

## К ЗАДАЧЕ ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ СРЕДСТВ ПРЕОБРАЗОВАНИЯ МОДЕЛЕЙ

ГРАЧЕВ И.В., ассист.

**Представлено экспериментальное исследование производительности средств преобразования моделей, реализующих операционный, реляционный и гибридный подходы к преобразованию. В качестве базы исследования выступают преобразования IDEF0-моделей.**

The paper addresses the experimental evaluation of model transformation performance, which involved several model transformation tools implementing operational, relational and hybrid transformation approaches. The research is based upon transformations of IDEF0 models, the latter being widely used for description of socio-economic and information systems.

*Ключевые слова:* модельно-ориентированный процесс разработки программного обеспечения, средства преобразования моделей, нотация функционального моделирования IDEF0, стандарт MOF QVT.

*Key words:* MDD (model-driven development), model transformation tools, performance evaluation, IDEF0, MOF QVT.

**Введение.** В настоящее время значительный интерес привлекает модельно-ориентированный процесс разработки (МОПР) ПО, который представляет собой итерационное создание и уточнение комплекса моделей, описывающих программную систему с различных точек зрения и организованных в виде иерархии по степени платформу-зависимости [1].

Одним из способов создания моделей необходимого вида является преобразование существующих моделей. Формализация преобразования моделей позволяет осуществлять его как вручную, так и при помощи специальных средств преобразования моделей (СПМ). Автоматизация рутинных операций преобразования моделей позволяет повысить эффективность разработки ПО, что составляет одно из преимуществ МОПР.

**Исходные данные.** Обследование и описание бизнес-процессов во многих организациях производится с использованием нотации IDEF0. При этом IDEF0-модели представляют собой модели высшего уровня абстракции. Так как существует большая база описаний бизнес-процессов в виде IDEF0-моделей, а сама нотация известна широкому кругу специалистов, то обеспечение использования этих моделей в МОПР является актуальной задачей.

В рамках МОПР поддерживается преобразование моделей, метамодели которых описаны средствами языка Meta Object Facility (MOF). В разработанной нами MOF-метамодели для нотации функционального моделирования IDEF0 [2] в качестве основы абстрактного синтаксиса выбрано дерево функций, а конкретный синтаксис базируется на стандарте IDEF0 и формате IDL-файлов, предназначенных для обмена IDEF0-моделями между CASE-средствами.

Метамодель IDEF0 выполнена в виде статического представления на языке UML с использованием ограничений, задаваемых при помощи языка OCL. При этом использованы только те элементы UML, для которых есть аналоги в MOF (класс, атрибут, ассоциация). Соответствие метамодели требованиям MOF

проверено путем импорта полученного описания в XML-файл. Далее UML-модель импортирована в среду разработки Eclipse с преобразованием ее в Ecoge-модель. Следует отметить, что базовая модель Ecoge практически совпадает с EMOF (часть MOF 2.0), отличия заключаются в названиях элементов. На базе полученной Ecoge-модели средствами каркаса для средств моделирования EMF созданы подключаемые модули к среде разработки Eclipse, обеспечивающие программные операции с IDEF0-моделями и интерактивное редактирование IDEF0-моделей.

Маршрут преобразования IDEF0-модели, по которому будет выполняться эксперимент, приведен на рис. 1.

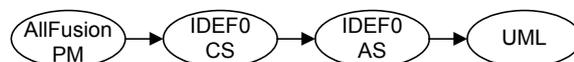


Рис. 1. Маршрут преобразования IDEF0-модели в общем виде

Исходная IDEF0-модель создается в СА AllFusion Process Modeler 7 и экспортируется в формате XML либо генерируется с помощью генератора тестовых моделей и сразу сохраняется в конкретном синтаксисе предлагаемой метамодели IDEF0 (обозначено как IDEF0 CS). В первом случае полученный XML-файл необходимо преобразовать в модель типа IDEF0 CS. IDEF0-модель в конкретном синтаксисе преобразуется в IDEF0-модель в абстрактном синтаксисе (обозначено как IDEF0 AS). Последняя преобразуется в UML-модель.

Преобразование IDEF0 CS → IDEF0 AS описано в [2], а преобразование IDEF0 CS → UML описывается аналогично.

В ходе эксперимента использовались реальные и тестовые IDEF0-модели. В качестве реальных используются демонстрационные модели из комплекта поставки СА AllFusion Process Modeler 7 (Procords и Quill1), обобщенная функциональная модель системы разработки и постановки продукции на производство (SRPP) [3] и модель системы менеджмента качества ВлГУ (QMS).

Для создания тестовых моделей разработана программа-генератор, использующая упомянутый ранее подключаемый модуль к среде разработки Eclipse для программных операций с IDEF0-моделями. Названия тестовых моделей имеют вид testNxM, где N – глубина модели (число уровней декомпозиции), а M – количество блоков на каждой диаграмме декомпозиции.

Все диаграммы декомпозиции имеют одинаковую структуру, на рис. 2 показана диаграмма первого уровня тестовой модели. В указанных реальных моделях среднее количество стрелок, связанных с блоком, составляет 6,2. В генерируемых тестовых моделях этот показатель изменяется в пределах от 5,5 до 5,9, в зависимости от параметра M. Кроме того, количество перекрестков на реальных моделях существенно больше, чем на тестовых. Таким образом, тестовые модели имеют более простую структуру по сравнению с реальными, однако можно генерировать модели большого размера благодаря повторяемой структуре. Так, в модели test4x6 содержится 1555 функциональных блоков, что эквивалентно детализированной эталонной модели операций в цепочке разработки – Design Chain Operations Reference-model (DCOR).

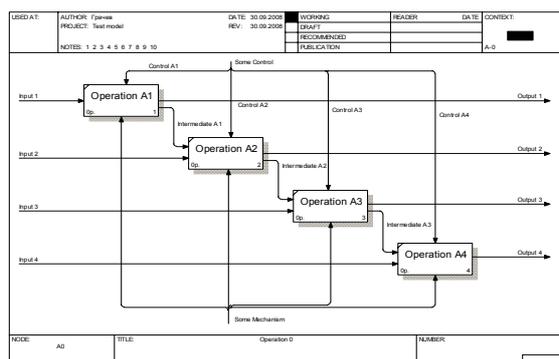


Рис. 2. Диаграмма первого уровня тестовой IDEF0-модели (M = 4)

В ходе эксперимента использовались СПМ, представленные в табл. 1. Все использованные СПМ опираются на каркас EMF для выполнения операций с моделями и, за исключением ModelMorf, выполнены в виде подключаемых модулей к среде разработки Eclipse, благодаря этому выполнение преобразований интегрировано в меню запуска приложений в Eclipse.

Реализуемые различными СПМ подходы к преобразованию моделей приведены согласно классификации Чарнецки-Хелсена. К реляционным относят декларативные подходы, в которых основной концепцией являются математические отношения, фиксирующие типы исходных и результирующих элементов и специфицируемые при помощи ограничений. Операционными же называют императивные подходы, языковые средства которых ориентированы на преобразование моделей (то есть имеется специальный язык описания преобразования, а

не набор API для универсальных языков программирования).

Таблица 1. Используемые средства преобразования моделей

СПМ	Реализуемый подход	Язык описания преобразования	Поддержка стандарта	Редактор	Отладчик
Eclipse Operational QVT	Операционный	QVT-Operational	+	+	-
medini-QVT	Реляционный	QVT-Relations	+/-	+	+
ModelMorf	Реляционный	QVT-Relations	+/-	-	-
SmartQVT	Операционный	QVT-Operational	+	+	+
Tefkat	Реляционный	Tefkat	-	+	+

В 2007 г. организация Object Management Group приняла окончательную редакцию стандарта MOF 2.0 Query/View/Transformation (QVT), который определяет три языка описания преобразования моделей: QVT-Core, QVT-Relations и QVT-Operational. В исследовании использовался единственный нестандартный язык описания преобразования – Tefkat. В настоящее время все СПМ не полностью реализуют язык QVT-Relations, что отражено в табл. 1.

#### Методика проведения эксперимента.

Для проведения эксперимента использовался персональный компьютер со следующими характеристиками: процессор – Intel Core 2 Duo T5600 с рабочей частотой 1,83 ГГц, объем ОЗУ – 1 Гб. На компьютере установлены ОС Windows XP, пакет разработчика Sun JDK версии 1.6.0, среда разработки Eclipse версий 3.2.1, 3.3 и 3.4.

Каждое преобразование выполнялось для каждой модели 5 раз, полученные значения усреднялись. Запуски преобразований осуществлялись без перезапуска среды разработки Eclipse. При этом предполагалось, что разница между физическим чтением модели с диска и из кэша ОС незначительна.

В ходе эксперимента измерялось полное время преобразования, которое включает в себя время загрузки исходной модели, время загрузки описания преобразования и его подготовки к выполнению (за исключением СПМ SmartQVT, где исходный код на языке QVT-Operational заранее транслируется в Java-программу), время преобразования и время сохранения результирующей модели.

Для измерения полного времени преобразования потребовалось добавить в СПМ Eclipse Operational QVT и SmartQVT функцию таймера, остальные СПМ по умолчанию предоставляли соответствующую информацию.

**Результаты эксперимента.** В процессе выполнения эксперимента выяснилось, что получить результаты для некоторых СПМ не удастся. Так, используемая в СПМ ModelMorf версия каркаса EMF устарела, что приводит к

сложностям при работе с метамоделями, создаваемыми в новых версиях Eclipse, также это является наиболее вероятной причиной невозможности работать с метамоделью языка UML. Преобразование IDEF0 CS → IDEF0 AS, реализованное в СПМ Tefkat, выполнялось слишком долго даже для наименее сложной модели test2x4, поэтому соответствующие результаты не приводятся.

Для преобразования IDEF0-моделей, экспортируемых в формате XML из CA AllFusion Process Modeler 7, в конкретный синтаксис предлагаемой метамодели IDEF0 принято решение использовать универсальный язык преобразования XML-документов XSLT, а не специализированный язык описания преобразования моделей. Во-первых, XML-схему для экспорта IDEF0-моделей PMSchema.xsd затруднительно использовать в качестве основы для метамодели, прежде всего, потому что в ней отсутствуют ограничения целостности, позволяющие трактовать значения элементов или атрибутов как ассоциации. Во-вторых, в преобразовании используется лишь малая часть исходного XML-документа, что делает работу с ним как с моделью неэффективной. В-третьих, XSLT-процессоры представляют гибридный подход к преобразованию моделей; в этом подходе в зависимости от стиля XSLT-преобразование может быть императивным или декларативным, а операции выполняются над промежуточными представлениями моделей, то есть над XML-документами, поэтому производительность XSLT-процессоров также представляет интерес.

Для выполнения XSLT-преобразования использовался процессор Xalan, входящий в Sun JDK, предварительная компиляция преобразования не выполнялась.

На рис. 3 приведены маршруты преобразования, реализованные в рамках эксперимента.

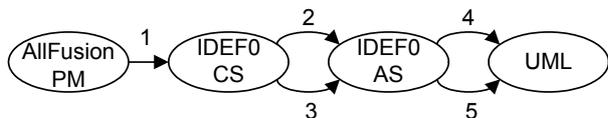


Рис. 3. Реализованные маршруты преобразования IDEF0-моделей: 1 – XSLT; 2 – SmartQVT E3.2; 3 – SmartQVT E3.3; 4 – mediniQVT; 5 – Eclipse Operational QVT

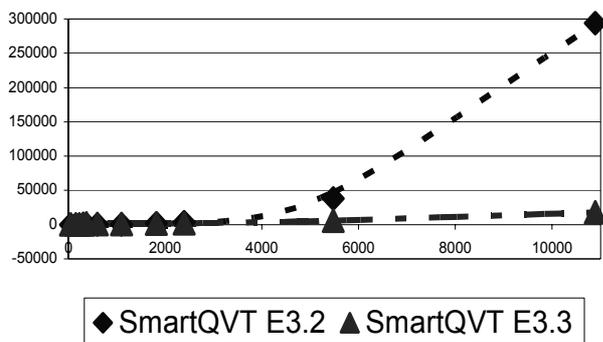


Рис. 4. Модель полного времени преобразования для преобразования IDEF0 CS → IDEF0 AS

В табл. 2 приведены следующие характеристики исходного кода разработанных преобразований: TLOC – общее количество строк;  $N_m$  – количество отображений;  $N_q$  – количество запросов (в отличие от отображений, запросы не создают элементы результирующей модели).

Таблица 2. Характеристики исходного кода преобразований

Преобразование	Язык описания преобразования	TLOC	$N_m$	$N_q$
PM → IDEF0 CS	XSLT	215	6	0
IDEF0 CS → IDEF0 AS	QVT-Operational	160	22	5
IDEF0 CS → IDEF0 AS	Tefkat	148	19	2
IDEF0 AS → UML	QVT-Operational	126	15	1
IDEF0 AS → UML	QVT-Relations	206	19	3

Результаты, полученные в ходе эксперимента по преобразованию IDEF0-моделей, сведены в табл. 3. В заголовке для СПМ SmartQVT и Operational QVT указана использованная версия среды разработки Eclipse. Также в табл. 3 приведены характеристики IDEF0-моделей: количество блоков  $N_b$  и количество стрелок  $N_a$ .

По полученным результатам построены регрессионные модели, графики которых приведены на рис. 4, 5, где также отмечены значения, полученные экспериментально.

### Заключение

По результатам работы можно сделать следующие выводы.

1. Подтвержден тезис [2] о том, что предлагаемая MOF-совместимая метамодель для нотации IDEF0 позволяет использовать IDEF0-модели в модельно-ориентированном процессе разработки ПО.

2. СПМ, реализующие операционный подход, демонстрируют значительное преимущество на моделях большого размера:

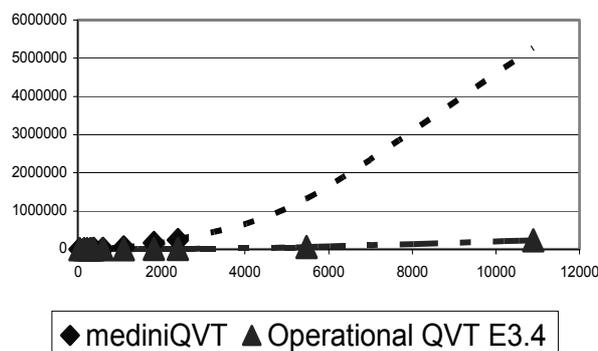


Рис. 5. Модель полного времени преобразования для преобразования IDEF0 AS → UML

Таблица 3. Результаты эксперимента по преобразованию IDEF0-моделей

Модель			Полное время преобразования, мс				
			PM → IDEF0 CS		IDEF0 CS → IDEF0 AS		IDEF0 AS → UML
Назв.	N <sub>b</sub>	N <sub>a</sub>	XSLT	SmartQVT E3.2	SmartQVT E3.3	mediniQVT	Operational QVT E3.4
Procords	6	38	509	15	437	109	88
test2x4	21	120	– <sup>1</sup>	37	563	497	178
test2x5	31	180	– <sup>1</sup>	46	625	1222	325
Quill1	58	247	1800	234	1075	2610	325
test2x6	43	252	– <sup>1</sup>	66	703	2541	428
SRPP	48	325	3109	459	2140	8415	681
test3x4	85	472	– <sup>1</sup>	156	860	7968	1078
QMS	103	750	15059	– <sup>2</sup>	– <sup>2</sup>	– <sup>2</sup>	– <sup>2</sup>
test3x5	156	880	– <sup>1</sup>	522	1109	44734	2803
test3x6	259	1476	– <sup>1</sup>	1828	1640	163552	7250
test4x4	341	1880	– <sup>1</sup>	3560	2116	242887	12262
test4x5	781	4380	– <sup>1</sup>	38234	5894	– <sup>3</sup>	61991
test4x6	1555	8820	– <sup>1</sup>	293940	17944	– <sup>3</sup>	234417

Примечания:

1. Генератор тестовых моделей сохраняет модели в формате метамодели.
2. Исходная модель BPWin не соответствовала ограничениям целостности, поэтому полученная IDEF0-модель не могла быть обработана.
3. Время ожидания результата превысило 20 мин.

разница во времени выполнения преобразования между операционными и реляционными СПМ составляет один порядок. Правда, на моделях небольшого и среднего размера реляционные СПМ также обеспечивают приемлемое время преобразования.

Если судить по характеристикам исходного кода преобразований (табл. 2), реляционные и операционные преобразования имеют сравнимую трудоемкость разработки. Но реляционный подход, в отличие от операционного, обеспечивает поддержку инкрементальных преобразований и построен на базе аппарата математической логики, что позволяет выполнять формальную верификацию преобразований.

Таким образом, наибольшего эффекта можно достичь за счет «расщепления» исходных моделей, то есть уменьшения числа их элементов. Это особенно важно при модельно-ориентированной адаптации ПО, так как в этом случае продолжительность цикла внесения изменений играет важную роль.

3. Использование универсального языка XSLT для преобразования моделей менее предпочтительно по сравнению со специализированными языками. Немаловажно также и то, что в XSLT отсутствует поддержка прослеживаемости преобразования, из-за этого в преобразовании PM → IDEF0 CS пришлось реализовать прослеживаемость «по построению». Фактически преобразование PM → IDEF0 CS выполнено в структурно-ориентированном стиле. Быстродействие XSLT-преобразования ниже, чем у операционных СПМ, но предположительно не ниже, чем у реляционных СПМ.

4. Необходимо использовать последние версии СПМ и базовой платформы (для рас-

сматриваемых СПМ – Eclipse), так как в них улучшается поддержка стандарта и увеличивается производительность, что особенно заметно на примере SmartQVT.

5. Необходимо выполнять валидацию преобразуемых моделей, при этом наиболее важно гарантировать правильность исходных моделей. Это демонстрирует пример модели QMS, ошибки в структуре которой привели к невозможности выполнять преобразования с ее MOF-совместимым эквивалентом.

Рассмотренный вариант маршрута преобразования фактически является частью общего маршрута преобразований в реальном МОПР. Поэтому предполагается продолжить исследование в направлении оценки производительности для более полного маршрута преобразований, а также использования пространственных СПМ, основанных на нестандартизированных языках описания преобразований, таких как ATL и openArchitectureWare.

#### Список литературы

1. Александров Д.В., Грачев И.В., Фадин Д.Н. CASE-технологии: Учеб. пособие / Владим. гос. ун-т. – Владимир: Изд-во Владим. гос. ун-та, 2006.
2. Грачев И.В. Метамодель для нотации функционального моделирования IDEF0 / Математические методы в технике и технологиях – ММТТ-20: Сб. тр. XX Международ. науч. конф. В 10 т. Т. 6. – Ярославль: Изд-во Яросл. гос. техн. ун-та, 2007. – С. 284–288.
3. Грачев И.В. Обобщенная функциональная модель системы разработки и постановки продукции на производство / Трансформация экономики регионов в условиях устойчивого развития. Теория и практика: Матлы межвуз. науч.-практич. конф. – Владимир: Изд-во Владим. гос. ун-та, 2008. – С. 178–180.

Грачев Иван Викторович.  
ГОУ ВПО «Владимирский государственный университет»,  
кафедра ИСИМ (информационных систем и информационного менеджмента),  
ассистент,  
тел. (раб.) (4922) 479-977;  
e-mail: ivgratchev@rambler.ru

Ivan V. Gratchev;  
Vladimir State University,  
Chair for Information Systems and Information Management;  
junior member of teaching staff;  
workphone – +7.4922479977;  
e-mail – ivgratchev@rambler.ru