addconf.ru

CUSTIS®

Application Developer Days

Конференция программистов

29-30 АПРЕЛЯ 2011. Санкт-Петербург

Опыт CUSTIS

Необъектные модели предметной области

Докладчик:

Максим Цепков (M.Tsepkov@custis.ru)
www.CUSTIS.ru

CUSTIS®

Application Developer Days

О чем будет доклад?

DDD – эффективный способ проектирования

Объектная модель предметной области – понятно ☺

Необъектная модель

- Зачем?
- И что это такое?
- Чем они полезны?

Рассмотрим на примерах…

CUSTIS®

Application Developer Days

Немного истории

Объектная модель – не первая

– Реляционная модель – вместе с базами данных

– Функциональная модель (Lisp)

– Исчисление предикатов (Prolog)

Объектная модель

– проработана в начале 90-х

– является основной в настоящее время

Появляются новые и композитные модели

– Модель взаимодействующих лиц с обменом сообщениями (Erlang)

– Модель многомерных показателей (гиперкубов)

– В объектную модель добавляют другие парадигмы (LINQ)

– И так далее, идет активное развитие…

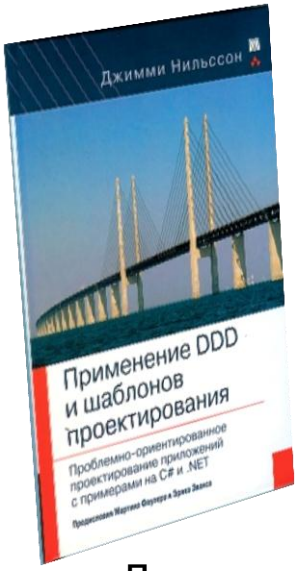
DDD – эффективный способ проектирования

Что такое DDD

Концептуальная книга Эрика Эванса

– на английском – в 2003 г.

– на русском – только в 2010 г.



Практическая книга Джимми Нильссона

– на английском – в 2006 г.

– на русском – в 2007 г. (почти сразу!)

Для знакомства – можно смотреть материалы, ссылки, слайды и видеозапись тренинга Андрея Бибичева: http://lib.custis.ru/ddd-training

CUSTIS®

Application Developer Days

DDD – проектирование по модели

Строим модель предметной области,
вырабатываем для этого единый язык

Воспроизводим модель в архитектуре программы
и коде – соответствие должно быть очевидным

Модель предметной области
становится моделью системы

CUSTIS®

Application Developer Days

Требования к модели

Заказчик владеет единым языком
и понимает модель без перевода

Разработчик может реализовать модель
в коде без дополнительного проектирования

Модель можно проследить в бизнесе и в коде

P.S. Аналитик тоже есть – он строит модель

Парадигмы построения моделей

CUSTIS®

Application Developer Days

Что есть парадигма моделирования на примере объектной модели…

Элементы единого языка для предметной области
и способ их соединения в сложные конструкции

Объекты с атрибутами и методами

Визуальный образ для эффективного представления

Диаграмма классов и другие диаграммы UML

Способ отражения модели в реализацию

Объекты в программе

CUSTIS®

Application Developer Days

Достоинства объектной модели

- ☺ Соответствует парадигме современных языков
- ☺ Понятна разработчикам и аналитикам
- ☺ Имеет эффективные визуальное представление
- ☺ Соответствует реальному миру и понимается заказчиком – если проектировать бизнес-объекты

Недостатки объектной модели

По опыту разработки корпоративных приложений

☹ Плохо представляет цикл жизни объекта

☹ Плохо подходит для отражения потоков ресурсов

☹ Плохо подходит для систем связанных показателей

Если для области автоматизации эти аспекты важны, можно применять другие парадигмы

CUSTIS®

Application Developer Days

Как отражать модель в реализацию

Язык, реализующий парадигму

Framework, часто с диаграммами

например, MS Workflow Foundation
для реализации документооборота

DSL, лучше графический,

с компилятором или интерпретатором

Диаграммы и понятия единого языка
и шаблоны их отражения в реализацию

Для сложных областей

Если нет готового – приходится разрабатывать

CUSTIS®

Application Developer Days

Как сделать необъектную модель?

Выбрать или придумать парадигму моделирования

Объекты обмениваются сообщениями

Ресурс выделяется действующим лицам

Определить визуальный образ единого языка

Диаграммы взаимодействия и синхронизации

Образ разрезания пиццы

Разработать правила отражения модели в код – иначе элементы модели не найти в реализации

Лучше, если отражение будет по шаблонам

Документооборот и State Entity

CUSTIS®

Application Developer Days

Бизнес-задача – обобщенный документооборот

Документ имеет несколько этапов обработки

На каждом этапе определенные сотрудники могут совершать определенные действия

Для передачи на следующий этап должны выполняться определенные условия

CUSTIS®

Application Developer Days

Идея решения

Документу приписываем **состояние**

Состояние определяет этап документооборота:

– какие действия можно совершать над документом

– кто отвечает за обработку документ

– кто имеет права на совершение тех или иных действий

Возможные изменения состояний документа образуют граф переходов

CUSTIS®

Application Developer Days

Язык модели

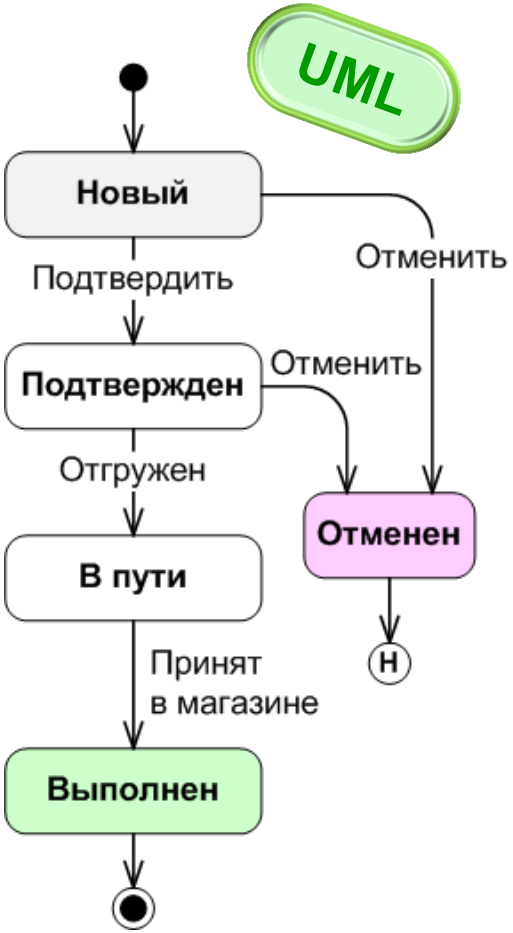
Структура документов –
объектная модель

Действие над документом –
вызов метода

Состояния документа и
методы- переходы между ними,

Граф состояний – State machine diagram

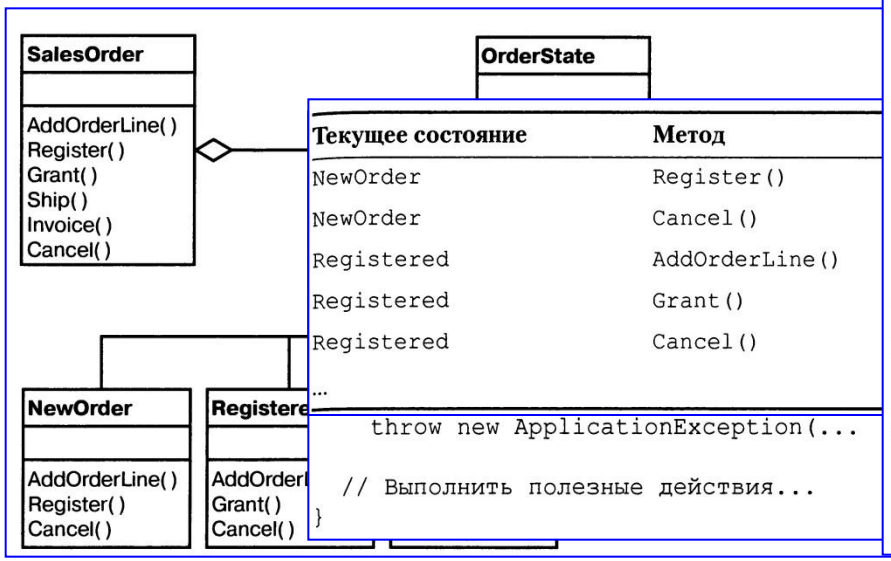
Названия состояний и переходов –
на языке бизнеса



Варианты шаблона реализации

Из книги Нильссона

1. Императивно в коде конкретных методов
2. Императивно в едином методе смены состояния
3. Декларативно – через таблицу переходов и состо
4. Через иерархию классов-состояний




```
private void _ChangeState(OrderState newState)
{
  if (newState == _currentState)
    retun; // Не считать ошибкой переход состояния

  switch (_currentState)
  {
    case OrderState.NewOrder:
      swith (newState)
      {
        case OrderState.Registered:
        case OrderState.Cancelled:
          _currentState = newState;
          break;

        default:
          throw new ApplicationException(...
          break;
      }
    case OrderState.Registered:
      switch (newState)
      {
        case OrderState.NewOrder:
        case OrderState.Granted:
        case OrderState.Cancelled:
          _currentState = newState;
          break;

        default:
          throw new ApplicationException(...
          break;
      }
...
// И так далее...
  }
}
```

Теперь метод AddOrderLine() выглядит следующим об

```
public void AddOrderLine(OrderLine orderLine)
{
    _ChangeState(OrderState.NewOrder);

    // Выполнить полезные действия...
}
```

Что выбрать?

Модель должна прозрачно отражаться в реализацию

Используем декларативное описание (3)

Или комбинацию (1) и (3):

– императивно изменяем состояние в методе перехода

– контролируем, что изменение соответствует декларативной разметке

Таблица метаданных – декларативное описание – однозначно соответствует диаграмме состояний

Почему не иерархия состояний?

Application Developer Days

Иерархия классов-состояний – в объектной модели

– Полнее выразить реализацию в диаграмме классов

– Но поведение документа – за рамками, оно только в графе переходов

– Поэтому для единого языка, понимаемого заказчиком – не подходит

Реализаци

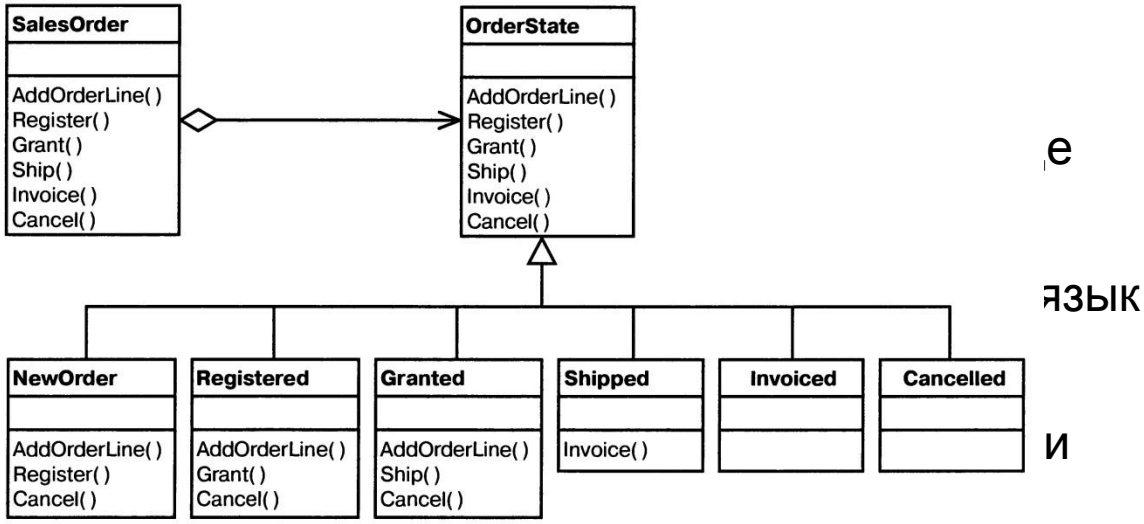
– Таблица п … е

– И однозна …

– Диаграмм… …язык

– При этом … …

– Однако, ре… …и



Сравним модели…

Модель

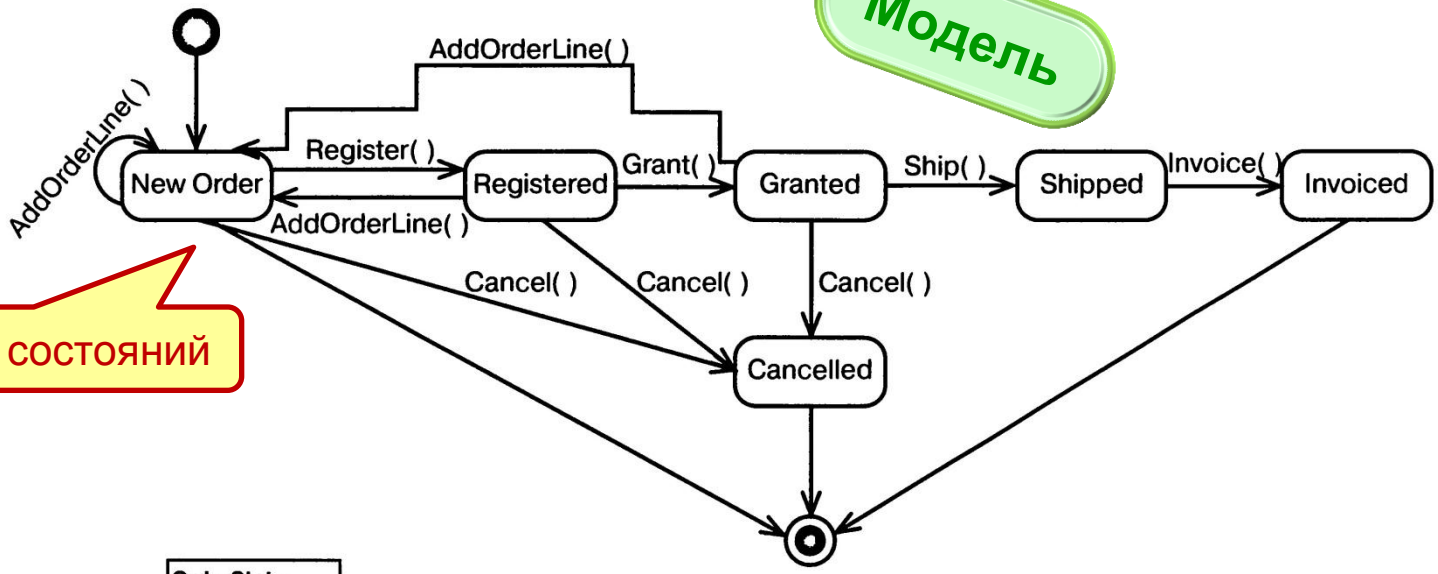


Диаграмма состояний

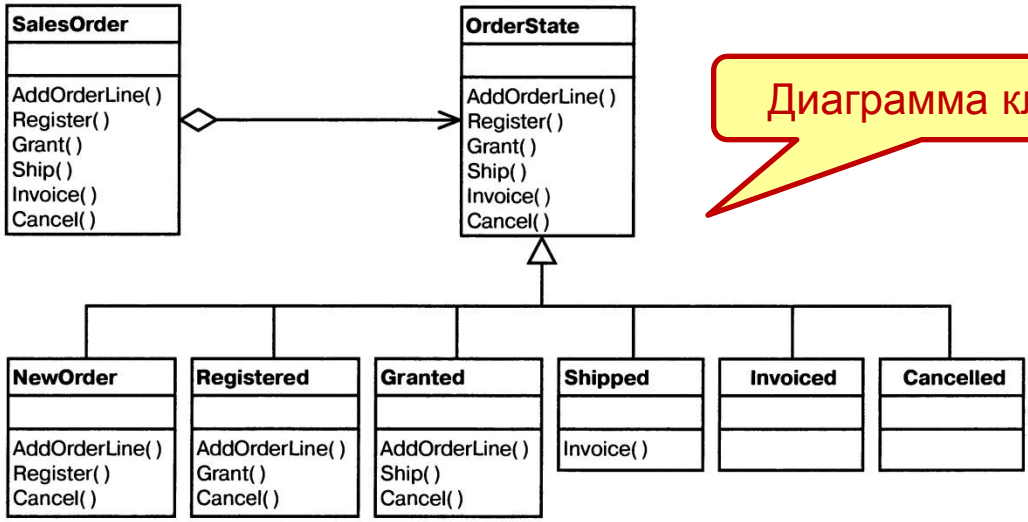


Диаграмма классов

Это – лишнее

CUSTIS®

Application Developer Days

Реализация на разных платформах

Отдельный проект

– Инициализация таблицы переходов для каждого документа

– Общая функция проверки перехода для всех методов (вместе с логом)

– Таблица допустимых действий для каждого состояния (тоже с логом)

Собственный объектный framework в Oracle

– Таблица переходов и прав в метаданных

– Вызов процедур-методов динамически с проверками

Реализация на разных платформах

Собственный ORM на C#

– Разметка методов метаданными – состояния, права
– Описание в метаданных графа состояний и условий
– Обработка на посткомпиляции при создании реализации

2008

Об описании графа и условий доклад Гребнева и Алексеева на ADD-2010
http://lib.custis.ru/217-Static-checking-with-DDD-add-2010

```
[Method(AutoSave = true)]
[StateRestriction(RequestForShipmentState.New)]
[StateTransition(RequestForShipmentState.New, RequestForShipmentState.Created)]
[GrantInvocation(RmsRole.Manager)]
public virtual void PrepareForShipment()
{
    State = RequestForShipmentState.Created;
    ...
```

Учет
модель и реализация

Бизнес-задача: учет остатков и потоков товаров, денег, других ресурсов

Исполнение документов изменяет учетные показатели – остатки и обороты в разрезе *аналитик* (товаров, клиентов)

Учетные показатели влияют на обработку документов и решения пользователей

Нужно в большинстве управленческих систем, а не только в бухгалтерии

Примеры показателей

– Остаток на складе по ответственным
– Поступление товара за период
– Поставка в магазины за месяц

Товар	Было	Пришло	Ушло	Стало
Куртка К12-S	122	40	75	87
Куртка К15-М	187	50	90	147
…				

CUSTIS®

Application Developer Days

Учетная модель – не объектная

Сложность объектного представления учета

– Нет идентификации единичного объекта
– Работа идет с показателями, текущее значение которых меняется
– Изменение числового значения может менять состояние с точки зрения принятия бизнес-решения
– Часто интерес представляют агрегаты, а не отдельные значения

Представление учета оказалось за рамками UML

И вообще эффективного представления ☹

CUSTIS®

Application Developer Days

Что входит в учетную модель

Элементы учета

– Синтетические счета и их аналитика

– Проводки

– Показатели – остатки и обороты

Для представления учетной модели мы придумали Диаграммы учета

Диаграммы показывают

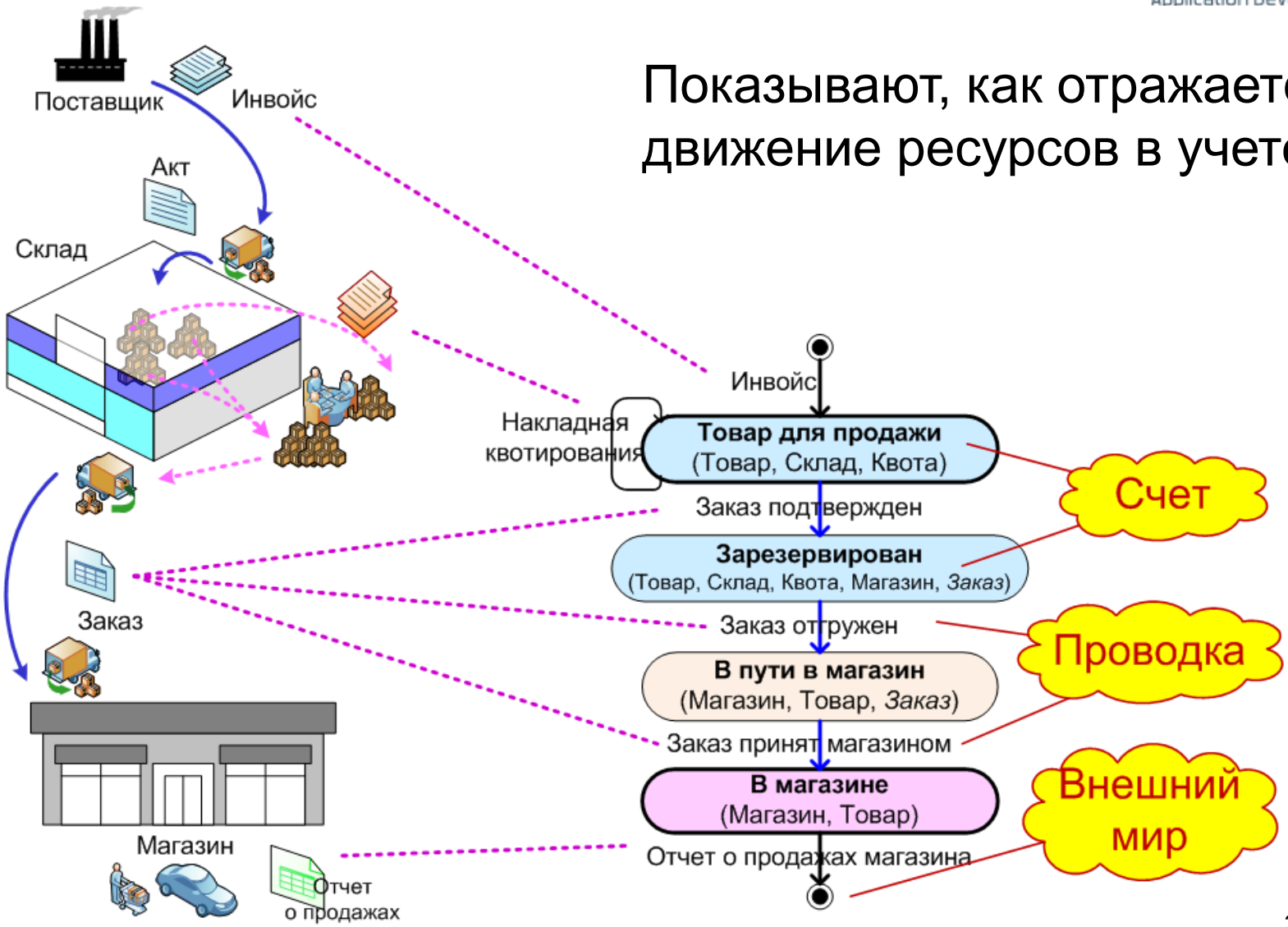
– как проводки перемещают ресурсы по синтетическим счетам

– какая аналитика счетов

– по каким переходам исполняются проводки

Диаграммы учета

Показывают, как отражается движение ресурсов в учете



CUSTIS®

Application Developer Days

Подробно о диаграммах учета

К сожалению, подробного описания нет ☹

Есть выступления на конференциях

ЛАФ-2010 – http://lib.custis.ru/Accounting-diagrams

Презентация и видео

«Диаграммы планов счетов –
средство моделирования и проектирования учета»

Презентация и статья

SECR-2010 – http://lib.custis.ru/Simplify-security-accounting

«Учет ценных бумаг – сделать сложное простым»

CUSTIS®

Application Developer Days

Способ реализации учета

Есть Patterns for Accounting Мартина Фаулера – отражение учета в объектную реализацию

– учетные счета и проводки

– источник проводок – события

У нас – более развитая реализация

– хранение аналитических признаков на счетах и проводках

– ведение остатков и оборотов учетных счетов

– ведение детальных и агрегированных показателей

Есть собственный язык описания – GL-XML

CUSTIS®

Application Developer Days

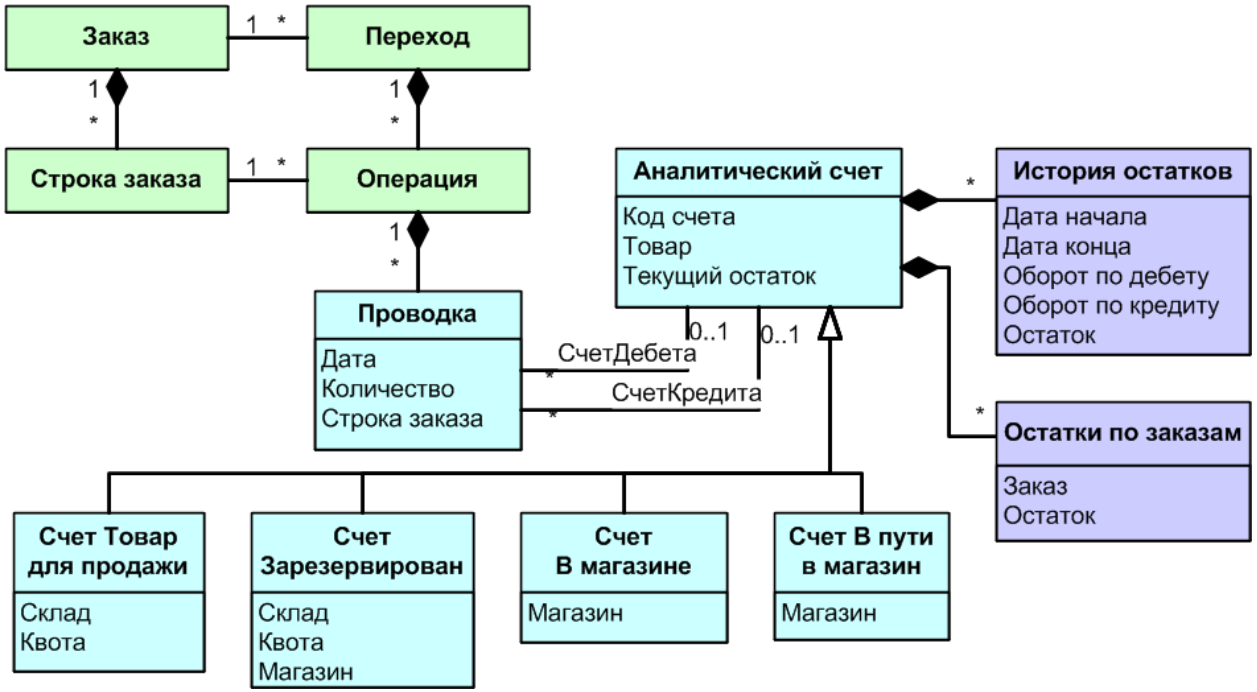
Реализация учета – классы

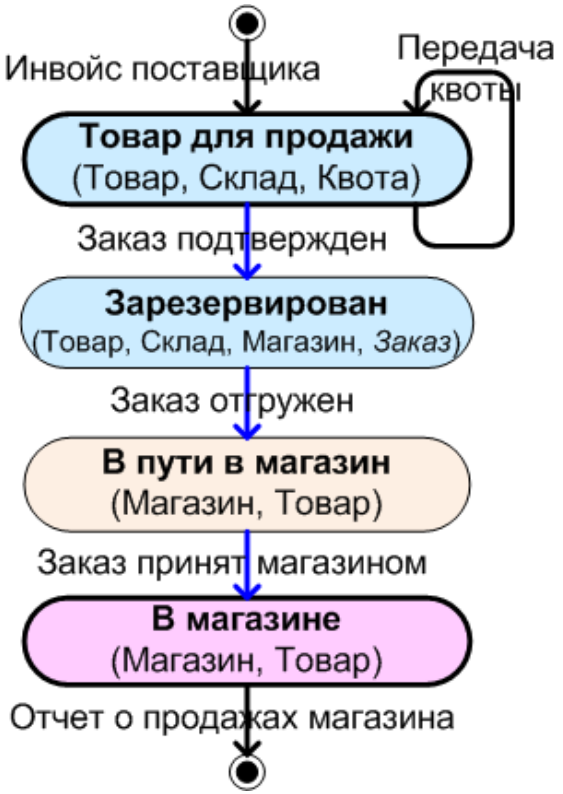
Реализация учета выполняется по шаблону

Классы и таблицы имеют стандартную структуру

Она порождается по описанию на GL-XML

Диаграмма классов – не интересна





CUSTIS®

Application Developer Days

Реализация учета разработчиком

Сделать реализацию по диаграмме учета – описать набор счетов, аналитику счетов и проводок

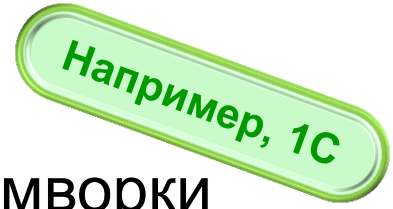
Описать хранимые показатели

Обеспечить создание проводок на учетных событиях – переходах документов, императивно или декларативно

CUSTIS®

Application Developer Days

А специальные учетные системы?

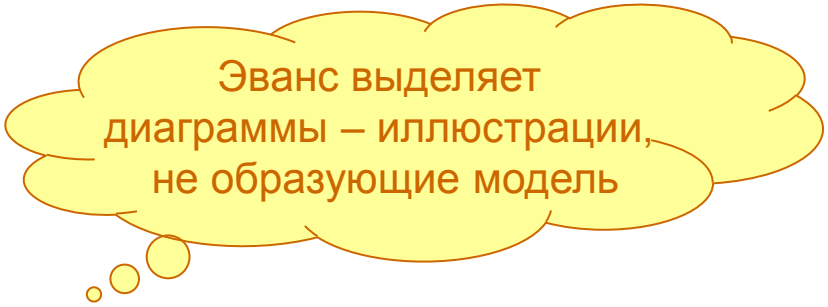


Специальные учетные системы – это фреймворки

Счета, аналитики настраиваются (декларативно)

Проводки – шаблонами или на встроенном языке

Можно строить модель учета,
используя диаграммы учета,
меняется лишь отражение в реализацию



Модель или иллюстрация?

CUSTIS®

Application Developer Days

Диаграмма – не всегда модель

В процессе проектирования часто используют диаграммы, описывающие предметную область

Когда диаграммы становятся моделью?

Они входят в единый язык, их понимают разработчики

Их можно сопоставить с реализацией, то есть кодом

Проектирование реализации по диаграмме носит технический характер, часто это применение шаблона

Иначе это не модель, а иллюстративные диаграммы

ЗАКЛЮЧЕНИЕ
(что я хотел сказать)

CUSTIS®

Application Developer Days

Модель – основа взаимопонимания

Используйте модели, понятные заказчику

Шаблоны – способ перевода с языка модели в код

Необъектные модели – эффективно, хотя непривычно

Программисты – умные и поймут незнакомые модели!

Спасибо!

Вопросы?

Максим Цепков (M.Tsepkov@custis.ru)