимодействие таблицы состояний и диаграммы переходов состояний.

Для любых систем ПО необходимо учитывать комбинацию функционального поведения, особенности обработки данных и изменения состояния. Состояний, в которых в каждый момент времени могут находиться системы реального времени и приложения, управляющие процессами, немного. Изменение состояния происходит, только когда удовлетворяются четко определенные критерии, такие как получение определенного сигнала на входе при определенных условиях. Примером может служить перекресток магистрали с установленными датчиками движения, защищенной полосой поворота, а также разметкой и сигналами для пешеходного перехода.

Многие информационные системы имеют дело с бизнес-объектами — заказы на покупку, счета-фактуры, товарно-материальные ценности и. т. п. — для жизненных циклов которых возможно несколько различных состояний, или статусов. При описании сложных изменений состояний на естественном языке высока вероятность упустить из внимания разрешенное изменение состояния или появление запрещенного изменения. В зависимости от структуры спецификации требований к ПО, требования, относящиеся к поведению состояния механизма, могут быть в ней разобщены. В этом случае весьма трудно понять поведение системы.

Диаграмма переходов состояний

Диаграмма переходов состояний (state-transition diagram) позволяет получить лаконичное, полное и недвусмысленное представление о состояниях объекта или системы. Она наглядно показывает возможные переходы между состояниями. Связанный с этой моделью прием — диаграмма состояний — включен в обладающий более богатым набором условных обозначений унифицированный язык моделирования (Unified Modeling Language, UML), который моделирует состояния объекта в течение его жизненного цикла (Ambler, 2005).

Элементы диаграммы переходов состояний

Диаграмма переходов состояний содержит три типа элементов:

- возможные состояния системы — показаны в виде прямоугольников. В некоторых нотациях для представления состояния используются кружки (Beatty и Chen, 2012). Можно использовать кружки или прямоугольники — просто выбрав что-то одно, нужно придерживаться выбранной нотации;
- разрешенные состояния, или переходы, — показаны в виде стрелок, соединяющих пары прямоугольников;
- события или условия, вызывающие каждый переход, — показаны в виде текстовых пояснений для каждой стрелки перехода. Текст может пояснять и событие, и соответствующую реакцию системы.

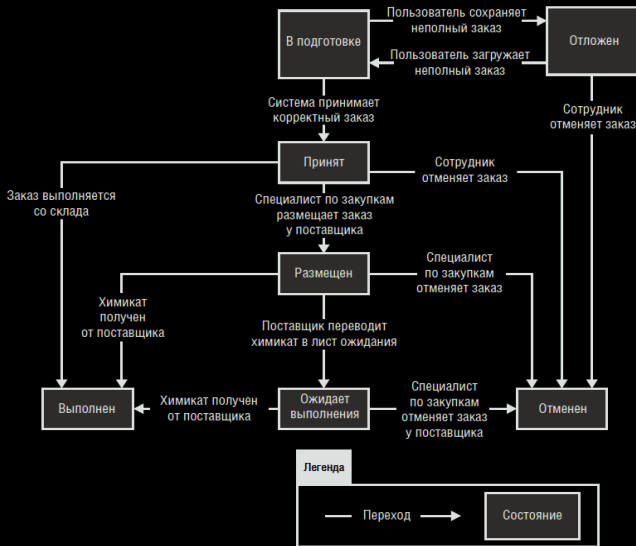
Вид диаграмм в зависимости от объекта

Диаграмма переходов состояний для объекта, имеющего определенный жизненный цикл, будет иметь одно или более конечных состояний, которые представляют конечные состояния, которые могут быть у объекта. У конечных состояний есть входящие стрелки переходов, но нет исходящих. Клиенты могут читать диаграммы переходов состояний после небольшого инструктажа по нотации — ведь это всего лишь прямоугольники и стрелки.

Пример. Заказ на химикат

«Сотрудники, разместившие заказ на химикат», размещают заказы химикатов, которые могут выполняться либо складскими работниками (если химикат есть на складе), либо сторонним поставщиком (для этого ему надо отправить заказ). Каждый заказ проходит через несколько состояний с момента его создания до момента либо его выполнения, либо отмены (два конечных состояния). Таким образом, диаграмма переходов состояний моделирует жизненный цикл заказа химиката.

Пример. Заказ на химикат



Описание диаграммы из примера

Приведенная выше диаграмма показывает, что заказ может находиться в одном из следующих семи состояний:

- В подготовке. Сотрудник создает новый заказ, инициировав эту функцию из другой части системы;
- Отложен. Сотрудник сохраняет частичный заказ для завершения его в будущем, не передавая его в систему и не отменяя операцию заказа;
- Принят. Сотрудник отправляет готовый заказ химиката, и система принимает его к исполнению;
- Размещен. Заказ должен быть удовлетворен сторонним поставщиком, а специалист по закупкам разместил заказ у продавца;

Описание диаграммы из примера

- Выполнен. Заказ удовлетворен: контейнер с химикатом поставлен либо со склада химикатов, либо от поставщика;
- Ожидает выполнения. У продавца не оказалось химиката в наличии, и он уведомил покупателя, что заказ отложен до поставки в будущем;
- Отменен Сотрудник отменил принятый системой заказ до того, как тот был выполнен, или специалист по закупкам отменил заказ у продавца, до того как тот был выполнен или пока ожидал выполнения.

Анализ диаграммы

Когда представители пользователей Chemical Tracking System просмотрели диаграмму переходов состояний для заказа химиката, они определили, что одно состояние не нужно, другое важное состояние отсутствует, и указали два неправильных перехода. При изучении соответствующих функциональных требований эти ошибки никто из них не заметил. В этом и заключается важность представления информации о требованиях на нескольких уровнях абстракции. Диаграмма переходов состояний не дает уровень деталей, достаточный разработчику для создания ПО. Следовательно, в спецификацию требований к ПО для Chemical Tracking System были включены функциональные требования, связанные с обработкой заказа химиката и возможными изменениями его состояния.

Таблица переходов состояний

Таблица состояний показывает все возможные переходы между состояниями в виде матрицы. Бизнес-аналитик может использовать таблицы состояний для проверки, что все переходы обнаружены, путем анализа всех ячеек матрицы. Все состояния выписаны в первом столбце и повторены в первой строке. Ячейки содержат информацию о том, возможны ли переходы из состояния в левом столбце в состояние, указанное в верхней строке, а также указывает событие перехода между состояниями.

Таблица переходов состояний

	В подготовке	Отложен	Принят	Размещен	Ожидает выполнения	Выполнен	Отменен
В подготовке	нет	Пользователь сохраняет неполный заказ	Система принимает корректный заказ	нет	нет	нет	нет
Отложен	Пользователь открывает неполный заказ	нет	нет	нет	нет	нет	нет
Принят	нет	нет	нет	Специалист по закупкам размещает заказ у поставщика	нет	Заказ выполняется со склада	Сотрудник отменяет заказ
Размещен	нет	нет	нет	нет	Поставщик переводит химикат в лист ожидания	chemical received from vendor	Специалист по закупкам отменяет заказ
Ожидает выполнения	нет	нет	нет	нет	нет	chemical received from vendor	Специалист по закупкам отменяет заказ
Выполнен	нет	нет	нет	нет	нет	нет	нет
Отменен	нет	нет	нет	нет	нет	нет	нет

Взаимодействие таблицы состояний и диаграммы переходов состояний

Эта таблица состояний соответствует диаграмме переходов состояний, приведенной выше. Таблица и диаграмма содержат одну и ту же информацию, но табличный формат помогает не пропустить никаких переходов, а формат диаграммы помогает заинтересованным лицам наглядно увидеть возможные последовательности переходов.

Может не требоваться создавать обе модели. Однако создав одну модель, скомпоновать вторую просто, если вам нужно проанализировать изменения состояний с двух точек зрения.

Две строки в таблице, в которых содержатся только значения «Нет», являются конечными состояниями; заказ в состоянии «Выполнен» или «Отменен» не может перейти в другое состояние.

Взаимодействие таблицы состояний и диаграммы переходов состояний

Диаграмма переходов состояний и таблица состояний предоставляют высокоуровневую картину, охватывающую много вариантов использования и пользовательских историй, каждая из которых может выполнять переход от одного состояния к другому.

В моделях состояний не показаны детали обработки, выполняемой системой, а только возможные изменения состояний, возникающие в результате этих процессов. Они помогают разработчику понять ожидаемое поведение системы. Модели способствуют тестированию на ранних этапах, потому что тестировщики могут вывести тесты на основании диаграммы переходов состояний, которая охватывает все допустимые пути переходов.

Обе модели хороши для проверки того, все ли необходимые состояния и переходы состояний корректно и полно описаны в функциональных требованиях.

Карты диалоговых окон. Содержание

- Что такое карты диалоговых окон.
- Особенности.
- Почему стоит изучать карту диалоговых окон? .
- Представление информации.
- Карты диалоговых окон и диаграммы потоков.
- Как упростить карту диалоговых окон? .
- Пример: «Заказ химиката» .
 - ▶ Последовательность действий
 - ▶ Операция удаления химиката

Что такое карты диалоговых окон

Карты диалоговых окон

Карта диалоговых окон представляет дизайн пользовательского интерфейса на высоком уровне абстракции. На ней показаны элементы диалоговых окон в системе и ссылки навигации между ними, но не показан подробный дизайн экрана.

Пользовательский интерфейс можно рассматривать как набор изменений состояний.

Особенности:

- Только один элемент диалогового окна доступен в определенный момент времени для ввода информации пользователем;
- Пользователь может перейти к другим определенным элементам диалогового окна, связанным с выполняемым действием;

Особенности

- Количество возможных путей навигации в сложном графическом интерфейсе велико, но конечно;
- КДО - пользовательский интерфейс, смоделированный в диаграммы переходов между состояниями;
- Карта перемещений - похожий приём с более богатым набором условных обозначений для представления различных типов элементов взаимодействия и контекстных переходов;
- Поток пользовательского интерфейса похож КДО, но показывает пути навигации между окнами пользовательского интерфейса в формате диаграммы swimlane.

Почему стоит изучать карту диалоговых окон?

Карта диалоговых окон

- позволяет рассмотреть возможные концепции пользовательского интерфейса с учетом вашего понимания требований;
- пользователям и разработчикам помогает выработать общее представление того, как пользователь может взаимодействовать с системой для выполнения задачи;
- полезны при моделировании визуальной архитектуры веб-сайта;
- ссылки для перемещений на них изображаются в виде переходов;
- связаны с системными раскадровками, в которые также включено краткое описание назначения каждого экрана.

Карты диалоговых окон отражают сущность взаимодействий системы и пользователя и потоки задач без тормозящих работу команды деталей макета экрана.

С помощью такой карты пользователи могут отследить отсутствующие, неправильные или ненужные переходы и, следовательно, отсутствующие, неправильные или ненужные требования.

Абстрактная концептуальная карта диалоговых окон, разработанная в ходе анализа требований, становится руководством для подробного дизайна пользовательского интерфейса.

Представление информации

В карте диалоговых окон:

- каждый элемент - состояние (прямоугольник),
- каждая допустимая возможность перемещения - переход (стрелка),
- условие, инициирующее перемещение по пользовательскому интерфейсу - текстовый ярлык на стрелке перехода.

Существует несколько типов инициализирующих условий:

- действие пользователя;
- значение данных;
- системное условие;
- некоторые комбинации вышеперечисленных элементов.

Карты диалоговых окон и диаграммы потоков

Карты диалоговых окон слегка напоминают диаграммы потоков, но у них другое назначение.

Диаграммы потока ясно показывают этапы процессы и точки решений, но не пользовательский интерфейс

На карте диалоговых окон не отображается процесс, выполняющийся по линиям перехода, которые соединяют диалоговые окна. Решения ветвления скрыто за окнами, которые показаны на карте диалоговых окон в виде прямоугольников, а условия, в результате которых отображается тот или другой экран, описаны над стрелками переходов

Как упростить карту диалоговых окон

- Пропускать глобальные функции;
- В разделе спецификации требований к ПО, посвященному пользовательскому интерфейсу, должна быть указана доступная функциональность, но демонстрация множества экранов справки на карте диалоговых окон вносит в модель беспорядок;
- При моделировании веб-сайта не нужно включать стандартные для каждой страницы ссылки перемещения;
- Можно опустить переходы, реализующие последовательность перемещений по веб-странице в обратном направлении.

Карта диалоговых окон

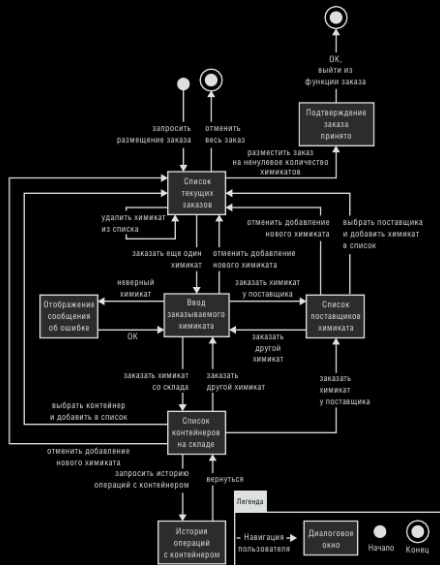
- прекрасный способ представить взаимодействия действующего лица и системы;
- позволяет отобразить альтернативные направления в виде ответвлений от нормального направления.

Где полезны карты диалоговых окон:

- на семинарах по сбору информации, касающейся вариантов использования (при изучении последовательности действий действующих лиц и реакций системы при выполнении задачи)
- для сравнения с готовыми вариантами использования и потоками процессов, чтобы убедиться, что все функции, необходимые для выполнения шагов, доступны при навигации пользовательского интерфейса.

Пример: «Заказ химиката»

- Нормальное направление - заказ контейнера с химикатом со склада.
- Альтернативное — заказ химиката у поставщика.
- Точка входа - «запросить размещение заказа»
- Точки выхода - линии перехода, ведущие в точки «отменить весь заказ» и «ОК, выйти из функции заказа»



Последовательность действий:

- 1 Пользователь: инициирует этот вариант использования, размещая заказ химиката из меню;
- 2 В карте диалоговых окон: действие пользователя - стрелка, направленная к прямоугольнику «Список текущих заказов»
- 3 Прямоугольник - основное рабочее пространство (список химикатов в текущем заказе пользователя);
- 4 Стрелки, исходящие из прямоугольника - все возможные навигации доступные пользователю в данном контексте:
 - ▶ отменить весь заказ;
 - ▶ разместить заказ, если он содержит как минимум один химикат;
 - ▶ добавить в список заказов еще один химикат;
 - ▶ удалить химикат из списка.

Операция удаления химиката

Не участвует какой-либо еще элемент диалогового окна

Обновляется список текущих заказов после того, как пользователь вносит изменения.

Элементы, отражающие остальные элементы варианта использования «Заказ химиката»:

- один путь для заказа химиката у поставщика;
- другой путь для выполнения заказа через склад химикатов;
- еще один возможный путь — просмотр истории операций с контейнером, хранящимся на складе химикатов;
- отображение сообщения об ошибке для обработки ввода неправильного идентификатора химиката или других возможных ошибочных условий.

- Некоторые переходы на карте диалоговых окон позволяют пользователю откатить операцию.
- Карты диалоговых окон позволяют облегчить и упростить работу, предлагая функции отката и отмены в стратегических точках.

Пользователь, изучающий карту диалоговых окон, может обнаружить недостающее требование.

Карта диалоговых окон представляет концептуальные элементы, участвующие во взаимодействии пользователя и системы

Не пытайтесь зафиксировать все детали дизайна пользовательского интерфейса на этапе работы над требованиями.

Лучшее применение — помочь всем заинтересованным лицам проекта выработать общее понимание предполагаемой функциональности системы

Таблицы и деревья решений. Таблицы событий и реакций. Содержание

- Введение.
- Таблицы и деревья решений.
- Влияющие на решение факторы.
- Пример таблицы решений.
- Пример дерева решений.
- Таблицы событий и реакций.
- Три класса системных событий.
- Примеры реакции системы.
- Таблица “событие-реакция”.

Часто система ПО управляется сложной логикой, учитывающей различные комбинации условий, результатом которых является различное поведение системы.

Например, если водитель нажимает кнопку ускорения в системе круиз-контроля автомобиля и автомобиль в данный момент двигается под управлением круиз-контроля, то система увеличит скорость автомобиля, в противном случае команда игнорируется.

Для спецификации требований к ПО необходимы функциональные требования, в которых будет описано, что должна делать система при всех комбинациях условий. Однако условие легко пропустить, в результате чего пропускается и требование. Эти пробелы трудно заметить, просматривая текстовые спецификации.

Таблицы и деревья решений

Таблицы решений и деревья решений — это два альтернативных приема для представления того, что система должна делать, когда в игру вступают сложная логика и решения.

В таблице решений (decision table) перечислены различные значения для всех факторов, влияющих на поведение системы, и приведены ожидаемые действия системы в ответ на каждую комбинацию факторов.

Факторы могут быть показаны либо как утверждения с различными условиями true и false, либо как вопросы с возможными ответами “да” или “нет”. Естественно, возможны таблицы решений с факторами, которые имеют более двух возможных значений.

Влияющие на решение факторы

- есть ли у пользователя, создающего запрос, право заказывать химикаты;
- имеется ли химикат в наличии на складе или у поставщика;
- включен ли химикат в список опасных химикатов, для работы с которыми необходима специальная подготовка;
- есть ли у пользователя, создающего запрос, соответствующая подготовка для работы с этим типом опасного вещества.

Пример таблицы решений

Номер требования					
Условие	1	2	3	4	5
Пользователь авторизован	false	true	true	true	true
Химикат есть в наличии	—	false	true	true	true
Химикат считается опасным	—	—	false	true	true
Сотрудник, разместивший заказ на химикат, прошел соответствующую подготовку	—	—	—	false	true
Действие					
Принять запрос			X		X
Отклонить запрос	X	X		X	

Рис. 12-6. Пример таблицы решений для Chemical Tracking System

Пример дерева решений

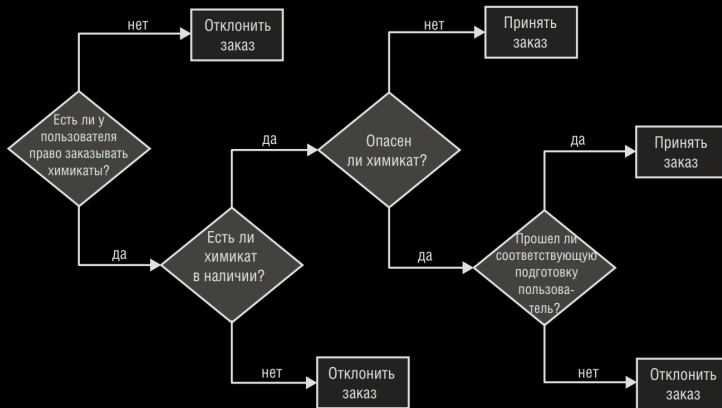


Рис. 12-7. Пример дерева решений для Chemical Tracking System

Таблицы событий и реакций

Варианты использования и пользовательские истории не всегда полезны или достаточны для обнаружения функциональности, которую должен реализовать разработчик. Это особенно верно в отношении систем реального времени.

Есть еще один способ выявления пользовательских требований — определить внешние события, на которые система должна реагировать.

Событием (event) называется какое-либо изменение или действие в среде пользователя, вызывающее реакцию системы ПО.

В таблице “событие-реакция” (event-response table) [ее также называют таблицей событий (event table) или списком событий (event list)] перечислены все такие события и ожидаемое поведение

Три класса системных событий

- Бизнес-событие. Действие пользователя, инициирующее диалог с ПО, например, вызывает вариант использования (иногда это называют бизнес-событием). Последовательность событий и реакций соответствует этапам взаимодействия в данном варианте использования или диаграмме swimlane.
- Сигнал. Это событие регистрируется, когда система получает контрольный сигнал, операцию чтения данных или прерывание, полученное от внешнего аппаратного устройства, например при изменении положения выключателя, изменении напряжения, запросе сервиса другим приложением или касании пользователем сенсорного экрана;
- Временное событие. Это событие инициируется по времени, например когда на часах компьютера наступает определенное время или когда истекает предварительно заданный период времени с момента определенного события.

Примеры реакции системы

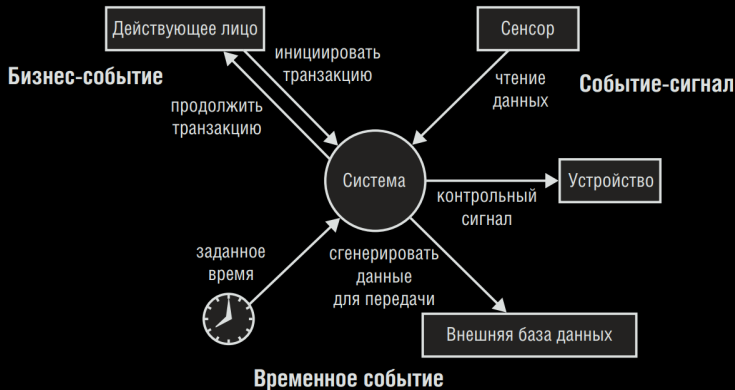


Рис. 12-8. Примеры реакции системы на бизнес-события, сигналы и временные события

Таблица “событие–реакция”

Таблицы “событие — реакция” особенно хороши для управления системами реального времени. Для определения событий возьмите все состояния анализируемого объекта и определите все события, которые могут переводить объект в эти состояния.

Изучите контекстные диаграммы на предмет наличия внешних сущностей, которые могут инициировать действие (инициировать событие) или требуют автоматического ответа (требуется инициирование временного события). В таблицу “событие-реакция” можно добавить другую информацию, в том числе:

- частоту события (сколько раз происходит событие за определенный период времени или предел количества таких событий);
- элементы данных, необходимых для обработки события;
- состояние системы после реакции на событие.

Таблица “событие–реакция”

Табл. 12-3. Таблица «событие–реакция» для автомобильных стеклоочистителей

Иден-тифи-катор	Событие	Состояние системы	Реакция системы
1	Установка системы управления «дворниками» в режим медленной очистки	«Дворники»отключены, выполняют быструю очистку или периоди-ческую очистку	Установка системы управления «дворни-ками» в режим медлен-ной очистки
2	Установка системы управления «двор-никами» в режим быстрой очистки	«Дворники» отключе-ны, выполняют медлен-ную очистку или перио-дическую очистку	Установка системы уп-равления «дворника-ми» в режим быстрой очистки
3	Отключение сис-темы управления «дворниками»	«Дворники» выполняют быструю очистку, мед-ленную очистку или периодическую очистку	1. Завершение текуще-го цикла очистки. 2. Выключение меха-низма «дворников»
4	Установка системы управления «двор-никами» в режим периодической очистки	«Дворники» отключены	1. Выполнение одного цикла очистки. 2. Чтение интервала очистки. 3. Инициализация таймера очистки
5	Установка системы управления «двор-никами» в режим периодической очистки	«Дворники» выполняют быструю очистку или медленную очистку	1. Завершение текуще-го цикла очистки. 2. Чтение интервала очистки. 3. Инициализация тай-мера очистки

Таблица “событие–реакция”

Табл. 12-3. (окончание)

Иден-тифи-катор	Событие	Состояние системы	Реакция системы
6	Истечение интервала таймера после завершения последнего цикла очистки	«Дворники» выполняют периодическую очистку	Выполнение одного цикла медленной очистки
7	Изменение интервала периодической очистки	«Дворники» выполняют периодическую очистку	1. Чтение интервала очистки. 2. Инициализация таймера очистки
8	Изменение интервала периодической очистки	«Дворники» отключены, выполняют быструю очистку или периодическую очистку	Нет реакции
9	Получение сигнала немедленной очистки	«Дворники» отключены	Выполнение одного цикла медленной очистки

Моделирование данных. Диаграмма сущность–связь.

Содержание

- Модель данных
- Диаграмма сущность-связь
 - ▶ Использование диаграммы
 - ▶ Сущности диаграммы
 - ▶ Атрибуты сущностей
 - ▶ Связи сущностей
 - ▶ Мощности связей
 - ▶ Пример диаграммы
- UML-диаграмма классов
- Функциональность системы

Модель данных

Модель данных предоставляет высокоуровневое представление данных системы, а словарь данных дает подробную картину.

Широко используется модель данных **диаграмма «сущность-связь»** (entity-relationship diagrams, ERD) (Robertson и Robertson, 1994).



Диаграмма сущность-связь: Использование

- Если диаграмма «сущность-связь» представляет логические группы информации предметной области и их взаимосвязи, нужно использовать диаграмму «сущность-связь» *в качестве инструмента анализа требований.*
- *Анализ диаграммы «сущность-связь»* помогает понять и связать компоненты данных компании или системы, даже без предположения, что продукт будет включать базу данных.
- При создании диаграммы «сущность-связь» в ходе разработки определяется логическая или физическая *структура базы данных системы.*
- Такое представление реализации расширяет или дополняет понимание системы, которое образовалось в процессе анализа, и оптимизирует ее реализацию, например, в среде реляционной базы данных.

Диаграмма сущность-связь: Использование

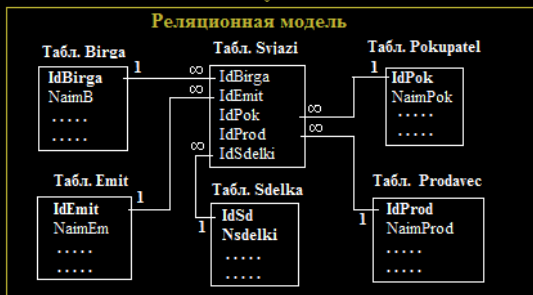
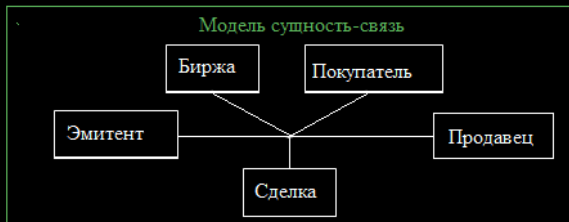


Диаграмма сущность-связь: Сущности

Сущностями (entities) называются физические элементы (включая людей) или агрегации элементов данных, важных для анализируемого бизнеса или для системы, которую планируется создать.

- Сущности именуются посредством *существительных в единственном числе*.
- На диаграмме «сущность–связь» сущности показываются в виде *прямоугольников*.
- В процессе проектирования физической реляционной базы данных сущности обычно становятся *таблицами*.

Диаграмма сущность-связь: Атрибуты сущностей

Каждая сущность описывается несколькими **атрибутами**. У разных экземпляров сущности значения атрибутов могут отличаться. В словаре данных приводятся детальные определения атрибутов — это гарантирует, что объекты на диаграмме «сущность–связь» и соответствующие им хранилища данных на диаграмме потоков данных определены одинаково.

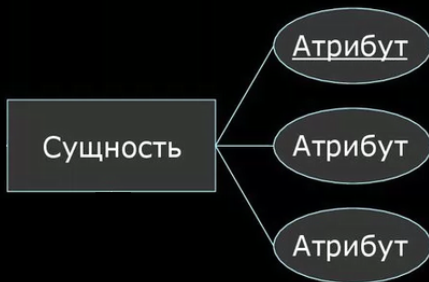


Диаграмма сущность-связь: Связи сущностей

*Ромбы на диаграмме «сущность–связь» обозначают **связи (relationship)**, показывающие логические и числовые связи пар объектов.*

Названия связям даются в соответствии с характером соединений.



Диаграмма сущность-связь: Мощности связей

Мощность (cardinality) каждой связи, или ее численность, показаны цифрами или буквами на линиях, соединяющих сущности и связи. Для диаграмм «сущность–связь» используются различные *нотации* для обозначения мощности.

Виды связей:

- «один ко многим»;
- «один к одному»;
- «многие ко многим»



Диаграмма сущность-связь: Мощности связей

В альтернативных нотациях диаграммы «сущность–связь» на линиях, соединяющих сущности и связи, для обозначения мощности используются другие символы.

В нотации Джеймса Мартина сущности также представлены прямоугольниками, но связи между ними обозначают на соединяющей их линии.

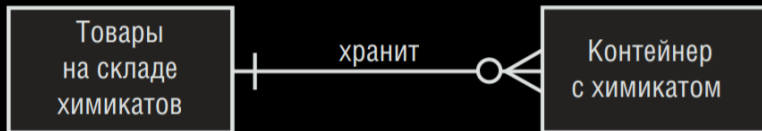
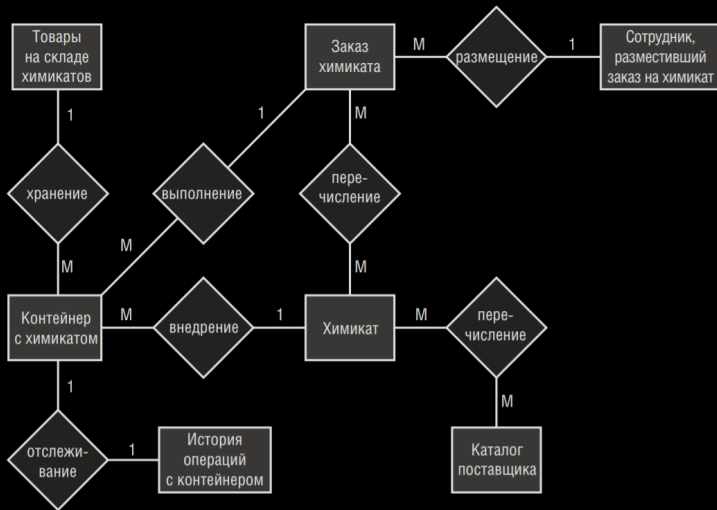


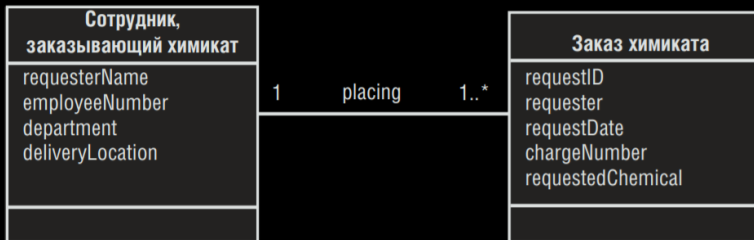
Диаграмма сущность-связь: Пример



UML-диаграмма классов

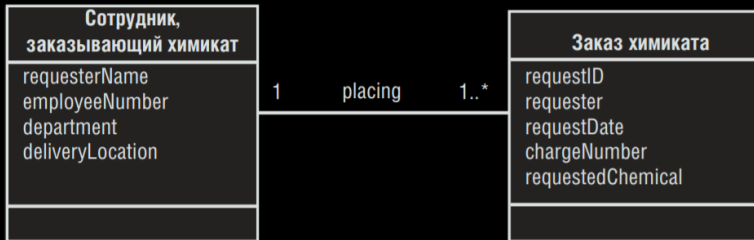
На UML-диаграммах классов показаны:

- **атрибуты** данных классов, что соответствует *сущностям* на диаграмме «сущность—связь»;
- логические **связи** между классами;
- **мощность** связей.



UML-диаграмма классов

- В диаграммах классов может применяться другая нотация для обозначения *мощности* (или множественности).
- На диаграмме классов в средней части прямоугольника также перечисляются *атрибуты класса*.
- Когда диаграммы классов используются для объектно-ориентированного анализа и проектирования, в нижней части прямоугольника обычно показываются *операции*, или *поведения*, которые объекты этого класса должны выполнять.



Функциональность системы

Система должна также включать **функциональность**, которая делает что-то полезное с данными. Отношения между сущностями часто позволяют обнаружить такую функциональность, которую можно зафиксировать в форме варианта использования, пользовательской истории или потока процессов.

При анализе требований проекта с помощью моделей данных могут быть обнаружены ненужные данные, которые были найдены в процессе обсуждения, но потом нигде больше не использовались.

Моделирование данных. Словарь данных. Содержание

- Определение.
- Словарь данных.
- Фрагмент словаря данных для Chemical Tracking System.
- Простейшие элементы данных.
- Структура.
- Повторяющаяся группа.

Определение

- Словарь данных (data dictionary) - набор подробной информации об используемых в приложении сущностях данных.
- Сбор информации о составе, типах данных, разрешенных значениях и т. п. в виде единого ресурса, служащего для определения критериев проверки данных.
- Словарь данных является дополнением к словарю терминов проекта, который определяет термины предметной области или бизнес-термины приложения, сокращения и акронимы.

Словарь данных

- Во время анализа требований информация словаря данных представляет элементы и структуры данных предметной области.
- Время, потраченное на создание словаря данных, будет более чем компенсировано временем, которое экономится путем избегания ошибок по причине того, что участники проекта по-разному понимают ключевые данные.
- При регулярном обновлении словаря данных, он остается ценным средством и при обслуживании системы, и при разработке схожих систем.

Ведение словаря данных — серьезный вклад в повышение качества.

- Использование единообразных определений данных в компании снижает вероятность возникновения ошибок интеграции и интерфейса.
- По сравнению с определениями данных, разбросанными в различных местах функциональных требований, отдельный словарь данных облегчает поиск необходимой информации, а также помогает избежать ненужных повторов и несогласованности.
- Сбор и консолидация информации в словаре обеспечивают целостность реплицируемых структур данных по мере их изменения.

Фрагмент словаря данных для Chemical Tracking System

Элемент данных	Описание	Структура или тип данных	Длина	Значения
Заказ химиката	Заказ нового химиката со склада или у поставщика	Идентификатор заказа + Сотрудник, разместивший заказ на химикат + Дата заказа + Счет затрат + 1:10{Заказанный химикат}		
Пункт назначения поставки	Место, куда нужно доставить заказанный химикат	Строение + Номер лаборатории + Отделение лаборатории		
Количество контейнеров	Количество заказываемых контейнеров определенного размера с химикатом	Положительное целое	3	
Емкость	Объем химиката в заказываемом контейнере	Целое	6	

Фрагмент словаря данных для Chemical Tracking System

Элемент данных	Описание	Структура или тип данных	Длина	Значения
Единица измерения	Единицы измерения, в которых указывается количество заказываемого химиката	Буквенные символы	10	Граммы, килограммы, миллиграммы и т. п.
Идентификатор заказа	Уникальный идентификатор заказа	Целое	8	Генерируемый системой порядковый номер, начиная с 1
Заказанный химикат	Описание заказываемого химиката	Идентификатор химиката + Количество контейнеров + Класс качества + Емкость + Единица измерения + (Поставщик)		

Фрагмент словаря данных для Chemical Tracking System

Элемент данных	Описание	Структура или тип данных	Длина	Значения
Сотрудник, разместивший заказ на химикат	Информация о сотруднике, заказывающем химикат	Имя сотрудника, заказывающего химикат + Личный номер сотрудника + Отдел + Пункт назначения поставки		
Имя сотрудника, заказывающего химикат	Имя сотрудника, заказывающего химикат	Буквенные символы	40	Может содержать пробелы, дефисы, точки и апострофы

Простейшие элементы данных

Простейшим элементом данных называется тот, дальнейшая декомпозиция или упрощение которого невозможно или ненужно. К определенным в фрагменте словаря простейшим элементам данных относятся:

- «Количество контейнеров»;
- «Емкость»;
- «Единица измерения»;
- «Идентификатор заказа»;
- «Имя сотрудника, заказавшего химикат»;

Другие столбцы словаря данных можно использовать для описания относящихся к простейшему элементу данных типу, длине, диапазону числовых значений, списку разрешенных значений (как для «Единицы количества») и других уместных атрибутов.

Структура

Структура данных (или запись) содержит несколько элементов данных. В фрагменте словаря показаны следующие структуры данных:

- «Запрос химиката»;
- «Пункт назначения поставки»;
- «Заказанный химикат»;
- «Сотрудник, разместивший заказ на химикат».

Столбец «Структура или тип данных» в словаре данных — место, где перечисляются элементы, из которых состоит структура, а элементы отделяются знаком «плюс» (+).

Структура

Структуры могут содержать другие структуры:

- Структура «Сотрудник, разместивший заказ на химикат» включает структуру «Пункт назначения поставки».

Каждый элемент данных в структуре должен быть определен в словаре данных.

Если элемент в структуре данных необязателен, он заключается в скобки.

- В структуре «Заказанный химикат» элемент данных «Поставщик» является необязательным, поскольку сотруднику, разместившему запрос, может быть безразлично или он может просто не знать, кто именно поставляет нужный химикат.

В таком макете словаря данных полезны гиперссылки.

- В качестве примера, элемент «Емкость» в структуре данных «Заказанный химикат» может иметь гиперссылку. Читателю достаточно будет просто щелкнуть эту ссылку, чтобы перейти к определению элемента «Емкость» в другом месте словаря данных.

Ссылки навигации очень полезны в объемном словаре данных, который может состоять из многих страниц или даже нескольких документов, если словарь данных проекта включает часть определений из общекорпоративного словаря данных.

Повторяющаяся группа

Если в структуре данных содержится несколько экземпляров элемента данных, то этот элемент необходимо заключить в фигурные скобки и показать допустимое количество возможных повторов в формате минимум : максимум перед открывающей скобкой.

- В фрагменте словаря элемент «Заказанный химикат» в структуре «Заказ химиката» является повторяющейся группой, которая оформляется так: 1:10 {Заказанный химикат}. Это означает, что заказ химиката должен содержать как минимум один, но не более десяти химикатов.

Если максимальное количество экземпляров в повторяющемся поле неограниченно, для указания этого факта используется «n».

Повторяющаяся группа

Точное определение элементов данных сложнее, чем может показаться, например, для типа данных, как «Буквенные символы», указанного для элемента «Имя сотрудника, заказывающего химикат»:

- Чувствительно ли имя к регистру, то есть «Карл» и «карл» — это одно или разные имена?
- Должна ли система приводить весь текст к верхнему или нижнему регистру?
- Могут ли использоваться другие символы помимо 26 букв английского алфавита, такие как пробелы, дефисы, точки или апострофы — все они могут присутствовать в именах?

Такие точные определения очень важны для разработчика, чтобы точно понимать, как проверять вводимые данные.

Спасибо за внимание.