CUSTIS®

Три точки опоры в архитектуре корпоративных систем

Докладчик:

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

CUSTIS®

Корпоративные приложения

Заказная разработка сложных систем

– Поддержка уникальных бизнес-процессов заказчика

– Частые изменения системы – вместе с изменениями бизнеса

– Постановки необходимо согласовывать с бизнес-пользователями

Большие долгоживущие системы

– Начальная разработка – 5–20 человеко-лет за 0,5–2 года

– Активное развитие весь срок эксплуатации – 5–10 лет (бывает дольше)

– Существенно меняется коллектив разработчиков и пользователей

Учетно-аналитические системы

– Учет – одна из основных функций системы

– Учетные показатели – основа бизнес-логики и принятия решений

– Связь учета с документооборотом должна быть прозрачна

CUSTIS®

Что нужно для проектирования?

Понятное описание системы

– Заказчик владеет языком описания и верифицирует постановки

– Разработчики тоже свободно общаются на языке описания

Прозрачное и ясное описание

– Компактно представляет сложные конструкции системы

– Соответствует реализации (программному коду)

– Связывается с тем, что видит пользователь в системе

Проект системы развивается вместе с системой

– На языке описания можно формулировать запросы на изменение

– Можно оценить трудоемкость реализации доработок

Ошибки в постановках – самый большой риск

Понятное и прозрачное описание его снижает

CUSTIS®

Как мы проектируем?

Мы строим модель на языке предметной области

- Сначала это – модель самой предметной области
- Но очень быстро превращается в модель системы
- Оставаясь на языке предметной области

Модель является архитектурой системы

Модель итеративно развивается вместе с системой

Применяем диаграммы и другие методы компактного представления сложных систем, разработанные в IT

CUSTIS®

Наш способ – в мировом тренде

Domain Driven Design – mainstream в проектировании

Архитектура на основе **модели предметной области**

Единый язык общения заказчиков и разработчиков

Итеративное проектирование и разработка

CUSTIS

Подробнее о 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®

Учетная машина – DDD для корпоративного приложения

Мы не просто применяем DDD при проектировании

У нас сложился шаблон модели – **Учетная машина**

Мы применяем его для учетных корпоративных приложений предприятий и банков

CUSTIS®

Шаблон учетной машины

CUSTIS®

Обобщенное учетное приложение

Пользователи создают документы

По необходимости заполняют справочники

Потом документы исполняют

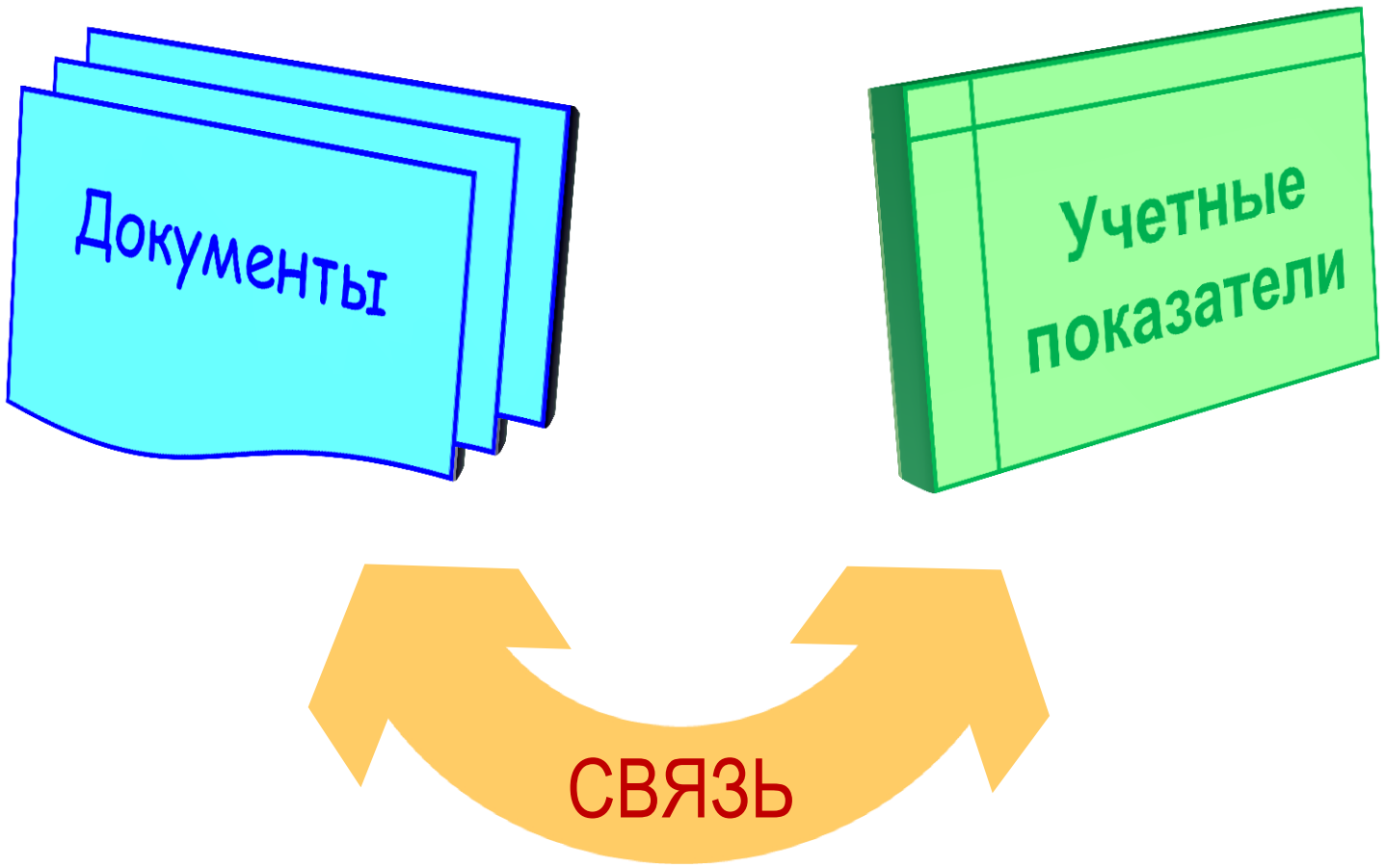
При этом меняются учетные данные

Которые влияют на исполнение документов

И отражаются в отчетах

CUSTIS®

Состав учетного приложения



CUSTIS®

Что такое учетные показатели?

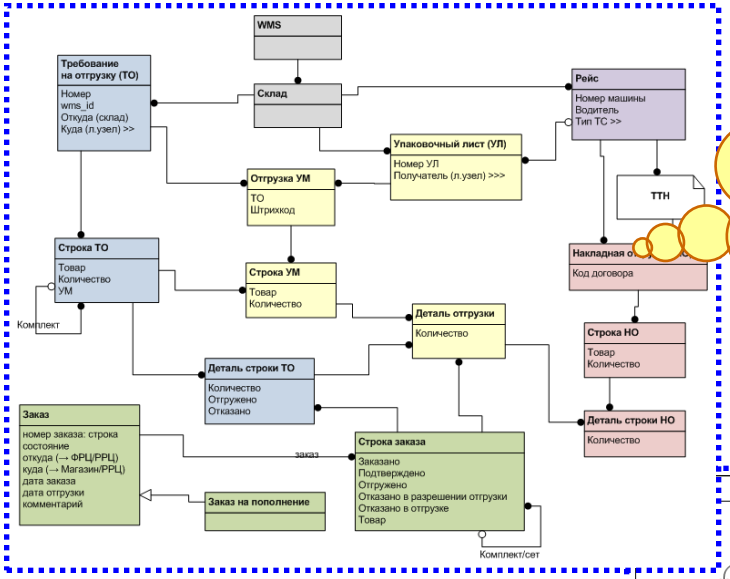
Учетные показатели –
отражение **реальных** потоков
обобщенных ресурсов
(товаров, денег, долгов и др.)

Учетные показатели нужны и важны
в большинстве управленческих систем,
а не только в бухгалтерии

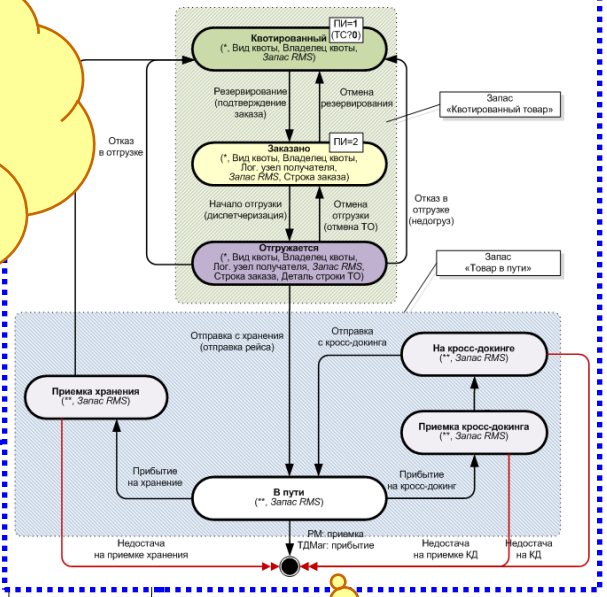
Бухгалтерия же отражает реальные
потоки весьма опосредованно

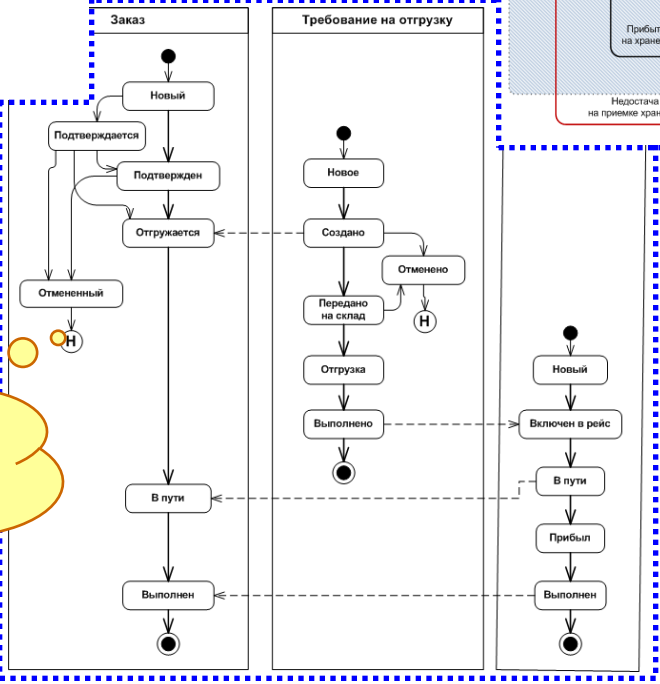
CUSTIS

Представление Учетной машины



Документы и справочники – диаграмма классов





Документооборот – диаграмма состояний

Учет – диаграмма учета

CUSTIS®

Как представить учет?

Учет – основное назначение учетной системы

Представление учета оказалось за рамками UML

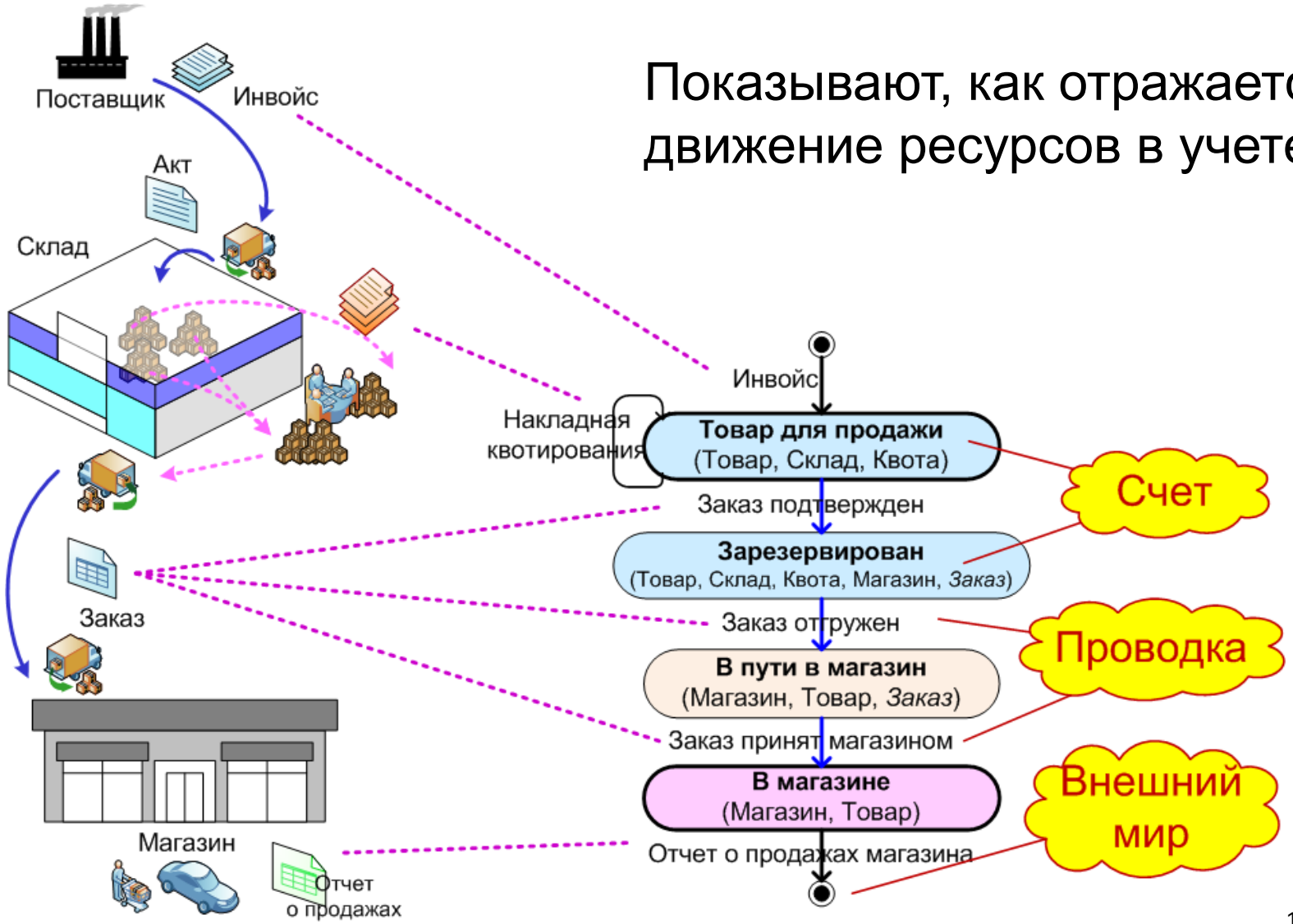
И вообще эффективного представления ☹

Мы придумали Диаграммы учета

Они дают целостную учетную модель системы

CUSTIS

Диаграммы учета



Показывают, как отражается движение ресурсов в учете

CUSTIS®

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»

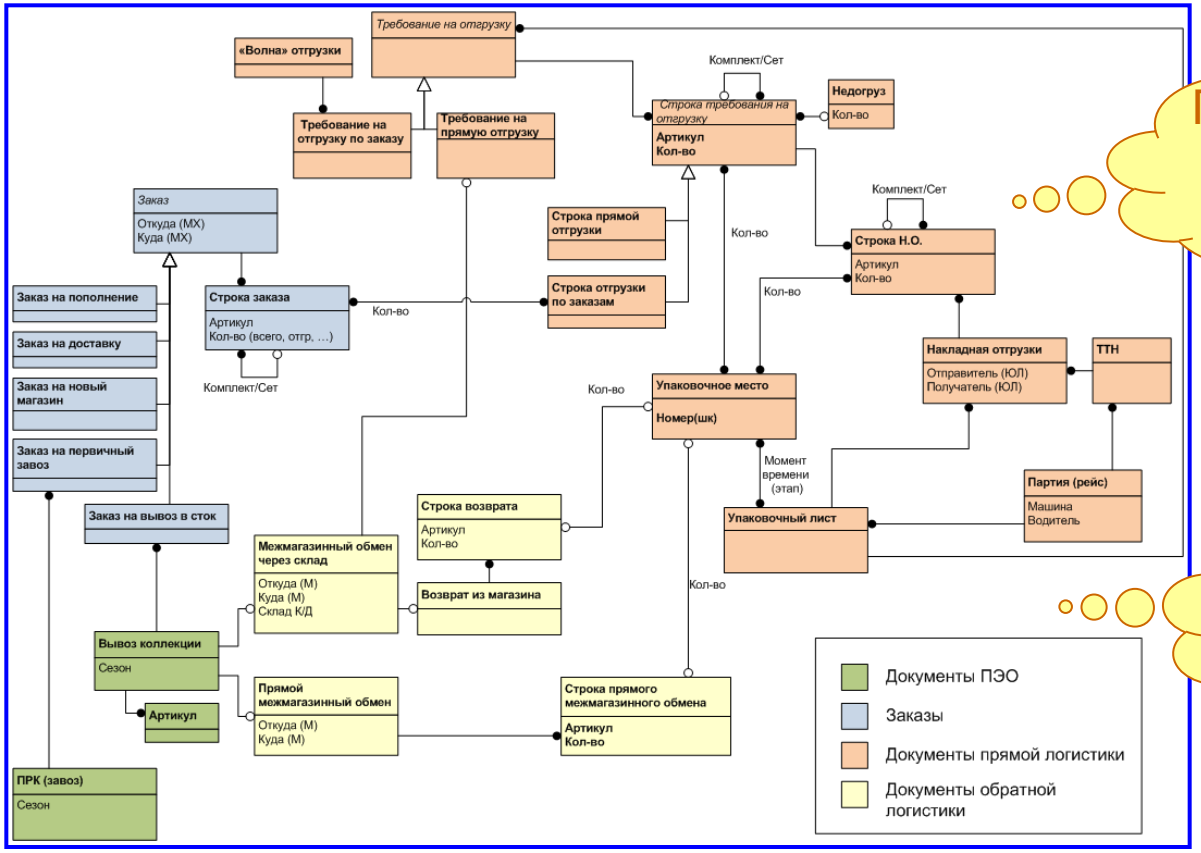
CUSTIS

Документы (и справочники)

Используем объектно-ориентированный подход

Классы соответствуют бизнес-объектам – заказчик видит знакомые названия

Единый язык



Представляем диаграммой классов

Используем цветовые выделения

CUSTIS®

Связь документов с учетом

При обработке и проведении документов возникают хозяйственные операции, отражаемые в учете

Документу приписываем **состояние**
Оно определяет отражение документа в учете и другие аспекты документооборота:

– текущий этап обработки документа

– ответственность за документ

– права на совершение действий

CUSTIS®

Диаграмма переходов документа

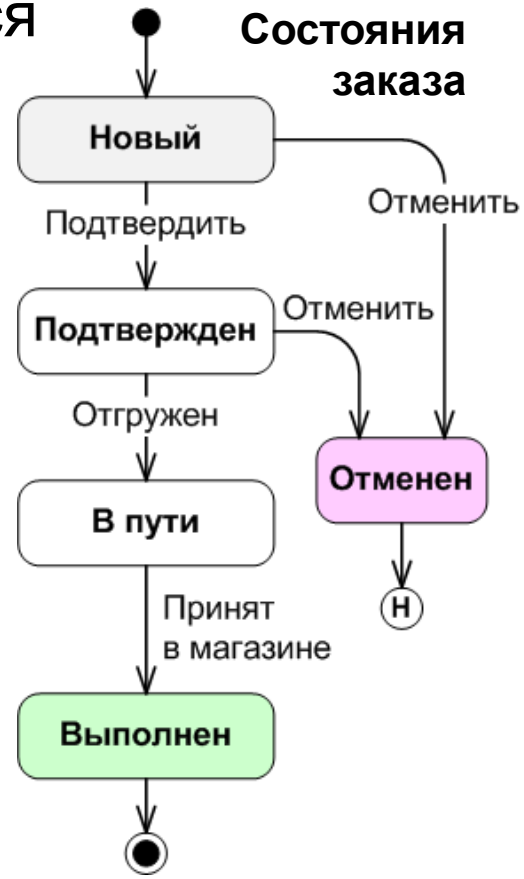
Граф состояний документа изображается на State Machine Diagram (UML)

– узлы – состояния

– ребра – методы-переходы

Состояния и переходы называем в терминах бизнеса

Единый язык



CUSTIS®

Итак, единая модель системы

Модель системы представляется в проекциях

– диаграмма классов – структура объектов и их методы
– диаграмма учета – учетная модель системы
– диаграмма состояний документов – документооборот

} В терминах предметной области

Все проекции связаны между собой через методы-переходы, которые есть на всех диаграммах

Эта модель – результат обобщения нашего опыта разработки корпоративных приложений

В большинстве случаев проекций модели достаточно

CUSTIS®

Развитие модели в ходе проекта

CUSTIS®

От модели предметной области – к модели системы

Начало проекта (1–2 месяца)

– Интерактив с заказчиком, анализ предметной области
– Диаграмма классов и диаграмма учета в крупном
– План поэтапной реализации по фрагментам предметной области

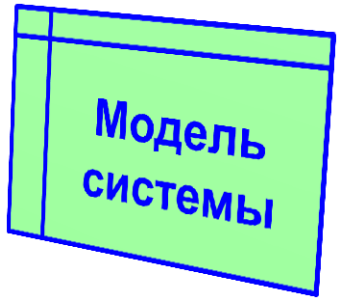
Итеративная разработка (6–18 месяцев)

По фрагментам

– Детальное проектирование фрагмента системы
– Уточнение структур данных и учета, проработка документооборота
– Реализация спроектированного фрагмента

Внедрение и эксплуатация (5–10 лет)

– Реализация бизнес-процессов в рамках модели системы
– Запросы на развитие системы в терминах модели



CUSTIS

Развитие объектной модели

Сначала основные классы без деталей

Подробности – по ходу разработки

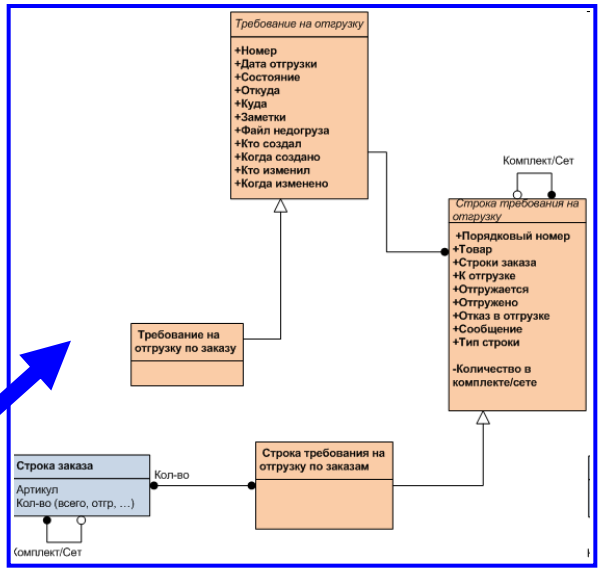
– Уточняем иерархию типов

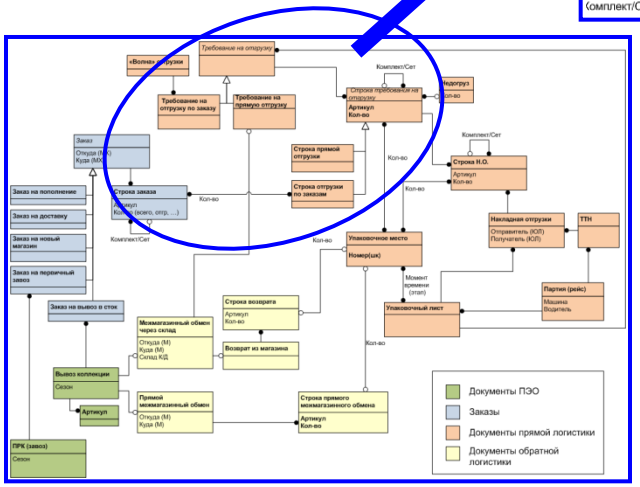
– Определяем атрибуты и методы

– Проектируем вспомогательные классы

– Уточняем структуру ассоциаций между классами

Не все делаем на первом этапе





CUSTIS®

Развитие учета

Уточнения

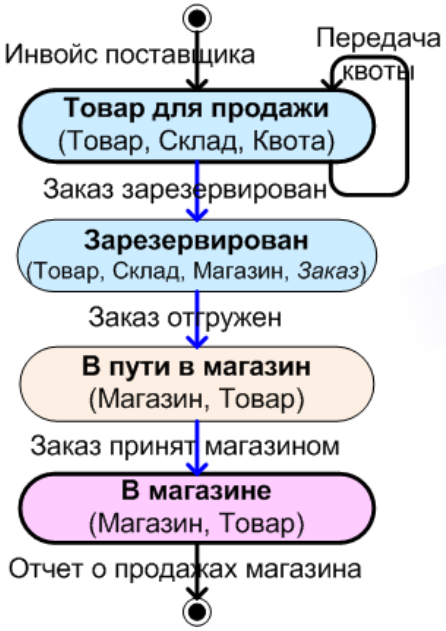
– определяем аналитики

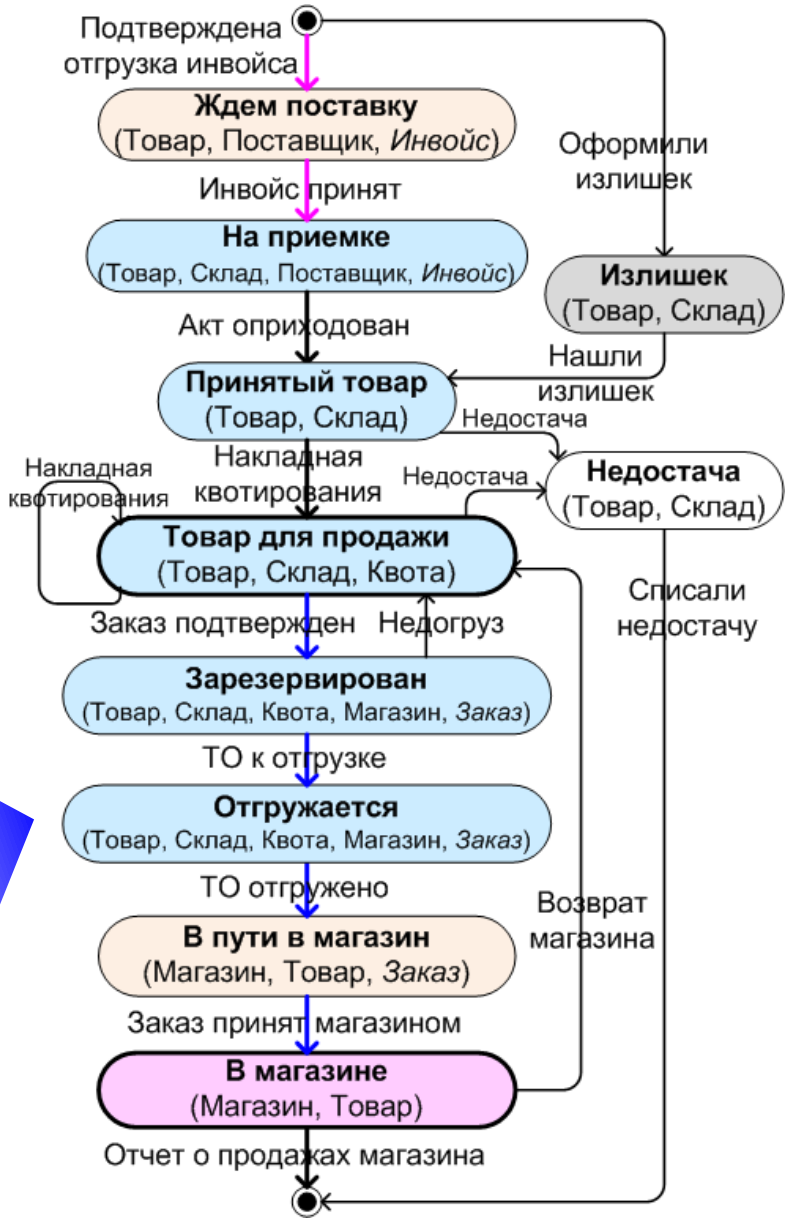
– определяем источники событий учета

Изменения

– добавляем новые участки учета

– добавляем транзитные счета

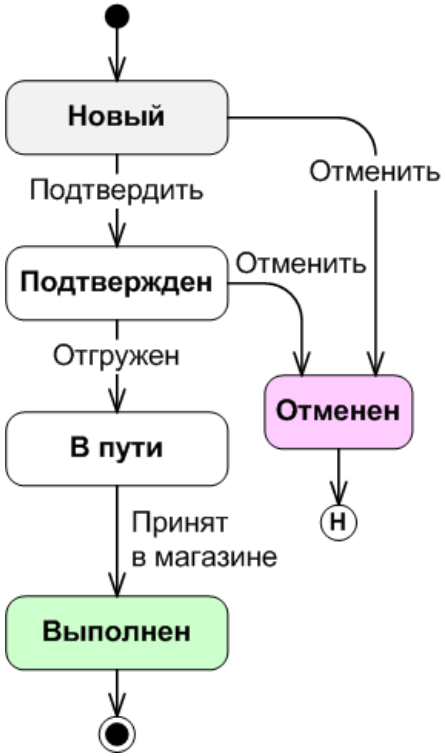


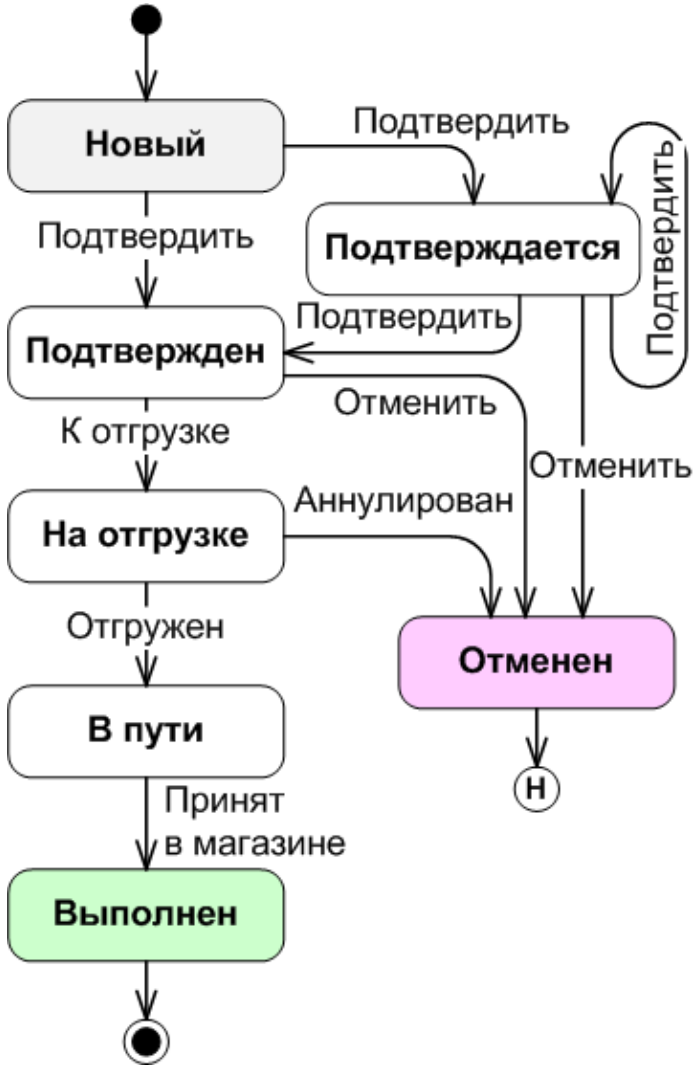


CUSTIS

Проработка документооборота

От простого,
определяемого учетом –
к сложной поддержке
бизнес-процессов





CUSTIS

Сложная модель связанных документов

CUSTIS®

Реализация Учетной машины

CUSTIS®

Платформа реализации

Большинство приложений мы делаем на C# и Oracle

Объектная модель реализуется понятным образом – создаем классы, реализуем методы и т. д.

Для модели документооборота и учета
нужно типовое отображение в объектную реализацию

Что значит «типовое»? Для реализации достаточно постановок с диаграммами переходов и учета, дополнительного проектирования не требуется

CUSTIS®

Реализация документооборота

ВЗГЛЯД БИЗНЕСА

Есть **журнал хозяйственных операций**,
возникающих при обработке
и проведении документов

РЕАЛИЗАЦИЯ

источник учета – **события**
(Event Sourcing, Фаулер),
они возникают в методах документов

Для прозрачной модели это **должно** совпадать:
учетные события – суть хозяйственные операции

CUSTIS®

Документооборот – как мы делаем

Для документов применяем design pattern *State Entity*

– У документов есть атрибут – ***состояние***

– Изменение состояния – через вызов метода объекта

– Метод изменения состояния называется также ***переходом*** документа

– Учетные события возникают на переходах документов

Используем реализацию на метаданных –
у метода описываем начальное и конечное состояния

Диаграмма состояний и описание семантики переходов достаточно для реализации разработчиком

CUSTIS®

Реализация 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

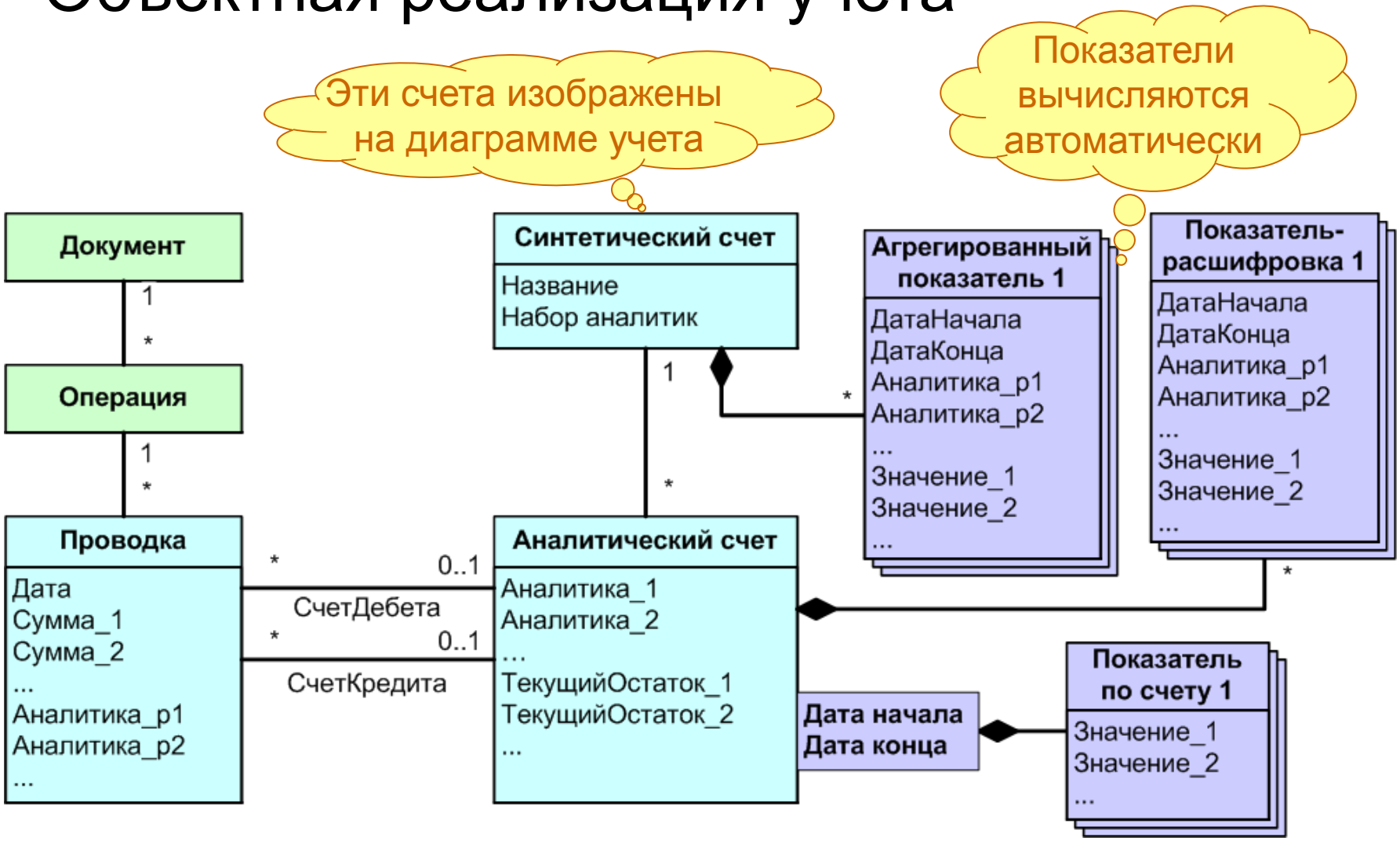
– ведение остатков и оборотов учетных счетов

– ведение детальных и агрегированных показателей

Есть собственный язык описания – GL-XML

CUSTIS

Объектная реализация учета



CUSTIS®

Реализация учета – как мы делаем

Диаграмма учета определяет

– набор учетных счетов и их аналитику

– проводки, выполняемые по конкретным переходам

Еще надо определить набор хранимых показателей, необходимых для эффективной работы отчетов

Задача разработчика

– описать набор счетов, аналитику счетов и аналитику проводок

– описать хранимые показатели

– обеспечить создание проводок на учетных событиях – императивно или декларативно

CUSTIS®

Заключение

(что я хотел сказать)

CUSTIS®

Архитектура по модели – эффективно

Domain Driven Design – архитектура на основе модели

- На едином языке предметной области
- Итеративно и сохраняя целостность

Domain Driven Design – не только объектные модели

CUSTIS®

Да будут архитектурные шаблоны

Шаблоны программирования –
- не только для отдельных фрагментов,
- но и для приложения в целом

Учетная машина – шаблон корпоративного приложения
- Проектирование на основе Domain Driven Design
- Типовое отображение в реализацию

CUSTIS®

Спасибо!

Вопросы?

Максим Цепков (M.Tsepkov@custis.ru)