Иерархии доменных объектов

Поскольку доменные объекты платформы CM 4/5 слишком просты и близки к записям в БД (в частности, не поддерживают хранения множественных значений полей), для удобства моделирования предметной области создаётся инфраструктура иерархий доменных объектов. Практически любой бизнес-объект, например, системы электронного документооборота может быть представлен в виде иерархии доменных объектов. Поэтому , по сути, такая инфраструктура предназначена для бизнес-объектов.

Инфраструктура иерархий доменных объектов включает в себя модельные Java-классы, представляющие иерархию, конфигурацию, позволяющую задавать правила построения иерархий, а также сервис, выполняющий основные операции с иерархиями доменных объектов. В будущем планируется также добавить модуль, осуществляющий генерацию конфигурации бизнес-объектов по Java-классам, созданным разработчиками бизнес-модулей системы, и аннотированным специальным образам. Именно эти классы, располагающие удобным бизнес-интерфейсом, будут передаваться в таком случае между сервисом бизнес-объектов и слоем бизнес-логики системы.

# Модель данных

Ключевым объектом модели данных для иерархии доменных объектов является бизнес-объект, хранящий саму иерархию. Узлы иерархии (дерева) – обычные доменные объекты платформы. Интерфейс класса позволяет обращаться к полям бизнес-объекта через эти доменные объекты либо напрямую с использованием DOEL-выражений.

Класс бизнес-объекта может наследоваться при создании конкретного бизнес-объекта, обладающего удобным интерфейсом для работы с его свойствами (полями). Связь бизнес-объекта с его Java-классом отражается в конфигурации и используется при создании экземпляров бизнес-объектов в коде.

Помимо доменных объектов, являющихся его частью, бизнес-объект может содержать представления (views) других доменных объектов, используемых при его отображении в пользовательском интерфейсе. Например, если бизнес-объект документа содержит маленькие доменные объекты «Согласующий», каждый из которых хранит ссылку на доменный объект «Персона», уже не являющийся частью документа, то для отображения информации о согласовании документа понадобятся значения некоторых полей ДО «Персона», например, имя и фамилия этого лица. Представление ДО хранит именно такой сокращённый набор полей.

Методы бизнес-объекта, позволяющие изменять ссылочные поля, по возможности, поддерживают актуальность представлений ДО, входящих в него (без обращения к БД). Для этого они оперируют не идентификаторами, а экземплярами доменных объектов. Однако при изменении ссылок в доменных объектов, входящих в состав бизнес-объекта, их собственными методами корректное представление сформировано быть не может.

Идентификатором бизнес-объекта является идентификатор его корневого доменного объекта.

# Конфигурация

Инфраструктура бизнес-объектов определяет один тег верхнего уровня конфигурации: business-object-type. Описание этого и вложенных в него тегов приводится ниже.

## Тег business-object-type

* Атрибуты:
  + **name** Имя бизнес-объекта (обязательный)
  + **root-domain-object** Тип корневого доменного объекта иерархии (обязательный)
  + **java-class** Квалифицированное имя Java-класса, представляющего бизнес-объект (необязательный). Класс должен быть потомком GenericBusinessObject.
* Вложенные теги:
  + **children** Дочерние доменные объекты, включаемые в иерархию (множественный). Описание см. ниже.
  + **child** Дочерний доменный объект, включаемый в иерархию, ссылка на который хранится в родительском объекте (множественный). Описание ниже.
  + **view** Представление доменного объекта, включаемое в бизнес-объекта (множественный). Описание ниже.

## Тег children

* Атрибуты:
  + **type** Тип дочернего доменного объекта (обязательный)
  + **link-field** Имя поля, хранящего ссылку на родительский объект (по умолчанию – parent)
* Вложенные теги:
  + **children, child** Так же, как в business-object-type. Вложенность тегов children может быть любой глубины
  + **view** Так же, как в business-object-type.

## Тег view

* Атрибуты (должен быть ровно один из перечисленных):
  + **field** Имя ссылочного (reference) поля, для которого создаётся представление.
  + **type** Тип доменного объекта, для которого создаётся представление. При использовании этого атрибута представления будут созданы для всех полей, ссылающихся на объекты этого типа.
* Вложенные теги:
  + **field** Поле, включаемое в представление объекта (множественный)

## Тег field

* Атрибут:
  + **name** Имя поля
* Вложенный тег (один из перечисленных или отсутствует. В последнем случае, наиболее типичном, впрочем, используется поле с соответствующим именем из самого объекта):
  + **doel** DOEL-выражение (внутри), используемое для вычисления значения поля
  + **java-class** Зарезервировано на будущее

# Сервис

Сервис бизнес-объектов является публичным сервисом (EJB) платформы. Аналогично CrudService,он предоставляет CRUD-операции для бизнес-объектов, описанных в конфигурации системы. Как и CrudService, сервис бизнес-объектов делегирует операции по физическому взаимодействую с БД сервису доменных объектов DAO-слоя.

При чтении бизнес-объекта читаются все доменные объекты, входящие в его состав, а также объекты, необходимые для формирования представлений. При сохранении бизнес-объекта представления игнорируются. Также при сохранении бизнес-объекта обеспечивается изменение его даты модификации, даже если поля изменялись только в дочерних доменных объектах. Следует также отметить, что операция изменения бизнес-объекта может вызывать создание или удаление дочерних доменных объектов.

Любая операция, предоставляемая сервисом, выполняется в одной транзакции, гарантируя целостность бизнес-объекта.