

Российская Академия Наук  
Сибирское Отделение  
Институт систем информатики им. А. П. Ершова

УДК 004.414, 004.415, 004.416

На правах рукописи

ТАРАБУХИНА Надежда Константиновна

**МЕТОДЫ ИЗВЛЕЧЕНИЯ БИЗНЕС-ЛОГИКИ  
НА ОСНОВЕ СЕМАНТИЧЕСКИХ СВОЙСТВ ПРОГРАММ**

Специальность 05.13.11 –  
Математическое и программное обеспечение  
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ  
диссертации на соискание ученой степени  
кандидата физико-математических наук

Новосибирск 2010

Работа выполнена в Учреждении Российской академии наук Институте систем информатики им. А.П. Ершова СО РАН

**Научный руководитель:** Бульонков Михаил Алексеевич,  
кандидат физико-математических наук

**Официальные оппоненты:** Терехов Андрей Николаевич,  
доктор физико-математических наук

Скопин Игорь Николаевич,  
кандидат физико-математических наук

**Ведущая организация:** Санкт-Петербургский государственный  
политехнический университет

Защита состоится 16 июня 2010 г. в 14 ч. 30 мин. на заседании  
диссертационного совета ДМ003.032.01 при Институте систем информатики  
им. А. П. Ершова Сибирского отделения РАН по адресу:  
630090, г. Новосибирск, пр. Акад. Лаврентьева, 6.

С диссертацией можно ознакомиться в читальном зале библиотеки ИСИ СО  
РАН (г. Новосибирск, пр. Акад. Лаврентьева, 6)

Автореферат разослан \_\_\_\_ мая 2010 г.  
Ученый секретарь  
диссертационного совета ДМ003.032.01,

к.ф.-м.н.



Мурзин Ф.А.

## **ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ**

### **Актуальность проблемы**

В современном обществе компьютеры используются практически во всех областях человеческой деятельности, и компьютерные программы составляют важную часть бизнес-процессов предприятий и организаций. Подобные программы создавались в течение не одного десятка лет и на сегодняшнем этапе требуют постоянной поддержки, модернизации и адаптации к меняющимся условиям их применения. Ситуация осложняется тем, что существенная часть этих приложений разработана с использованием устаревших языков программирования COBOL, PL/1, Natural и других, мало употребляемых современными программистами. Часто такие программы не имеют документации вообще или документация не соответствует коду, поскольку она уже не отражает многочисленных изменений, произведенных за годы эксплуатации. Кроме того, исходный код рассматриваемых приложений может оказаться частично утраченным, что затрудняет компиляцию и отладку.

Также нельзя не отметить, что для любого изменения существующих бизнес-приложений необходимо понимание предметной области их применения, которым программисты в большинстве случаев не владеют. Поэтому важно предоставить системным аналитикам средство работы с бизнес-логикой приложений, требующее от пользователя лишь минимальных навыков программирования. Полученные с помощью такого средства бизнес-правила имеют самостоятельную ценность и являются универсальным методом коммуникации между программистами и аналитиками. В дальнейшем бизнес-правила могут использоваться для корректировки работы имеющейся программы, создания спецификаций новых приложений или документирования кода.

Как показывает практика, многие бизнес-приложения не соответствуют современному уровню развития информационных технологий, поскольку большинство из них было создано до появления персональных компьютеров, сети Интернет и графического интерфейса пользователя. Эти приложения функционируют на устаревшей технике, под управлением архаичных операционных систем и не используют

современные средства хранения и передачи данных. Подобные бизнес-приложения считаются устаревшими или унаследованными (legacy) и, безусловно, нуждаются в модернизации. Возможны различные методы их модернизации, в том числе и с применением таких элементов реинжиниринга (или «обратного проектирования»), как автоматическое извлечение бизнес-логики и автоматический перевод кода приложения на другие языки программирования. Рассмотрим кратко некоторые из этих методов и отметим необходимость использования бизнес-правил для успешного их применения.

*Метод разработки нового приложения для полной замены существующего* является одним из самых дорогостоящих методов модернизации, требующим значительного времени на проектирование, разработку, тестирование и внедрение нового приложения. Предварительное извлечение бизнес-логики исходного приложения в виде бизнес-правил может быть полезно на всех этапах модернизации: в качестве спецификаций при проектировании и разработке и в качестве тестовых сценариев (use cases) при тестировании. Важно, что соответствие новой системы бизнес-правилам, извлеченным из старой системы, обеспечивает корректность модернизации.

*Метод автоматического перевода устаревшего бизнес-приложения на новые языки программирования и новые платформы* является более привлекательным. К преимуществам использования этого метода относится быстрота и относительно низкая стоимость проведения модернизации. Однако необходимо отметить, что полная автоматизация для реально используемых бизнес-приложений не всегда возможна и структура автоматически созданного кода не всегда является оптимальной. Данный метод может обеспечить синтаксически корректный перевод с одного языка на другой, но это, как правило, не приводит к улучшению структуры программы и не позволяет выделить действительно интересные фрагменты (бизнес-правила) из устаревшего кода.

*Метод извлечения знаний* предполагает использование средств извлечения (в том числе и автоматического) и анализа бизнес-правил с последующей реализацией (в том числе и автоматической) полученных правил в новом приложении на новых языках и новых платформах. Этот

метод обеспечивает высокое качество модернизации при высокой скорости проведения работы и относительно низкой ее стоимости.

При использовании любого метода модернизации детали технической реализации устаревших приложений обычно не представляют особой ценности при попытке создания современных аналогов этих приложений. В самом деле, для современных графических приложений, работающих в сети Интернет, совершенно неважно, как именно были реализованы детали пользовательского интерфейса на алфавитно-цифровом дисплее в оригинальной программе или какие средства использовались для обеспечения совместной работы пользователей. Наибольшую ценность представляют используемые в существующих приложениях бизнес-правила и алгоритмы, поскольку именно они определяют, что и как делает приложение.

На сегодняшний день объем существующих бизнес-приложений, нуждающихся в полной замене или дальнейшей модернизации, настолько велик, что актуальность автоматического извлечения бизнес-логики программ не вызывает сомнений.

### **Цель работы**

Цель работы заключается в исследовании методов автоматического построения бизнес-правил на основе семантических свойств программ. Основные задачи в рамках достижения этой цели включают:

- Разработку метода восстановления вычислений заданной программной переменной на основе графа информационных зависимостей программы.
- Разработку метода валидации экранных портов для построения бизнес-правил из всех проверок корректности значений, вводимых пользователем и попадающих в программу через операторы обработки экранных форм – экранные порты.
- Построение вспомогательного алгоритма для валидации сегментов бизнес-правил, позволяющего проверять актуальность правил после редактирования программного кода.

- Реализацию всех разработанных алгоритмов в рамках программного средства анализа бизнес-логики старых приложений Business Rule Manager.

### **Методы исследования**

Для решения поставленной задачи нами использовались различные алгоритмы анализа графов, алгоритмы поиска по шаблону. Для обеспечения многопользовательского режима работы применялась оптимистическая стратегия. Кроме того, использовались результаты работы синтаксических анализаторов языков COBOL, PL/1 и др. Входными данными для автоматического извлечения бизнес-правил служили результаты анализа влияния (impact analysis), а именно специальный информационный граф, построенный путем анализа потоков данных. В процессе реализации разработанных алгоритмов использовались методы объектно-ориентированного программирования.

### **Научная новизна**

Исследования в области бизнес-логики занимают важное место в современной информатике. В данной работе исследована возможность автоматического восстановления бизнес-логики на основании анализа семантических свойств программ, выявляемых по исходному коду. В работе определена оригинальная модель представления бизнес-правил, сформулированы основные задачи восстановления бизнес-логики, представлены алгоритмы, которые могут быть не только использованы для старых приложений на языках COBOL, PL/1 и др., но и преобразованы для анализа более современных приложений.

Автоматически построенные бизнес-правила могут быть использованы, в частности, для анализа, документирования, сопровождения и реинжиниринга существующих приложений, а также для проектирования на их основе новых приложений.

### **Практическая ценность**

Практическим результатом работы явилось создание средства Business Rule Manager как одной из важнейших подсистем системы Modernization Workbench анализа старых приложений, которая в настоящее время успешно применяется многими компаниями Европы,

Азии и Америки для выполнения работ в области реинжиниринга. Описанные в работе методы извлечения бизнес-логики широко используются при анализе и модернизации приложений в области банковской деятельности, страхования, снабжения и управления.

### **Апробация работы**

Результаты работы были представлены на конференциях-конкурсах «Технологии Microsoft в теории и практике программирования», проходивших в Новосибирске в 2006 и 2007 годах, на XLV Международной научной студенческой конференции «Студент и научно-технический прогресс» в 2007 году, на VIII Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям, проходившей в Институте вычислительных технологий СО РАН в 2007 году, на рабочем семинаре «Наукоемкое программное обеспечение» в рамках седьмой международной конференции памяти академика А. П. Ершова «Перспективы систем информатики» (PSI'09) в 2009 году, а также обсуждались в ряде встреч с российскими и зарубежными специалистами в области реинжиниринга.

Автором опубликовано 9 печатных работ, из них по теме диссертации – 7 работ.

### **Структура и объем работы**

Диссертационная работа состоит из введения, четырех глав, заключения и списка литературы. Объем диссертации – 123 стр. Список литературы содержит 40 наименований. Работа включает 4 таблицы и 56 иллюстраций, в том числе полученных с использованием разработанного программного обеспечения.

### **СОДЕРЖАНИЕ РАБОТЫ**

**Во введении** описаны основные проблемы реинжиниринга, перечислены возможные методы их решения, обоснована актуальность задачи извлечения бизнес-логики программ на основе анализа исходного кода, приведена структура диссертации.

**В первой главе** проводится знакомство с бизнес-логикой программ, обсуждаются вопросы стандартизации, представлены различные модели

представления бизнес-правил (в частности, предложена модель представления бизнес-правил, ориентированная на работу с компьютерными программами), сформулированы основные подходы в области автоматизации работы с бизнес-правилами, приведен обзор существующих программных средств.

В частности, в разделе 1.1 приводится определение *бизнес-правила*, как правила, согласно которому работает некоторая бизнес-система. В дальнейшем это определение уточняется и детализируется применительно к программным системам. *Бизнес-логика* определяется как совокупность всех бизнес-правил системы. Попутно обсуждаются история исследований бизнес-правил и стандарты их представления. Среди множества моделей представления бизнес-правил, созданных за последние двадцать лет, рассматриваются модель, предлагаемая в стандарте SBVR (Semantics of Business Vocabulary and Business Rules), и модель, используемая в разработанном нами программном средстве Business Rule Manager, описанном в данной работе. Для удобства первая модель названа SBVR-моделью, а вторая – BRM-моделью.

Необходимо отметить, что, несмотря на определенную схожесть представленных моделей бизнес-правил, между ними есть существенные различия. С нашей точки зрения, основными критериями выбора той или иной модели являются адекватность представления и удобство использования. BRM-модель была успешно использована для построения конкретной системы работы с бизнес-правилами и тем самым доказала свою эффективность.

В разделе 1.2 рассматриваются две задачи: 1) построение аналитиком набора бизнес-правил и их дальнейшее использование в разработке приложения и 2) восстановление бизнес-логики программы на основе ее кода. Первая задача близка к синтезу программ на основе спецификации. Вторая задача может возникать в разных контекстах: для документирования и сопровождения существующего приложения, либо для полной переработки приложения, в частности, для переноса на другую платформу.

Здесь же дается обзор имеющихся программных средств, автоматизирующих работу с бизнес-логикой программ, с классификацией по типу решаемой ими задачи и кратким описанием используемых ими



методов. В обзоре представлены продукты компаний: IBM, Bizrules, Business Rule Solution, XpertRule, SoftwareMining, blackboxIT, Databorough, Trinity Millennium Group, iLog и American Cybernetic Corporation.

**Вторая глава** содержит постановку задачи и рассмотрение методов извлечения (восстановления) бизнес-логики программ. Существующие методы разделены на два класса: слабую и сильную автоматизацию.

Раздел 2.1 посвящен собственно постановке задачи. Поскольку извлечение бизнес-логики – это несколько неточная формулировка, конкретизировать ее можно различными способами. В данном разделе рассмотрены различные варианты конкретизации и указаны те, решение которых предлагается в нашей работе.

Раздел 2.2 рассматривает два класса методов восстановления бизнес-логики программ. Каждый класс рассмотрен в соответствующем подразделе.

В работе предложены собственные методы как слабой, так и сильной автоматизации, а так же рассмотрены методы, реализованные в других программных средствах.

Методы слабой автоматизации включают, например, предоставление интерфейса для создания бизнес-правил бизнес-аналитиком. Любая система управления бизнес-правилами имеет такой интерфейс. В инструменте Business Rule Manager он также реализован. Помимо этого, в качестве метода слабой автоматизации мы предлагаем групповое создание бизнес-правил из выбранных конструкций кода с заполнением атрибутов каждого из бизнес-правил на основе атрибутов конструкций согласно заданному общему шаблону.

Методы сильной автоматизации представляют больший интерес, поскольку могут провести часть анализа и тем самым упростить последующую работу бизнес-аналитика. И именно в разработке таких методов заключалась основная цель данной работы. Нами разработаны и реализованы два метода: метод восстановления вычислений и метод валидации экранных портов. Кроме этого, кратко рассматривается метод поиска использований заданной переменной, который в некотором смысле противоположен методу восстановления вычислений. Разработка этого метода является направлением для дальнейших исследований.

Главная идея предложенных нами методов заключается в построении бизнес-правил на основе специального графа информационных зависимостей. Таким образом, мы проводим анализ семантических свойств программ, который является более глубоким по сравнению с синтаксическим и обеспечивает независимость алгоритмов от языков программирования, использованных при написании исходных программ.

**Третья глава** посвящена собственно предлагаемому нами решению задачи автоматического извлечения бизнес-логики программ. В ней предлагаются три различных метода для разных конкретизаций данной задачи. Работа методов демонстрируется на простых примерах.

В разделе 3.1 подробно рассматривается *метод восстановления вычислений*. Предлагаемый метод строит набор бизнес-правил, последовательное выполнение которых (с учетом условий их выполнения) вычисляет заданную программную переменную.

Метод использует в качестве входных данных специальный информационный граф. Как и стандартный информационный граф, он строится путем анализа потоков данных, ведущих к указанной переменной. Особенность специального графа состоит в том, что он содержит дополнительную информацию об условных зависимостях. На основе данного графа строится операторный граф, который содержит информацию обо всех операторах, используемых для вычисления выбранной переменной, зависимостях по данным между операторами и зависимостях от условий.

После предварительных преобразований операторного графа, включающих разрезание циклов и распространение условий на операторы, с помощью специального обхода графа строится корректная последовательность бизнес-правил. Последовательность считается корректной, если выполнение всех ее бизнес-правил подряд приводит к такому же результату, который выдает исходная программа. Помимо главной цели упорядочения – получить корректную последовательность, существует дополнительная цель – максимально приблизить порядок бизнес-правил к порядку исполнения операторов в программе.

Далее полученные бизнес-правила группируются последовательно разными способами: по принадлежности к программе (в случае наличия

межпрограммных вызовов), по принадлежности к блоку программы, по общим условиям выполнения.

Результатом работы метода является набор бизнес-правил, сгруппированных в бизнес-процедуры. Все бизнес-процедуры связаны между собой посредством триггеров (бизнес-правил, вызывающих исполнение бизнес-процедуры) и реализуют единственную бизнес-функцию.

В качестве примера для демонстрации работы метода рассмотрим фрагмент программы на языке COBOL, вычисляющий скидку (discount) по некоторым правилам.

```
INIT.
  MOVE 0 TO DISCOUNT-FACTOR1 DISCOUNT-FACTOR2.
  MOVE 1 TO I.
  IF UPDATE-NEEDED = "TRUE" THEN
    IF CURDAY > 5 THEN
      COMPUTE DISCOUNT-FACTOR2 = PRICE-0 - 50
      MOVE PRICE-0 TO PRICE
      MOVE 0 TO TAX
    ELSE
      COMPUTE PRICE = PRICE-0 + 10
      MOVE 0.6 TO TAX
    END-IF
  END-IF.

FACTOR-CALC.
  IF I > MAX THEN
    GO TO MAIN-CALC.
  COMPUTE RES = DISCOUNT-FACTOR1 + CONST.
  ADD CONSTS TO RES.
  COMPUTE DISCOUNT-FACTOR1 = RES * 0.25.
  ADD 1 TO I.
  PERFORM FACTOR-CALC.

MAIN-CALC.
  ADD DISCOUNT-FACTOR2 TO RES.
  COMPUTE DISCOUNT-FACTOR2 = 0.5 * DISCOUNT-FACTOR2.
  COMPUTE TAX = TAX * DISCOUNT-FACTOR2 + RES.
  COMPUTE DISCOUNT = DISCOUNT-FACTOR1 * PRICE + DISCOUNT-FACTOR2.
```

Переменная DISCOUNT в последней строке является точкой интереса. Легко заметить, что не все операторы приведенного фрагмента используются для ее вычисления. Неиспользуемые операторы отмечены курсивом. В частности, считаем таковыми операторы MOVE 1 TO I и ADD 1 TO I, которые нужны только для вычисления переменной из условия. Операторы PERFORM и GO TO участвуют в вычислениях неявно, осуществляя лишь передачу управления. Бизнес-правила из таких операторов не создаются.

Для рассматриваемого примера вычисления скидки последовательность бизнес-правил до выделения бизнес-процедур состоит из 9 бизнес-правил.

1. **MOVE 0 TO DISCOUNT-FACTOR1 DISCOUNT-FACTOR2**  
Выход: DISCOUNT-FACTOR1, DISCOUNT-FACTOR2
2. *При условиях: UPDATE-NEEDED = "TRUE", CURDAY > 5*  
**MOVE PRICE-0 TO PRICE**  
Вход: PRICE-0  
Выход: PRICE
3. *При условиях: UPDATE-NEEDED = "TRUE", NOT CURDAY > 5*  
**COMPUTE PRICE = PRICE-0 + 10**  
Вход: PRICE-0  
Выход: PRICE
4. *При условиях: NOT I > MAX*  
**COMPUTE RES = DISCOUNT-FACTOR1 + CONST**  
Вход: DISCOUNT-FACTOR1  
Выход: RES
5. *При условиях: NOT I > MAX*  
**ADD CONSTS TO RES**  
Вход/Выход: RES
6. *При условиях: NOT I > MAX*  
**COMPUTE DISCOUNT-FACTOR1 = RES \* 0.25**  
Вход: RES  
Выход: DISCOUNT-FACTOR1
7. *При условиях: UPDATE-NEEDED = "TRUE", CURDAY > 5*  
**COMPUTE DISCOUNT-FACTOR2 = PRICE-0 - 50**  
Вход: PRICE-0  
Выход: DISCOUNT-FACTOR2
8. **COMPUTE DISCOUNT-FACTOR2 = 0.5 \* DISCOUNT-FACTOR2**  
Вход/Выход: DISCOUNT-FACTOR2
9. **COMPUTE DISCOUNT = DISCOUNT-FACTOR1 \* PRICE + DISCOUNT-FACTOR2**  
Вход: DISCOUNT-FACTOR1, PRICE, DISCOUNT-FACTOR2  
Выход: DISCOUNT

Результат группирования по условиям приведен ниже. Бизнес-процедуры обозначены римскими цифрами. Триггеры имеют вид «CALL номер группы».

- I. 1) **MOVE 0 TO DISCOUNT-FACTOR1 DISCOUNT-FACTOR2**  
2) *При условиях: NOT I > MAX*  
**CALL II.**  
3) *При условиях: UPDATE-NEEDED = "TRUE"*  
**CALL III.**  
4) **COMPUTE DISCOUNT-FACTOR2 = 0.5 \* DISCOUNT-FACTOR2**  
5) **COMPUTE DISCOUNT = DISCOUNT-FACTOR1 \* PRICE + DISCOUNT-FACTOR2**
- II. 1) **COMPUTE RES = DISCOUNT-FACTOR1 + CONST**  
2) **ADD CONSTS TO RES**  
3) **COMPUTE DISCOUNT-FACTOR1 = RES \* 0.25**
- III. 1) *При условиях: NOT CURDAY > 5*  
**COMPUTE PRICE = PRICE-0 + 10**  
2) *При условиях: CURDAY > 5*  
**CALL IV.**
- IV. 1) **MOVE PRICE-0 TO PRICE**  
2) **COMPUTE DISCOUNT-FACTOR2 = PRICE-0 - 50**

Легко заметить, что полученная программа на языке бизнес-правил эквивалентна исходной программе, однако является более структурированной.

Существенным является также тот факт, что в общем случае при создании бизнес-процедур исключается их повторение: если бизнес-процедура с необходимым содержимым уже существует, триггер устанавливается на нее. Это сокращает количество созданных бизнес-правил и упрощает восприятие.

Раздел 3.2 посвящен *методу валидации экранных портов*. Задачей валидации экранных портов мы назвали задачу поиска всех проверок корректности значений, вводимых пользователем и попадающих в программу через операторы обработки экранных форм – экранные порты. Такая постановка требует некоторого уточнения. Под корректностью значений можно понимать проверку соответствия типов или попадание значения в заданный диапазон. Но поскольку, вообще говоря, нам неизвестно, какие именно проверки проводились в исходной программе, рассматриваются всевозможные сравнения значений, переданных в программу с экранной формы, с любыми другими значениями. Также заметим, что проверка значения на корректность может производиться не сразу, а после некоторых преобразований. Задача решалась для языка COBOL, хотя метод применим и для других языков.

С учетом вышесказанного задача переформулируется более точно: в программе на языке COBOL для указанного экранного порта RECEIVE необходимо найти все условные операторы, в которые можно попасть из переменных данного порта по информационным зависимостям. Также требуется указать последовательности операторов, ведущие из порта в условный оператор.

Задача решается путем построения операторного графа на основе специального информационного графа и создания бизнес-правил с помощью анализа построенного графа.

Заключительный раздел 3.3 третьей главы посвящен *методу поиска использований заданной переменной*. Задача поиска использований является в некотором смысле противоположной задаче восстановления вычислений, и поэтому для ее решения предлагается использовать аналогичные алгоритмы со сменой ориентации графа. Однако при этом

возникают некоторые сложности, исследование и разрешение которых является нашей дальнейшей целью.

**В четвёртой главе** описано реализованное программное средство Business Rule Manager. Этот компонент системы Relativity Modernization Workbench предназначен для создания и изменения бизнес-правил. Подобные инструменты называют *системами управления бизнес-правилами* (Business Rules Management Systems). Они должны удовлетворять следующим требованиям:

- предоставлять сервисы для хранения, просмотра, изменения, защиты и восстановления данных, называемые сервисами репозитория;
- управлять жизненным циклом бизнес-правила: создание, активация, замена, удаление и т.п.;
- проверять бизнес-правила на соответствие стандартам (логическая корректность) и проверять их действенность в бизнесе.

В данной главе демонстрируется, что Business Rule Manager удовлетворяет перечисленным требованиям и, кроме того, имеет дополнительные возможности.

Раздел 4.1 посвящен общему знакомству с инструментом. Здесь вводится необходимое для понимания работы инструмента представление бизнес-правил, подробнее рассматривается BRM-модель, упомянутая в первой главе, и описывается интерфейс пользователя.

Сначала рассматривается представление бизнес-правил и BRM-модель как часть модели репозитория, используемой в системе Relativity Modernization Workbench. Репозиторий служит для хранения информации о рассматриваемой программной системе, и модель репозитория определяет его структуру. Кроме того, модель используется средствами доступа к базе данных репозитория, поскольку запросы к этой базе сформулированы в терминах модели, что обеспечивает дополнительный уровень абстракции. Концептуально модель репозитория делится на два уровня. Модель первого уровня определяет общую структуру приложения. Второй уровень модели репозитория представлен набором подмоделей, специфичных для конкретного языка программирования, и

определяет структуру программы на уровне отдельных операторов. BRM-модель относится к первому уровню модели репозитория.

Чтобы описать интерфейс пользователя, следует отметить, что Business Rule Manager является одним из множества компонентов, связанных в общую систему HyperCode. Все эти компоненты работают совместно. Иными словами, при выборе бизнес-правила на панели Business Rule Manager соответствующий сегмент кода автоматически показывается на всех остальных компонентах системы HyperCode, где это имеет смысл. Пользователь может загружать и закрывать самые разные компоненты без потери контекста своей работы, создавая при этом самые различные отчеты, диаграммы и, безусловно, бизнес-правила. Одновременно допускается работа с множеством различных конфигураций системы HyperCode, каждая из которых открывается в собственном окне. Такая возможность позволяет параллельно исследовать несколько программ. Помимо HyperCode, в системе Modernization Workbench имеются и другие инструменты, в том числе и для исследования межпрограмного взаимодействия, на работе которых мы останавливаться не будем.

Возможности инструмента Business Rule Manager перечислены в разделе 4.2, они включают:

- поиск по различным атрибутам бизнес-правил;
- групповое редактирование бизнес-правил;
- копирование и перемещение бизнес-правил;
- валидацию сегментов – проверку на соответствие сегмента бизнес-правила и отредактированного кода;
- автоматический поиск входных и выходных данных бизнес-правила;
- подстановку терминов предметной области вместо имен переменных (с использованием специального бизнес-словаря);
- автоматическое построение бизнес-правил;
- экспорт и импорт бизнес-правил с использованием формата XML.

Более подробно каждая функциональность описана в соответствующем подразделе. Автоматическое построение бизнес-правил, суть которого была описана в предыдущей главе, вынесено в отдельный раздел 4.3, где рассматриваются особенности реализации его алгоритмов.

Раздел 4.4 содержит данные о тестировании инструмента и его основных алгоритмов на реальных приложениях. В частности, рассматриваются вопросы производительности.

В разделе 4.5 проведено сравнение Business Rule Manager с другими программными средствами, упомянутыми в первой главе.

**В заключении** описаны основные результаты работы, вклад автора в проделанную работу и идеи дальнейшего развития проекта.

### **ОСНОВНЫЕ РЕЗУЛЬТАТЫ**

- Разработан метод восстановления вычислений заданной программной переменной на основе графа информационных зависимостей программы.
- Разработан метод валидации экранных портов для построения бизнес-правил из всех проверок корректности значений, вводимых пользователем и попадающих в программу через операторы обработки экранных форм – экранные порты.
- Предложен вспомогательный алгоритм для валидации сегментов бизнес-правил, позволяющий проверять актуальность бизнес-правил после редактирования программного кода.
- Все разработанные алгоритмы реализованы в рамках программного средства анализа бизнес-логики старых приложений Business Rule Manager.

### **О ЛИЧНОМ ВКЛАДЕ АВТОРА**

Теоретическая часть работы выполнялась автором совместно с научным руководителем. Практическая реализация алгоритмов автоматического извлечения бизнес-правил, интерфейса пользователя и некоторых алгоритмов в системе Business Rule Manager выполнена автором лично с применением технологических решений системы Modernization Workbench, разработанной совместно российской



компанией «Ланит-Терком», американской компанией Relativity Technologies и лабораторией смешанных вычислений Института систем информатики СО РАН. Общий объем программного кода составляет около 8000 строк (или 350 Кб).

#### ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. **Бульонков М.А., Емельянов П.Г., Тарабухина Н.К.** Business Rule Manager – средство анализа бизнес-логики старых приложений // Вестник Санкт-Петербургского университета. Серия 10: Прикладная математика, информатика, процессы управления. Вып. 1. / СПб.: Изд-во С.-Петерб. Гос. Ун-та, 2010. – С. 23-34.
2. **Бульонков М.А., Тарабухина Н.К.** Валидация экранных портов в программах на языке COBOL // VIII Всероссийская конференция молодых ученых по математическому моделированию и информационным технологиям: Программа и тезисы докладов / Институт вычислительных технологий СО РАН. Новосибирск, 2007. – С. 87-88. (Электронная публикация доклада: <http://www.nsc.ru/ws/УМ2007/12731/Tarabukhina.htm>)
3. **Вольхина Н.К.** Автоматическое восстановление бизнес-логики программ // Молодая информатика. Вып. 2. – Новосибирск, 2006. – С. 90-102.
4. **Тарабухина Н.К.** Business Rule Manager – средство восстановления бизнес-логики программ // Седьмая международная конференция "Перспективы систем информатики". Рабочий семинар "Наукоемкое программное обеспечение": Инф. бюллетень. – Новосибирск, 2009. – С. 266-272.
5. **Вольхина Н.К.** Корректное упорядочение бизнес-правил на основе информационных зависимостей // Технологии Microsoft в теории и практике программирования: Тезисы докладов. – Новосибирск, 2007. – С. 4-6.
6. **Вольхина Н.К.** Поиск по образцу для валидации бизнес-правил // Технологии Microsoft в теории и практике программирования: Тезисы докладов. – Новосибирск, 2006. – с. 10-11.

7. **Вольхина Н.К.** Группировка бизнес-правил для улучшения понимания программы. // Материалы XLV Международной научной студенческой конференции "Студент и научно-технический прогресс": Математика / Новосиб. Гос. Ун-т. Новосибирск, 2007. – С. 147-148.

Тарабухина Н. К.

МЕТОДЫ ИЗВЛЕЧЕНИЯ БИЗНЕС-ЛОГИКИ  
НА ОСНОВЕ СЕМАНТИЧЕСКИХ СВОЙСТВ ПРОГРАММ

Автореферат

---

Подписано в печать

Объем 1,1 уч.-изд. л.

Формат бумаги 60 × 90 1/16

Тираж 100 экз.

Отпечатано в ЗАО РИЦ «Прайс-курьер»

630090, г. Новосибирск, пр. Ак. Лаврентьева, 6, тел. 334-22-02

Заказ №135