Управление большими объемами требований в BI–проектах

Автор: Дмитрий Зиновьев
старший руководитель проектов Citia BTC

Создание информационно-аналитической системы для крупной организации всегда представляет собой крупный и сложный проект, который нацелен на реализацию комплексных целевых задач, таких как, например, автоматизированный расчет показателей прибыльности продуктов и направлений бизнеса. Подобные задачи, как правило, решаются путем довольно сложных вычислений, на основе огромного количества составляющих и переменных. Да и сама специфика бизнеса, в силу многогранности деятельности крупной компании, диктует необходимость закладывать в проект множество требований к различным аспектам и атрибутам, на которых строится модель этого бизнеса в аналитической системе. Все это приводит к тому, что проект сопровождается колоссальным количеством бизнес, функциональных и технических требований, что, в свою очередь, обуславливает критичность для успеха проекта проработанной методологии и надлежащего инструмента управления требованиями.

Как правило, BI[1]-системы создаются с использованием ряда промышленных инструментов, как для обработки данных, так и представления их конечному пользователю. Такие инструменты, как например, SAP Business Objects Edge BI, IBM Cognos BI, MicroStrategy обеспечивают наличие в созданной системе развитого и стабильного пользовательского интерфейса с фиксированным набором функциональных возможностей и технических параметров, которые не требуют дополнительной проработки в ходе проекта. Это означает, что функциональные требования к системе будут сосредоточены на составе данных и различного рода расчетных величинах, представленных в виде показателей и размерностей, а пользовательский интерфейс будет описан на уровне отчетов и дешбордов, а также их компонент. Кроме того, в подобных проектах иногда также требуется проведение анализа бизнес-требований[2], например, в виде решаемых подразделениями аналитических задач или проблем, с последующим контролем реализации таких требований.

Отдельной строкой идут технические параметры будущего решения, которые в основной своей массе направлены на достижение приемлемого регламента работы системы, который бы удовлетворял потребностям бизнеса в своевременном получении необходимой информации. При этом технические требования достаточно статичны, представляют собой простое перечисление условий и не имеют длинных цепочек связей.

---

[1] *Business Intelligence, бизнес-аналитика*

[2] *Так бывает далеко не всегда, поскольку довольно часто инициатором внедрения аналитической системы выступает ИТ-подразделение, при этом обычным явлением становится ситуация, когда у проекта есть довольно четкие функциональные и технические требования никак не подкрепленные бизнес-задачами, для решения которых собственно и затевается внедрение системы. В таком случае проект сталкивается с существенным риском неприятия и недоверия к готовой системе со стороны конечных пользователей.*

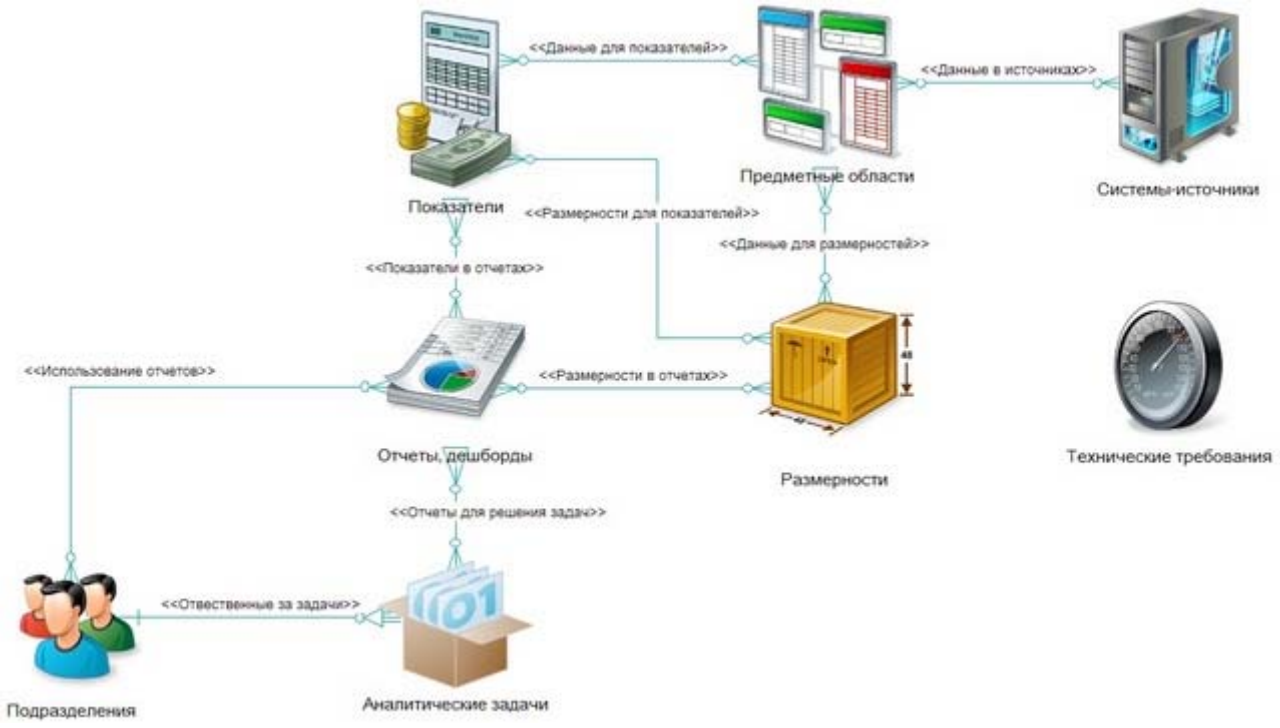

Рисунок 1. Виды требований и их взаимосвязи

На старте проекта очень часто приходиться пояснять Заказчику, зачем вообще мы хотим применять какую-либо специализированную систему управления требованиями. Ведь, на первый взгляд, все требования можно собрать и описать, используя лишь простые офисные средства – текстовый и табличный процессоры. Однако если рассмотреть подробнее, что именно должно быть сделано в части анализа, можно убедиться в том, что уже после первичного сбора требований, предъявляемых к системе, становится крайне обременительным искать и прорабатывать отдельные объекты в текстовых списках по несколько сотен позиций. Но, даже если проделать непростое упражнение и свести все требования в два-три разрозненных файла, то при появлении малейшей потребности в изменении исходных требований, сразу же возникает необходимость повторной ручной проработки всех объектов, которые так или иначе связаны с той частью, в которую были внесены первоначальные изменения. Кроме того, в ходе реализации проекта очень часто возникает необходимость представить исходные в удобном для проектировщиков виде, например, сформировать срез ряда атрибутов или показать цепочки взаимосвязанных объектов, отвечающих определенным критериям. В качестве иллюстрации можно привести необходимость сформировать в интересах разработчика отчетов документ, который бы содержал перечень требуемых отчетов, показателей и размерностей в каждом отчете с указанием структур в модели данных, на которых показатели и размерности формируются.

На рынке представлено несколько достаточно удачных инструментов управления требованиями, которые обладают более или менее равноценными возможностями. Тем не менее, Power Designer от SAP Sybase можно выделить на фоне остальных за счет развитого и удобного пользовательского интерфейса и возможностей по широкой настройке состава объектов и атрибутов. Наиболее, полезной и уникальной возможностью является действительно полноценная интеграция моделей для управления требованиями, проектирования баз данных и ряда других, достаточно полезных моделей. Речь идет о возможности перейти непосредственно от объекта требований к таблицам и атрибутам в логической или физической модели данных, на которых базируются требования.

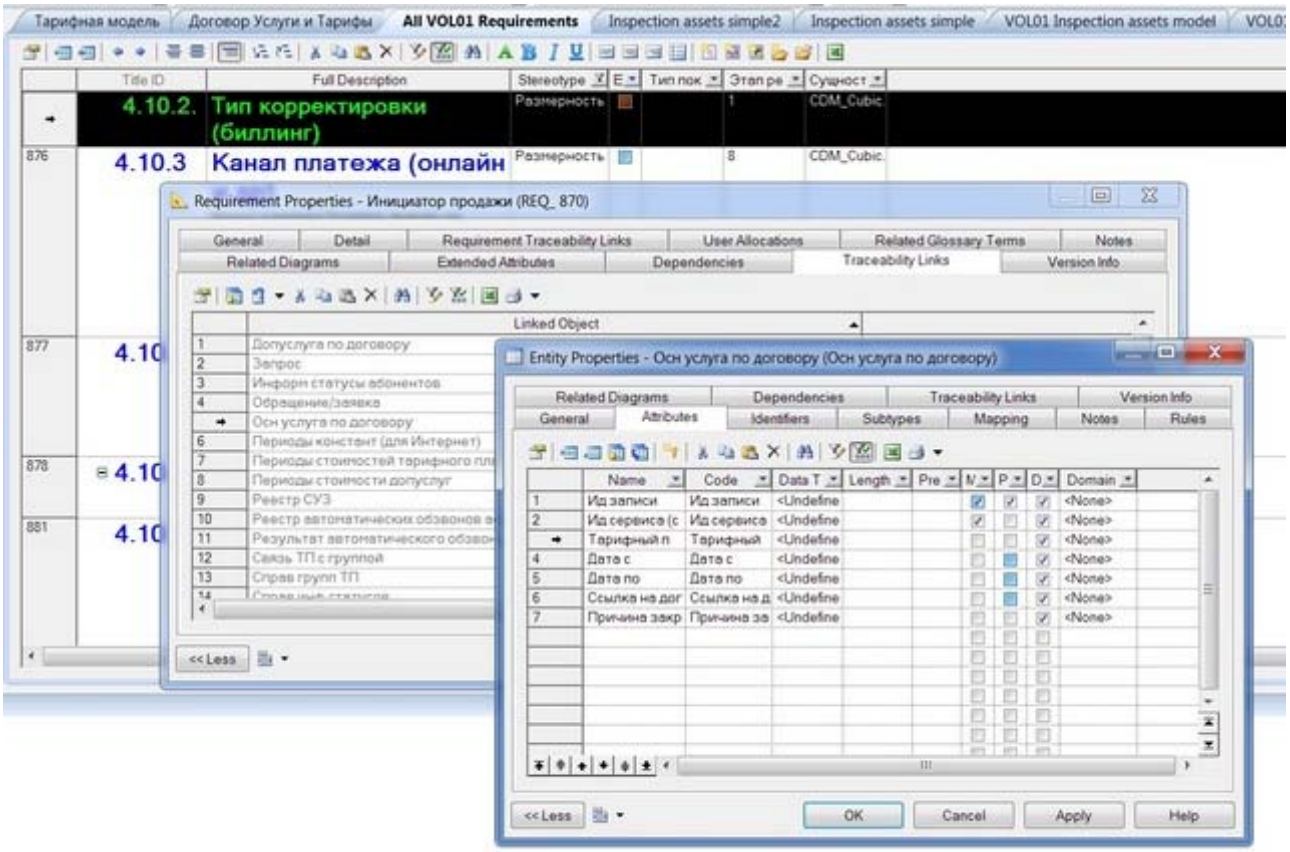

Рисунок 2. Просмотр свойств сущности в концептуальной модели данных, связанной с исходным требованием

Такая интеграция позволяет обеспечить команду проекта единой средой для анализа и проектирования, упрощает формирование проектной и пользовательской документации и выводит процесс создания аналитической системы на качественно новый уровень.

Наиболее разумным шагом на старте проекта будет разделение требований на некоторое логическое количество типов со своим атрибутным составом. Для каждого типа необходимо настроить в расширениях модели собственный стереотип. Довольно полезным является создание отдельного стереотипа для группировки требований. Такой стереотип необходим, поскольку существующий механизм использования пакетов[1] в модели недостаточно гибок, особенно когда необходимо систематизировать требования в иерархиях, например, разделить показателей на группы по предметным областям или направлениям анализа.

---

[1] *В англоязычной версии интерфейса Power Designer – Package*

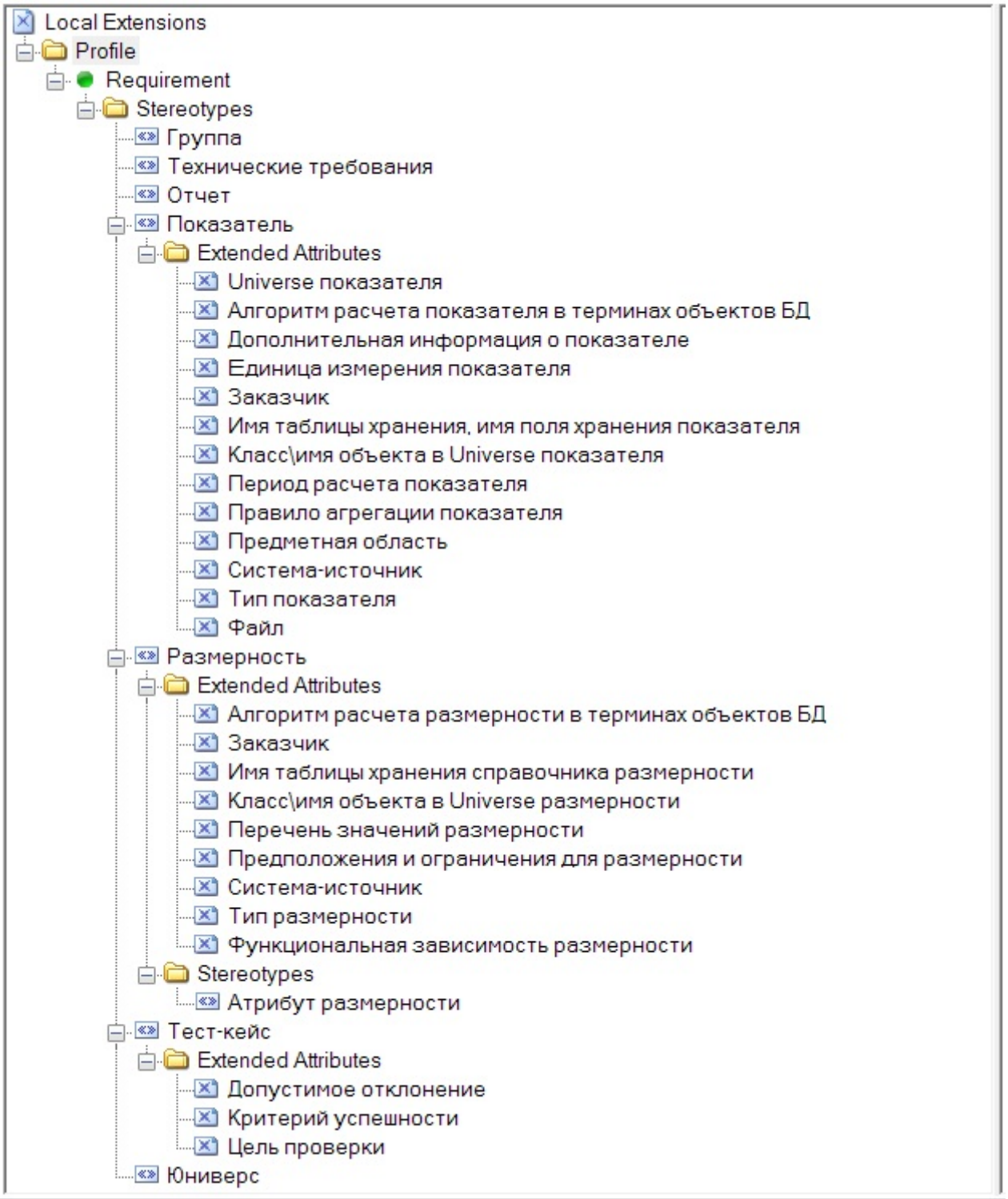

Рисунок 3. Пример настройки типов требований и их атрибутов

Первичная обработка и ввод собранных у Заказчика требований, как правило, происходит определенными наборами. Источником могут послужить опросные листы, интервью или заранее подготовленные наборы требований, которые подаются аналитикам проекта. На этом этапе можно легко обойтись без монотонного ручного набора текста, воспользовавшись инструментом импорта данных из формата XLS. Необходимо отметить, что это универсальный интерфейс, который позволяет импортировать данные в виде объектов, в том числе и в виде связей между объектами, любого типа в любой вид модели. Настройка импорта не составляет особой сложности и

при необходимости может быть сделана для нескольких файлов по одному шаблону, что также положительно сказывается на производительности труда аналитиков.

Как уже говорилось, через импорт из XLS возможно создание не только объектов, но и взаимосвязей между ними, поэтому, если имеется готовая матрица связей показателей с размерностями или отчетами, то следует эту информацию также загрузить в инструмент автоматически, не расходуя время на ручную обработку.

Следует отметить, что в случае, когда количество требований одного типа превышает 50, необходимо рассмотреть вопрос о создании отдельных пакетов в модели либо использования нескольких взаимосвязанных моделей требований в проекте. Такое разделение необходимо, поскольку при ручной проработке взаимосвязей матрицы и выпадающие списки при большом количестве записей становятся совершенно неуправляемыми и значительно замедляют работу аналитика и рабочего места. Например, если в проекте есть порядка 150 показателей, то следует их сначала разделить на укрупненные непересекающиеся группы по видам анализа, такие как продажи и складские запасы для розничного бизнеса или продукты и договора для финансовых организаций. Применять можно любой удобный для дальнейшей работы принцип дробления. Деление по пакетам или моделям никак не ограничивает возможности по работе с взаимосвязями, а лишь систематизирует и упрощает визуальное представление и удобство в дальнейшем при проработке и анализе требований.

Помимо логического дробления и систематизации требований, в работе может понадобиться некоторая автоматизация процессов детализации и связывания требований. Например, если в связи юниверса[1] добавляется ссылка на показатель, логично было бы добавлять туда же и ссылки на размерности, в которых показатель может быть рассчитан. Для этого необходимо создать обработчик событий[2], представляющий собой скрипт VBS[3], который будет проверять связи добавленного показателя и, при необходимости, создавать связь между целевым юниверсом и размерностью. Аналогичным способом можно решить задачу формирования рамок реализации в проекте, когда скрипт будет проверять все связи бизнес-требований, отобранных к реализации, и проставлять значение в соответствующем атрибуте всех связанных с бизнес-требованиями объектов.

По мере формализации требований к аналитической системе наступает время проектировать структуру базы данных и делать отображение требований на создаваемые таблицы и поля. Когда требований и таблиц не очень много, это можно без особых проблем сделать через базовые инструменты связи. Однако, если счет таблиц идет на десятки и сотни, а колонок на сотни и тысячи, то аналитики и архитекторы начинают прямо говорить, что проще скопировать и вставить в экселе названия, чем искать их в выпадающих списках или огромных матрицах. В этом случае также стоит потратиться на написание скрипт, который позволит через вставку нужного поля в текстовом виде автоматически находить и связывать требования с искомой таблицей.

---

[1] *Семантический слой, проецирующий структуры хранения на бизнес-термины, с помощью которого не-технические специалисты могут самостоятельно строить запросы к хранимым данным.*

[2] *Event handler*

[3] *Visual Basic Script*

Когда спецификация требований подходит к концу, встает вопрос — в каком виде эти требования согласовывать с заказчиком? В PowerDesigner существует несколько механизмов для формирования выходной документации. Прежде всего, это могучий, но далеко не самый простой в понимании и использовании генератор отчетов. Конечно, с помощью этого генератора можно выдать готовый документ «Техническое задание» со всеми необходимыми сведениями вплоть до корпоративных стилей и шрифтов. Однако на практике очень часто оказывается, что такой документ формируется только один раз, после чего в него вносятся косметические правки, поскольку требования в том или ином виде уже были доведены до представителей Заказчика и правки в документ вносятся по большей части косметические. Таким образом, время на настройку шаблона, которое могло бы быть скомпенсировано за счет повторной генерации документа по результатам правок в требованиях, оказывается потраченным впустую. Поэтому, как правило, данные для формирования результирующих документов наиболее эффективно извлекать через инструмент экспорта в формат XLS, который не требует никаких особых настроек и одним нажатием кнопки выдает в листах Excel содержимое представлений, матриц и списков именно в том виде, как они представлены в интерфейсе самого PowerDesigner.

За всеми этими настройками и операциями в инструменте есть один существенный вопрос — кто это все должен делать? Исходя из своего опыта — могу сказать: бизнес-аналитики существа возвышенные и достаточно далекие от скрипто-писательства или сложных настроек в моделях, поэтому для эффективной работы с расширением возможностей PowerDesigner в команде необходим человек с техническим бекграундом, например, системный аналитик или архитектор, который хорошо владеет или способен в короткие сроки разобраться со спецификой системы. Без такого человека вряд ли удастся гибко подстраивать методологию и технологию работы команды под возникающие в проекте изменения и вызовы.

Примеров по использованию и настройке PowerDesigner можно привести еще очень много, но общий их смысл сводится к одному: человек, который руководит работой команды должен очень четко себе представлять суть и методику выполняемых специалистами действий и реагировать на возникающие проблемы. Процесс развития методологии и инструментов по сути должен продолжаться от проекта к проекту, для того что бы труд аналитиков и архитекторов был максимально эффективным и результативным.