

электронный журнал

МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК

Издатель: ФГБОУ ВПО «Московский государственный технический университет им. Н.Э. Баумана»

УДК 004.94

Редактор модели реализации отраслевого каталога изделий на основе модели-ориентированного подхода

Пчелин И.Ю. Студент,
кафедра «Информационные системы и телекоммуникации»

Научный руководитель: А.М. Иванов,
ассистент кафедры «Информационные системы и телекоммуникации»

МГТУ им. Н.Э. Баумана
n4ejia@gmail.ru

В настоящее время многие предприятия сталкиваются с проблемой размещения и сопровождения справочной информации по своим изделиям. Это связано с тем, что на различных стадиях жизненного цикла производимой продукции используются различные системы автоматизации проектирования. Соответственно каждый участник стремится создать специализированный каталог оборудования, содержащий интересующие его данные. Данные, необходимые на дальнейших стадиях жизненного цикла в такие каталоги, как правило, не попадают. Между тем, вся информация по оборудованию имеется у производителя, который заинтересован в использовании своего оборудования в проектах.

Выходом из данной ситуации является создание единого отраслевого каталога оборудования, в котором производитель может опубликовать всю информацию, сопровождающую оборудование на протяжении всего жизненного цикла. Таким образом, формируется единая точка, в которой аккумулируются все данные по оборудованию.

Примером подобного каталога может служить отраслевой каталог оборудования атомных электростанций, разработанный на основе системы управления жизненным циклом изделий Enovia. Подробное описание самой системы Enovia дано в [5]. Данный отраслевой каталог – это систематизированная, удобная для использования информация о продукции, производимой на рынке для нужд атомной отрасли. Каждая единица каталога: теплообменник, насос, арматура – имеет 70-80 атрибутов. Это позволяет в конкурентной среде выбрать оборудование с лучшими свойствами по качеству и цене.

Рассматриваемый отраслевой каталог позволяет сгруппировать всю продукцию и ввести необходимую для удобства просмотра и поиска информации иерархию созданных групп продукции. Каждой группе в соответствие ставится свой набор параметров и отношений. Конкретные изделия можно добавить в любую из созданных групп. Создание подобного каталога позволяет четко структурировать информацию о продукции и облегчить ее изменение, а так же позволяет пользователям в любой момент получать всю необходимую информацию об определенном изделии и предоставляет удобный способ поиска необходимой информации.

Использованная для реализации каталога система управления жизненным циклом изделий Enovia предоставляет средства для создания иерархической структуры любой

сложности. Для этого в системе существует сложная структура различных взаимосвязанных данных. На ее основе строится пользовательская структура данных, в соответствии с которой создаются объекты, соответствующие изделиям реального мира.

Таким образом, в системе Enovia существует два различных уровня пользовательских данных:

1. Схема данных;
2. Данные, соответствующие схеме и представляющие объекты реального мира.

Создание и редактирование отраслевого каталога в системе управления жизненным циклом изделий Enovia представляет собой сложную задачу. Во-первых, для работы в системе Enovia требуется определенный уровень подготовки и довольно обширный набор знаний. Во-вторых, редакторам отраслевого каталога приходится создавать сущности, как схемы данных, так и объектов этой схемы, устанавливать порой достаточно сложные и неоднозначные взаимосвязи между этими сущностями и работать в различных модулях системы Enovia. Кроме этого редакторам необходимо заботиться не только об удобном для восприятия и понимания названии создаваемых сущностей, но и об их системных именах.

В свете вышесказанного можно заключить, что текущий вариант работы с отраслевым каталогом довольно сложен, и его можно значительно облегчить. Для этого необходимо создать редактор отраслевого каталога, в котором необходимо бы было задавать только минимум параметров для объектов каталога. Всю работу по связи с системой Enovia и предоставлению информации, необходимой ей для создания соответствующих объектам каталога сущностей, редактор должен осуществлять автоматически. Для работы с редактором должен требоваться минимальный уровень подготовки.

Разработка подобного редактора сложная и объемная задача. Для облегчения процесс разработки разбит на отдельные этапы. Первым этапом является реализация возможности работы с данными системы управления жизненным циклом изделий Enovia извне. На этом этапе редактор должен выполнять следующие задачи:

1. Выгрузка данных из системы управления жизненным циклом изделий Enovia;
2. Загрузка данных в систему;
3. Редактирование данных.

Подробный анализ платформ и подходов к разработке с целью определения наиболее предпочтительных для использования в рамках указанной задачи выходит за рамки данной статьи. Платформой для разработки был выбран Eclipse, что объясняется его открытостью, а так же возможностями по расширению с помощью механизма плагинов. Выбор данной платформы позволит использовать стандартный интерфейс среды для обеспечения работы пользователя, что значительно сократит время разработки. Механизм плагинов Eclipse позволяет добавлять необходимые элементы пользовательского интерфейса к стандартным. Подробное описание архитектуры и функционала данной платформы дано в [6].

В качестве подхода к разработке редактора была выбрана концепция разработки программного обеспечения, управляемой моделями (Model Driven Development MDD)[3]. Идея, лежащая в основе данного подхода, заключается в предельной автоматизации процесса генерации кода, благодаря чему разработчики могут сосредоточиться на создании самого алгоритма. Так же стоит отметить, что программное обеспечение, основанное на метапрограммировании, является гибким, так как оно написано не для конкретного случая данных, с конкретной структурой и с фиксированными зависимостями, а работает в общем случае. Это позволяет быстро и эффективно решать возникающие проблемы, а так же сделать данные семантически более насыщенными и явно выразить скрытую семантику структуры данных.

В качестве реализации концепции разработки программного обеспечения, управляемой моделями была выбрана среда Eclipse Modeling Framework (EMF)[1, 2]. Данная среда соответствует стандарту, разработанному Object Management Group для концепции MDD. EMF, так же как и Eclipse, является открытой средой и в настоящее время активно развивается. Еще одним преимуществом подобного выбора является возможность генерации не только кода модели данных, но и редактора модели для среды разработки программного обеспечения Eclipse.

Задачу разработки редактора отраслевого каталога в указанном выше объеме можно разделить на отдельные модули. Модуль соединения отвечает за связь с системой Enovia и передает объект управления соединением в другие модули. В соответствии с подходом MDD, модель данных описывается с помощью метамодели, по которой генерируется код. Вся работа с метамodelью так же вынесена в отдельный модуль. Разработанная метамодель представлена на рис.1.

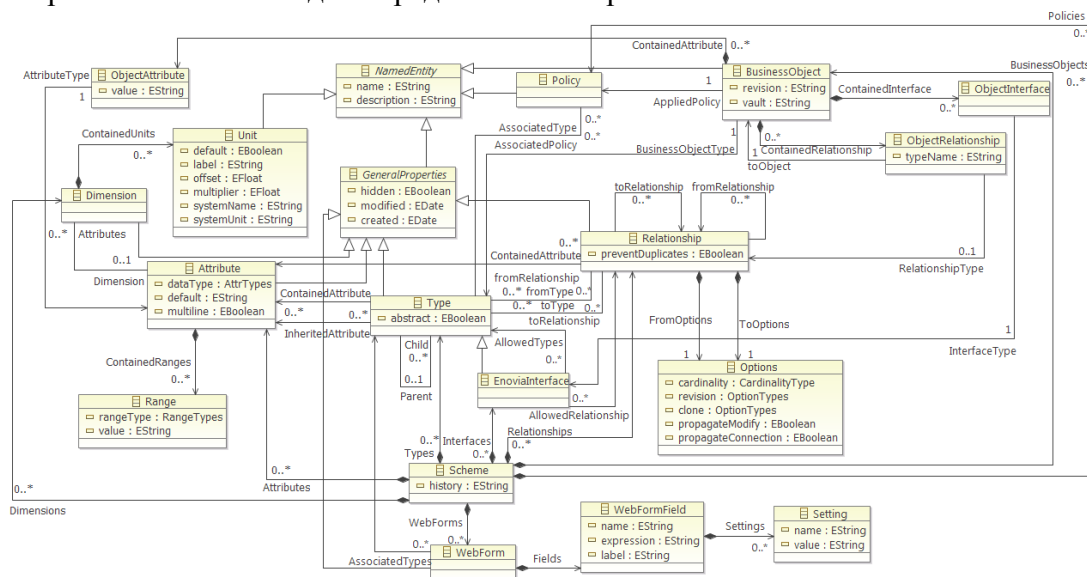


Рис. 1. Метамодель схемы данных системы управления жизненным циклом изделий Enovia

Данные получаемые из системы не имеют прямых взаимосвязей, поэтому для удобства работы с ними выгрузка данных разделена на два этапа:

1. Выгрузка данных в промежуточный набор классов, где связи представляются только именами требуемых элементов.
2. Создание на основе полученного набора классов модели XMI (стандартный формат для среды EMF), где уже представлены связи между элементами.

Чтобы редактор удовлетворял требованиям модифицируемости, для каждого элемента схемы данных был создан свой класс для его промежуточного хранения, полностью соответствующий структуре этого элемента.

Преимуществом использования промежуточного хранения, помимо удобства работы и сокращения времени выгрузки элементов за счет сокращения времени работы непосредственно с системой Enovia, является так же возможность реализации многопоточности для процесса выгрузки данных. Начиная с 2011 года, в стандартной библиотеке системы Enovia добавлена возможность использования объекта управления соединением Context из нескольких потоков.

Поскольку первый этап извлечения никак не требует установку связей между элементами, мы можем извлекать каждый отдельный тип элементов в отдельном потоке.

При этом время выгрузки данных сокращается в несколько раз. Диаграмма зависимости времени извлечения элементов от количества элементов в модели данных для режимов работы в один и несколько потоков представлена на рис.2.

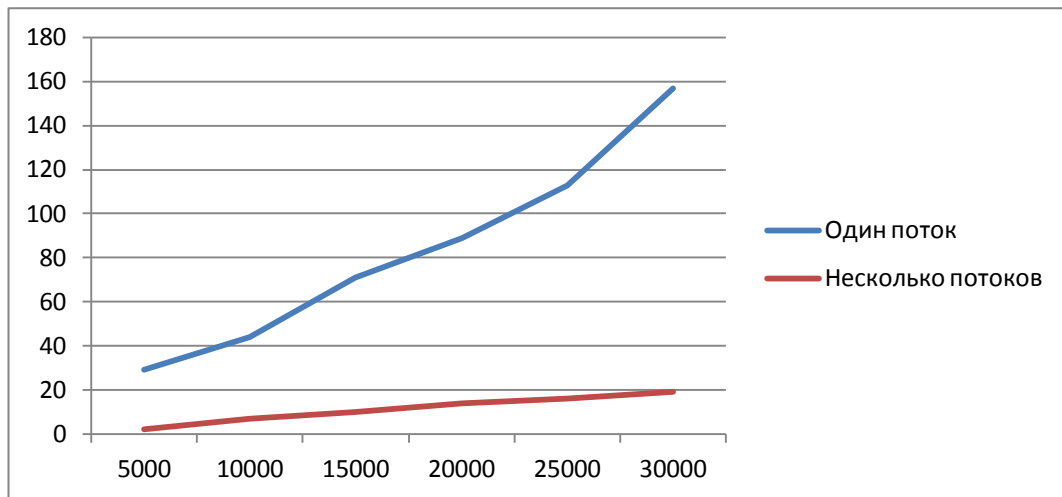


Рис. 2. Зависимость времени выгрузки данных в минутах от количества элементов в модели данных для режимов работы в один и несколько потоков

Если при выгрузке данных возможность разделения работы на отдельные потоки присутствовала, то при загрузке данных в систему Enovia подобной возможности у нас нет. Это связано с тем, что данные очень сильно связаны между собой и, кроме того, существенным фактом является то, что в системе Enovia важен порядок создания элементов. В совокупности указанные факторы могут привести к созданию неверных данных в системе, либо к ошибкам в процессе загрузки.

Загрузка данных в систему может быть реализована несколькими способами:

1. Удалять все данные в системе перед началом загрузки;
2. Проверять в процессе загрузки наличие в системе Enovia загружаемого элемента и сравнивать состояние имеющегося и загружаемого элементов.
3. Загружать не все данные, а только те, что были изменены с последней загрузки.

Первый способ самый простой в реализации, но время загрузки в этом случае максимально. Кроме того, постоянная полная перезапись всех данных не сопоставима с количеством изменений, которые вносятся в модель данных. Еще одним недостатком данного метода является строгое требование постоянного качественного соединения с системой Enovia. Нарушение соединения может привести к тому, что в системе не окажется данных, и она станет недоступной для предоставления информации на некоторое время.

Для второго способа продолжительность загрузки данных так же представляет собой значительную величину. Хотя при использовании данного подхода гарантируется работоспособность системы Enovia в любой момент времени, продолжительность загрузки, так же как и для первого способа не сопоставима с количеством вносимых изменений в систему.

Для первых двух способов на время загрузки влияет не столько количество изменений вносимых разработчиками каталога, сколько его текущий объем. Поскольку планируется, что отраслевой каталог будет содержать информацию по нескольким десяткам тысяч изделий, а разработчик при этом за время между загрузками в состоянии изменить или добавить лишь очень малую часть изделий в каталог, использование данных двух подходов неприемлемо.

Третий способ самый сложный в реализации. Несмотря на это он обладает рядом преимуществ:

1. Время загрузки значительно сокращается;
2. Время загрузки напрямую зависит от количества вносимых изменений;
3. Загрузка данных не нарушает работоспособность системы Enovia.

Использование именно третьего способа наиболее предпочтительно. Несмотря на сложность в реализации, он дает существенный выигрыш по времени.

Запись истории изменений реализована с помощью стандартных средств платформы Eclipse Modeling Framework, названным механизмом адаптеров. Общая диаграмма работы механизма записи изменений представлена на рис.3. Подробное описание процесса перехвата и записи изменений дано в [2].

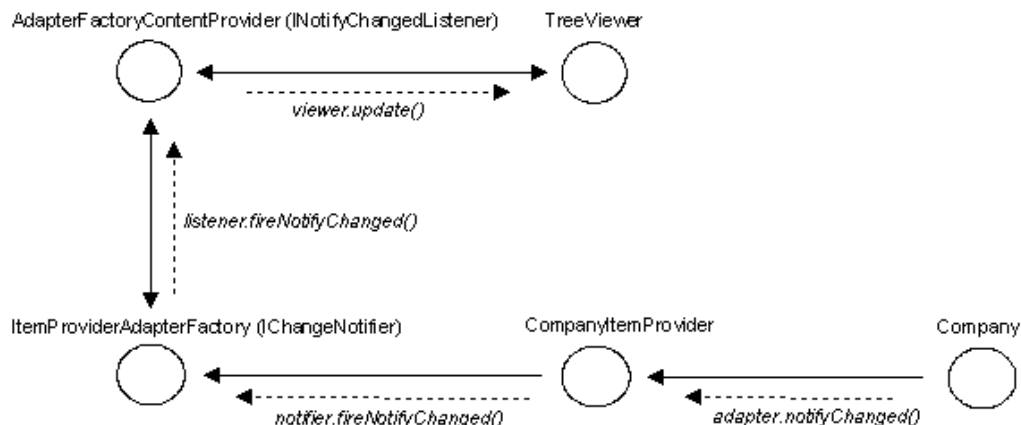


Рис. 3. Диаграмма работы механизма адаптеров EMF

Вносимые в модель данных изменения так же хранятся в модели EMF. Мета модель для истории изменений представлена на рис.4.

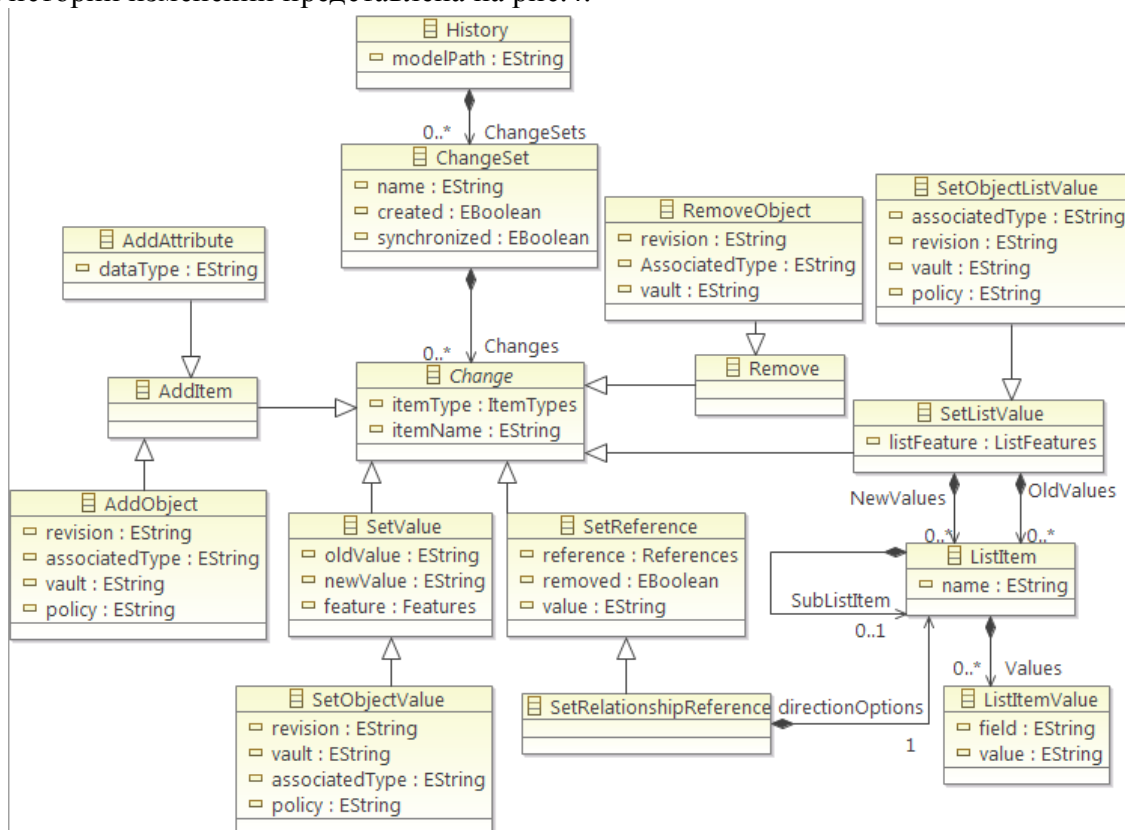


Рис. 4. Мета модель модели изменений схемы данных

Возникающие в процессе загрузки данных в систему ошибки должны быть предоставлены пользователю, поэтому в редакторе предусмотрено окно отчета об ошибках загрузки.

Для предотвращения возникновения ошибок загрузки, связанных с введением некорректных данных, модель перед загрузкой проверяется на ошибки с помощью использования механизма EMF Validation[2].

Так же следует рассмотреть ситуацию, когда пользователю необходимо будет работать не со всем объемом данных, содержащимся в системе Enovia, а только с некоторым его подмножеством. Поэтому в редакторе предусмотрена возможность выбора пользователем выгружаемых данных, включающая в себя указание типов элементов, которые необходимо выгружать и набора префиксов имен для каждого из выбранных типов.

Таким образом, было разработано приложение, способное работать с моделью реализации отраслевого каталога изделий, созданного в системе управления жизненным циклом изделий Enovia. Хотя использование данного редактора по-прежнему требует знаний основных структур данных системы Enovia и их взаимосвязей, оно позволяет разработчикам каталога изменять весь набор данных из единого места.

Основное направление развития – упрощение модели данных реализации отраслевого каталога так, чтобы разработчики задавали только необходимый минимум полей объектов, а все остальное автоматически генерировалось самим редактором. При разработке редактора был разработан DSL язык для описания модели данных реализации отраслевого каталога изделий в системе Enovia. В дальнейшем следует разработать более простой DSL язык для описания модели данных непосредственно самого отраслевого каталога изделий и настроить трансформации данных между моделью каталога и моделью его реализации. При использовании подобного подхода разработчикам каталога не нужно будет ориентироваться в системе Enovia и структуре ее данных. Так же данный подход позволит значительно сократить время разработки каталога, за счет сокращения количества необходимых для создания объектов каталога параметров.

Литература

1. Eclipse Modeling Framework Documentation [Электронный ресурс] / The Eclipse Foundations – Режим доступа: <http://www.eclipse.org/modeling/emf/docs/>, свободный.
2. Frank Budinsky, Dave Steinberg, Ed Merks, Ray Ellersick, Timothy J. Grose. Eclipse Modeling Framework. Second Edition, 2009.
3. J. Mukerji, J. Miller, eds., "Model Driven Architecture", 2001.
4. Jon Siegel, Developing in OMG's Model Driven Architecture, 2006.
5. ENOVIALiveCollaboration Introduction to the ENOVIA Collaboration Platform, 2008.
6. Eclipse Documentation [Электронный ресурс] – Режим доступа: <http://www.eclipse.org/documentation/> свободный.