Российская Академия Наук
Институт Проблем Информатики

На правах рукописи

Ступников Сергей Александрович

Моделирование композиционных уточняющих спецификаций

05.13.17 — Теоретические основы информатики

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Москва 2005

Работа выполнена в лаборатории Композиционных методов проектирования информационных систем Института проблем информатики РАН.

Научные руководители —	доктор физико-математических наук, профессор Л. А. Калиниченко
	доктор технических наук, профессор В. А. Сухомлин
Официальные оппоненты —	доктор физико-математических наук А. К. Петренко
	кандидат физико-математических наук Р. Э. Яворский
Ведущая организация —	Институт Проблем Управления РАН

Защита диссертации состоится „ __“ _____ 200_ г. в __ ч. __ мин. на заседании диссертационного совета Д.002.073.01 при Институте проблем информатики РАН по адресу: 119333 г. Москва, ул. Вавилова, 44, корп. 2.

С диссертацией можно ознакомиться в библиотеке Института проблем информатики РАН.

Отзывы в одном экземпляре, с заверенной подписью, просим направлять по адресу: 119333, Москва, ул. Вавилова, 44, корп. 2, в диссертационный совет.

Автореферат разослан „ ___“ _____ 200_ г.

Ученый секретарь
диссертационного совета Д.002.073.01
доктор технических наук,

С. Н. Гринченко

Общая характеристика работы

Актуальность темы. В настоящее время наблюдается устойчивая тенденция к все большему вовлечению формальных методов в процесс разработки информационных систем (ИС). Это связано с увеличением количества задач, для решения которых необходимо *доказательное* рассуждение о свойствах систем. Именно для решения такого рода задач и предназначены опирающиеся на математическую логику и алгебру формальные методы, представляющие собой совокупность языков, технологий и инструментальных средств спецификации и верификации сложных ИС. *Спецификация* ИС есть описание системы и ее желаемых свойств на некотором языке спецификаций. *Верификация* разрабатываемой системы есть ее формальная проверка на соответствие заданным требованиям. В контексте методов верификации систем формализуется одно из наиболее важных понятий в области формальных методов – понятие *уточнения*. В настоящей работе уточнение понимается следующим образом: система B *уточняет* систему A, если пользователь может использовать систему B вместо системы A, не замечая факта замены A на B.

Формальная спецификация, верификация и теория уточнения с успехом применялись в большом количестве проектов по разработке ИС в различных областях: медицине, ядерной энергетике, телефонии, транспорте, космической технике, разработке микропроцессорной техники, верификации протоколов и пр.

Тем не менее, в области применения формальных методов для разработки ИС остается много нерешенных задач, интенсивно использующих понятие уточнения. Для их решения уточнение должно быть формально доказано. Задачи требуют формализации и разработки средств для формальной верификации. В множестве таких задач можно выделить три больших класса:

1. задачи, связанные с удовлетворением специфических нефункциональных требований к ИС;
2. задачи интеграции множественных неоднородных источников данных и сервисов;
3. задачи композиции ИС из существующих программных и информационных компонентов в интероперабельных средах, таких как Web Services, Grid и различных видах промежуточного слоя, расположенного между операционными системами и прикладными системами.

К первому классу относятся, например, задачи разработки систем, ошибки которых критичны для безопасности функционирования человека в таких

системах (safety-critical), или задачи разработки *отказоустойчивых* (fault-tolerant) систем, способных продолжать работу при наличии сбоев.

Задачи интеграции неоднородных источников (сервисов) и задачи композиции ИС из компонентов становятся все более актуальными в настоящее время, когда развиваются и появляются новые технологии промежуточного слоя (CORBA, Java RMI, .NET, Web Services, Semantic Web, Grid и другие). В рамках этих технологий накоплено большое количество программных и информационных технически интероперабельных компонентов. Технологии промежуточного слоя обеспечивают техническую возможность интеграции источников и конструирования распределенных, интероперабельных ИС из компонентов; позволяют накапливать репозитории компонентов для их дальнейшего использования при создании новых ИС. Ввиду широкой распространенности этих технологий, необходимы методы достижения *семантической интероперабельности* компонентов. Понятие семантической интероперабельности означает комбинацию способностей решения следующих вопросов: релевантности имеющихся компонентов разрабатываемому применению, соответствии их прикладных контекстов контексту применения, а также также непротиворечивости интероперабельной композиции ресурсов в контексте разрабатываемого применения.

Решение задач интеграции неоднородных источников и сервисов может быть основано на идее *предметных посредников*, предназначенных для преобразования несистематизированного набора информационных источников, предоставляемых различными провайдерами, в хорошо структурированную коллекцию, поддерживаемую интегрированными однородными спецификациями.

Посредник обеспечивает пользователей метаинформацией, однородно представляющей предметный контент охватываемых источников, а также поддерживает унифицированную (каноническую) информационную модель. Такая модель необходима для однородного представления разнообразных моделей представления информации, используемых в неоднородных источниках. Каноническая модель является *расширяемой* – ядро модели фиксировано, и для каждой модели информационного источника M строится такое расширение ядра E, что E *уточняется* моделью M. В случае баз данных под *моделью информационного источника* подразумевается совокупность языка определения данных и языка манипулирования данными, которые использовались для спецификации источника.

Объединение таких расширений ядра составляет синтез канонической модели посредника для моделей источников. На основании одного и того же

ядра возможен синтез множества канонических моделей для различных наборов информационных источников.

Посредник поддерживает процесс систематической регистрации и классификации источников, содержит унифицированную онтологическую и мета-информацию для обнаружения и композиции существующих источников. Регистрация неоднородных источников в посреднике, согласование онтологий посредника и информационных источников, а также поиск подходящих информационных источников существенным образом опираются на уточнение структурных и онтологических спецификаций посредника структурными и онтологическими спецификациями источников.

Для преобразования запросов в терминах посредника в запросы к конкретным источникам [3], а также для обратного преобразования результата, полученного от источников, служат *адаптеры* информационных источников. Генерация доказательно правильных адаптеров также базируется на уточнении спецификаций посредника спецификациями источников.

Основная идея *композиционного проектирования* состоит в том, чтобы построить корректную композицию спецификаций существующих компонентов (информационных, программных), уточняющую спецификацию требований к разрабатываемой ИС. При этом спецификации требований и существующих компонентов представляются в канонической модели. Реальные компоненты реализуются в разнообразных языках программирования, моделях данных. Техническая интероперабельность неоднородных компонентов достигается применением архитектур и компонентных моделей, подобных CORBA. Тем самым технически обеспечивается возможность композиции компонентов. В целях проектирования спецификации компонентов приводятся к однородному представлению в канонической модели. Предполагается также, что и спецификация требований представляется в канонической модели (хотя для этого может потребоваться преобразование в такую модель из некоторого другого языка спецификаций, например из UML). В настоящей работе рассматриваются следующие виды композиции: *соединение* и *пересечение* абстрактных типов данных[1]. Неформально, соединение $T_1 \sqcup T_2$ спецификаций типов T_1 и T_2 включает всю информацию, содержащуюся в спецификациях T_1 и T_2; пересечение $T_1 \sqcap T_2$ включает лишь общую информацию из спецификаций T_1 и T_2.

Методы решения вышеперечисленных задач разрабатывались в течении ряда лет в Лаборатории композиционных методов проектирования инфор-

[1]Понятие *абстрактного типа данных* рассматривается в разделе автореферата, посвященном содержанию главы 2 диссертационной работы.

мационных систем Института проблем информатики РАН. Для однородного представления разнообразных информационных источников, описания моделей предметных областей, проектирования и программирования ИС в интероперабельных средах было разработано ядро канонической модели представления информации – язык СИНТЕЗ. Для достижения всех указанных целей в языке СИНТЕЗ совместно используются парадигмы моделей представления знаний о предметных областях и спецификаций требований к ИС, моделей концептуального проектирования ИС, объектно-ориентированных моделей данных, логического программирования в дедуктивных базах данных, систем управления неоднородными мультибазами данных, предикативных спецификаций ИС.

Были также разработаны методы расширения канонической модели, методы и средства композиционного проектирования ИС, методы и средства интеграции множественных неоднородных источников. Необходимо заметить, что использование языка СИНТЕЗ в качестве ядра канонической модели не предполагает отказа от распространенных методов и моделей, таких как OMT, UML: с их помощью могут осуществляться анализ ИС и обратная инженерия. Для решения более сложных задач – интеграции информационных источников и композиционного проектирования ИС – требуется более точная информационная модель. Поэтому такие модели, как OMT и UML должны быть отображены в каноническую информационную модель [8].

Необходимым требованием к разработанным методам и средствам являлось обеспечение возможности автоматизированного доказательства факта уточнения спецификации требований композицией спецификаций существующих компонентов.

Цель и задачи работы. Целью диссертационной работы является исследование и разработка формальных оснований автоматизации доказательства уточнения полных спецификаций требований спецификациями компонентов и их композиций в процессе интеграции множественных неоднородных источников данных и сервисов и композиционного проектирования ИС.

Достижение цели предполагает решение следующих задач:

1. для проведения доказательных рассуждений о моделях информационных ресурсов, например, рассуждений об их непротиворечивости, об уточнении или отображении моделей, разработать метод формального определения канонических информационных моделей и на его основе определить формальную семантику ядра канонической модели (языка СИНТЕЗ);

2. для автоматизации формального доказательства уточнения полных спецификаций требований спецификациями компонентов, а также доказательства непротиворечивости спецификаций, разработать метод отображения канонических информационных моделей в теоретико-модельный формальный язык спецификаций и на его основе – алгоритмы отображения ядра канонической модели в формальный язык. В качестве такого языка в настоящей работе выбрана Нотация Абстрактных Машин (Abstract Machine Notation, AMN), что позволит использовать существующую технологию доказательства уточнения (B-technology) и инструментальные средства доказательства уточнения (B-Toolkit, Antelier B) для доказательства уточнения спецификаций канонической модели;
3. разработать метод доказательства корректности отображения информационных моделей с использованием их денотационно-аксиоматической семантики; на основе этого метода доказать корректность разработанных алгоритмов отображения ядра канонической модели в AMN;
4. на основе этих алгоритмов разработать инструментальное средство автоматического отображения спецификаций канонической модели в AMN и методику использования этого средства совместно с B-Toolkit при решении практических задач проектирования ИС.

Методы исследования. При решении поставленных в работе задач использовались методы денотационной и аксиоматической семантики, методы теории множеств и логики предикатов, методы теории уточнения спецификаций.

Научная новизна и результаты, выносимые на защиту. В диссертационной работе получены следующие новые научные результаты:

- метод формального определения канонических информационных моделей и на его основе – формальная денотационно-аксиоматическая семантика языка СИНТЕЗ;
- метод отображения канонических информационных моделей в язык AMN и на его основе – алгоритмы отображения языка СИНТЕЗ в AMN;
- метод доказательства корректности отображения информационных моделей с использованием их денотационно-аксиоматической семантики; с использованием метода было построено доказательство корректности алгоритмов отображения языка СИНТЕЗ в AMN;
- программное средство автоматического отображения спецификаций ядра канонической модели в AMN;

- методика использования формального аппарата доказательства уточнения при интеграции множественных неоднородных источников и при композиционном проектировании ИС.

Практическая ценность. Разработанные методы и средства созданы как составная часть инструментария эксперта-конструктора, осуществляющего композиционное проектирование ИС как в рамках локальных, так и в рамках распределенных библиотек компонентов. При этом процесс проектирования ИС становится формально верифицируемым, что позволяет корректно использовать существующие компоненты, уменьшает время отладки и тестирования систем, позволяет проектировать системы, надежность которых критична, и отказоустойчивые системы.

Разработанные методы и средства предназначены также для встраивания в качестве составной части средств интеграции неоднородных информационных источников и сервисов для решения таких задач, как: синтез канонических моделей для посредников над разнородными информационными источниками; согласование онтологий посредника и информационных источников; поиск информационных источников и сервисов, семантически релевантных заданным требованиям; регистрация неоднородных информационных источников в предметных посредниках; генерация доказательно правильных адаптеров информационных источников.

Реализация результатов работы. Результаты диссертационной работы использованы в проектах РФФИ 01-07090084, 03-01-00821, 05-07-90413-в; проекте № 1-10 программы фундаментальных исследований ОИТВС РАН "Фундаментальные основы информационных технологий и систем", НИР Контекст "Контекстуализация неоднородных информационных источников в предметном информационном посреднике", НИР И^{3}НИ "Композиционные методы решения задач в инфраструктурах интеграции информации для научных исследований".

Разработанные инструментальные средства нашли применение в качестве составной части инструментария эксперта-конструктора, реализующего композиционное проектирование ИС, и прототипа средств поддержки предметных посредников.

Апробация работы. Основные результаты диссертации докладывались на Международных конференциях ADBIS (Братислава 2002, Будапешт 2004), на XXIV Конференции молодых ученых механико-математического факультета МГУ им. М.В. Ломоносова (Москва, 2002), на Международном симпозиуме по базам данных конференции VLDB (Берлин 2003), на II научной сессии

ИПИ РАН (Москва, 2005), на научных семинарах Лаборатории композиционных методов проектирования информационных систем Института проблем информатики РАН.

Публикации. По теме диссертации автором опубликовано 9 работ. Список работ приведён в конце автореферата.

Структура и объем работы. Диссертация состоит из введения, пяти глав, заключения, списка литературы и четырех приложений. Основное содержание работы изложено на 157 страницах текста, включая 7 рисунков и 3 таблицы. Список литературы содержит 104 наименования.

Содержание работы

Во введении обосновывается актуальность темы диссертационной работы, формулируется цель, научная новизна и практическая значимость полученных результатов, дается краткое содержание глав работы.

В первой главе изложены основные черты ядра канонической информационной модели (языка СИНТЕЗ) и мотивация формального определения канонической модели. Рассмотрен обзор основных существующих методов формального определения информационных моделей, близких к ядру канонической модели.

Единицей определения языка СИНТЕЗ является *модуль*. Каждый модуль задает обобщенное представление информационных источников, либо является модулем спецификации предметной области или концептуального проекта ИС. Язык содержит унифицированную систему типов, включающую универсальный конструктор типов (Абстрактный Тип Данных, АТД), а также представительный набор встроенных типов. Описание абстрактного типа данных инкапсулирует спецификации атрибутов, методов и инвариантов типа. Методы и инварианты типов описываются встроенным *типом функции*. АТД может быть *объектным* (экземпляры типа – объекты – при модификации сохраняют идентифицируемость) или необъектным (экземпляры типа представляют собой неизменчивые значения). Спецификация типа функции включает описание параметров функции и предикативную спецификацию функции. Для задания предикативных спецификаций функций в канонической модели используется язык логических формул многосортного объектного исчисления.

В течении последних лет в лаборатории композиционных методов проектирования ИС ИПИ РАН проводились интенсивные исследования по разработке методов решения задач над неоднородными информационными источниками, таких как композиционное проектирование ИС, регистрация источ-

ников в предметных посредниках, интеграция моделей источников, переписывание запросов в среде посредников [3]. В процессе этих исследований постоянно проявлялась необходимость формального определения канонической объектной информационной модели (при использовании языка СИНТЕЗ в качестве ее ядра) при манипулировании разнообразными информационными моделями – задания ее (канонической модели) формальной семантики. Формальная семантика необходима для проведения доказательных рассуждений о моделях информационных ресурсов. Имея формальную семантику, можно строго формулировать утверждения о непротиворечивости и уточнении спецификаций, выраженных в модели. Появляется возможность доказательства корректности отображений одних моделей информационных ресурсов в другие модели. В частности, синтез канонической модели строится как расширение ее ядра для каждой конкретной модели информационного ресурса так, что ядро вместе с таким расширением должно уточняться этой конкретной моделью. Для таких манипуляций информационными моделями и выраженными в них спецификациями требуется выражение их семантики в некотором формальном языке, позволяющем осуществлять доказательство непротиворечивости и уточнения спецификаций. В настоящей работе в качестве такого языка выбрана Нотация Абстрактных Машин (AMN), которая позволяет осуществлять необходимые рассуждения относительно спецификаций. Для AMN разработаны специальные инструментальные средства, составляющие в совокупности так называемую B-технологию. Для того, чтобы можно было манипулировать спецификациями информационных моделей в рамках B-технологии (в частности, доказывать корректность отношения уточнения между спецификациями, выраженными в различных моделях), необходимо корректно отобразить такие модели в язык AMN с сохранением семантики. В настоящей работе создан метод такого отображения и показано, как его следует применять на примере ядра канонической информационной модели (языка СИНТЕЗ).

Наличие формальной семантики ядра канонической модели также позволяет определить формальную семантику его расширений при синтезе канонической модели следующим образом. Расширение означает выделение таких обобщенных (параметризованных) конструкций канонической модели и таких аксиом, что данные конструкции, ограниченные аксиомами, уточняются соответствующими конструкциями исходной модели. При этом выделяемые конструкции и аксиомы описываются в синтаксисе ядра канонической модели. А потому их семантика определяется при помощи семантики ядра. Для доказательства факта уточнения расширения ядра исходной моделью

могут быть использованы инструментальные средства доказательства уточнения (например, B-технология). Их использование будет доказательно правильным, если доказана корректность отображения ядра в соответствующую модель (например, AMN).

Таким образом, одной из важнейших проблем моделирования композиционных уточняющих спецификаций является формализация соответствующих им языковых конструкций. В разделе 1.2 приведен обзор основных существующих методов формального определения информационных моделей, близких к ядру канонической модели. Цель обзора состояла в выявлении общих закономерностей и особенностей формального определения информационных моделей, которые могут быть использованы при формальном определении ядра канонической модели. Рассмотрены также специфические черты канонической модели, не представленные или слабо представленные в существующих информационных моделях, потребовавшие разработки адекватных методов формального определения. Среди таких особенностей: определенные в канонической модели операции композиции типов и решетка типов; четкое разделение понятий *типа данных*, как описания структуры и поведения значений типа, и *коллекции*, как множества значений некоторого типа; иерархия обобщения-специализации на множестве типов и на множестве коллекций; предикативные спецификации функций – формулы, задающие смешанные пред и постусловия функции и содержащие предикаты-коллекции, вызовы функций (в том числе с побочными эффектами), кванторы существования и всеобщности; язык правил (использующихся, в частности, как запросы к коллекциям), содержащий операции соединения, пересечения, объединения, селекции и проекции объектных и необъектных коллекций.

В разделе 1.3 рассмотрены основные существующие методы формализации уточнения, связанные с ними технологии и инструментальные средства.

В разделе 1.4 рассмотрены существующие подходы к моделированию в AMN информационных моделей, близких к ядру канонической модели. Указано, что ни один из известных подходов не ориентирован на доказательство уточнения с использованием B-технологии. Выделены основные черты, отличающие представленный в настоящей работе метод моделирования спецификаций в AMN от других существующих методов: однородность (AMN-спецификация, полученная в результате отображения канонической спецификации, может быть использована и как уточняющая спецификация, и как уточняемая); использование усовершенствованного *экстенсионального принципа* моделирования АТД; представление в AMN специфических черт канонической модели.

Во второй главе рассматривается метод формального определения канонических информационных моделей. Метод демонстрируется на примере построения формальной семантики подмножества ядра канонической модели (языка СИНТЕЗ), включающего основные средства представления типов и коллекций. Синтаксис подмножества изложен в приложении.

Формальное определение ядра заключается в сообщении канонической модели комбинированной денотационно-аксиоматической семантики. Методология денотационной семантики в обобщенном смысле предполагает отображение синтаксических конструкций модели в математические объекты: целые числа, логические значения, функции и т.д. В настоящей работе методы денотационной семантики используются для отображения синтаксиса модели в объекты теории множеств и логики предикатов первого порядка.

Методы денотационной семантики удобны для определения пространства состояний системы, задаваемой спецификацией модели. Множество допустимых состояний системы задается при помощи наложения ограничений $\mathbb{R}$ на пространство состояний $\mathbb{S}$. Множество ограничений $\mathbb{R}$ представляет собой теорию (множество формул) первого порядка, формулы которой определены на $\mathbb{S}$. Кроме $\mathbb{R}$, в аксиоматическую часть семантики модели входит множество предикатов $\mathbb{F}$, отвечающих спецификациям методов АТД и спецификациям функций, не являющихся методами АТД. Таким образом, семантикой спецификации модели является совокупность пространства состояний $\mathbb{S}$, теории $\mathbb{R}$, а также множества предикатов $\mathbb{F}$.

Построение денотационной семантики модели начинается с так называемого *конкретного* синтаксиса. Конкретный синтаксис, изложенный в приложении, определяет точную синтаксическую и фразовую структуры спецификаций. *Абстрактный* синтаксис, определенный в разделе 2.1, используется для того, чтобы категоризировать типы синтаксических структур модели. Абстрактный синтаксис модели представляет собой набор продукций, в левой части которых стоят названия синтаксических доменов, а в правой – выражения, включающие имена синтаксических доменов, элементарные синтаксические домены и операторы конструирования синтаксических доменов. В число операторов входят декартово произведение $\times$, сумма $+$ и список $(\cdot)^\star$.

С синтаксическими доменами ассоциированы метапеременные, которые используются вместо элементов соответствующих доменов при описании семантических функций. Например, синтаксическому домену *ModuleSpec* (спецификация модуля – основной единицы определения канонической модели) соответствует переменная MS:

$MS : ModuleSpec = TypeSection \times FunctionSection \times IRSection$

Спецификация модуля состоит из секции типов *TypeSection*, секции функций *FunctionSection* и секции описания информационных источников *IRSection*.

В разделе 2.2 излагается основная идея, на которой базируются принципы построения пространства состояний – идея экстенсиональной интерпретации абстрактных типов данных.

Определение. *Абстрактный тип данных* T есть тройка $\langle V_T, O_T, I_T \rangle$, где V_T – экстенсионал типа, O_T – множество операций типа, I_T – инвариант типа.

Спецификацию типа в канонической модели составляют O_T и I_T, экстенсионал V_T используется для придания формальной семантики спецификации типа и является предопределенным множеством (семантическим доменом). Различается *собственный* и *полный* экстенсионалы типа. Собственным экстенсионалом V_T^p типа T называется множество значений типа T, не являющихся значениями никакого подтипа типа T. Полным экстенсионалом V_T типа T называется объединение V_T^p и полных экстенсионалов всех подтипов типа T. Объединение экстенсионалов всех типов есть множество $\mathcal{AVAL}$ (Abstract VALues).

Множество операций O_T представляет собой объединение множеств атрибутов и методов $A_T \cup F_T$. Семантически, атрибут представляется константной функцией $a : V_T \to E(a)$. При этом $E(a)$ – множество значений, которое может принимать атрибут. Например, если тип a – целочисленный, то $E(a) = \mathbb{Z}$ (множество целых чисел), если тип a – необъектный абстрактный тип T, то $E(a) = V_T$, если тип a – объектный абстрактный тип, то $E(a) = \mathbb{U}$, где $\mathbb{U}$ – предопределенное множество (семантический домен) объектных идентификаторов.

Состояние системы, задаваемой спецификацией модуля канонической модели, характеризуется составом источников, определенных в спецификации, а также состоянием объектов. Состав источника с экземплярами типа T есть некоторое подмножество экстенсионала V_T. Состав всех источников в совокупности задается функцией $\mathcal{C} \in [I_C \to \mathbb{P}(\mathcal{AVAL})]$, где I_C - множество имен коллекций, определенных в спецификации. Состояние объектов задается функцией $\mathcal{S} \in [\mathbb{U} \nrightarrow \mathcal{AVAL}]$. Пространство состояний системы, которому принадлежит упорядоченная пара функций $\mathcal{S} \mapsto \mathcal{C}$, обозначается $\mathbb{S} = [\mathbb{U} \nrightarrow \mathcal{AVAL}] \times [I_C \to \mathbb{P}(\mathcal{AVAL})]$ (декартово произведение множества частичных функций из $\mathbb{U}$ в $\mathcal{AVAL}$ и множества тотальных функций из I_C в множество подмножеств множества $\mathcal{AVAL}$).

Следующий элемент денотационной семантики – семантические домены – определяют абстрактные математические объекты, которые мы хотим счи-

тать семантикой спецификации. На основе элементарных доменов, например, множества логических значений $\mathbb{B} = \{true, false\}$, при помощи операторов конструирования доменов строятся производные семантические домены. В число операторов входят декартово произведение $\times$, сумма $+$, конструктор множества подмножеств $\mathbb{P}(\cdot)$, конструктор упорядоченной пары $\cdot \mapsto \cdot$, конструктор функционального домена $[\cdot \to \cdot]$.

Семантические функции отображают абстрактные синтаксические структуры в соответствующие семантические объекты. Семантические функции, формирующие пространство состояний, являются, по существу, сложными конструкторами семантических доменов, и само пространство состояний представляет собой сложный семантический домен. Семантические функции задаются при помощи сигнатур и семантических соотношений. Сигнатура определяет имя и область определения функции. Семантические соотношения определяют, каким образом функция действует на том или ином синтаксическом домене. Аргументы семантических функций – синтаксические объекты – заключаются в специальные скобки $[\![\cdot]\!]$, для того, чтобы отделить их от семантических объектов. Семантические домены и семантические функции построения пространства состояний описаны в разделе 2.3. Например, семантическая функция *sModuleSpec*, формирующая пространство состояний по спецификации модуля, определяется следующим образом:

$$sModuleSpec[\![TS\ FS\ IRS]\!] \triangleq sTypeSection[\![TS]\!] \times sIRSection[\![IRS]\!]$$

Сигнатура функции состоит из имени – *sModuleSpec* и области определения – синтаксического домена *ModuleSpec*. Метапеременные TS, FS, IRS принимают значения из синтаксических доменов *TypeSection*, *FunctionSection* и *IRSection*, соответственно. Имена всех семантических функций имеют префикс s, от слова semantic – семантический. Знак $\triangleq$ используется в работе при определении семантических функций. Пространство состояний конкретного модуля определяется как декартово произведение результатов семантических функций *STypeSection* и *sIRSection*.

Итак, в разделе 2.3 показано, каким образом семантические функции формируют пространство состояний на основании некоторой спецификации канонической модели. Пусть семантический домен $\mathbb{S}$ есть пространство состояний, полученное на основании некоторой спецификации M.

Заметим, что система, задаваемая спецификацией M, может находиться не во всех состояниях из $\mathbb{S}$. Множество $\mathbb{S}^a$ состояний, в которых система действительно может находиться, – *множество допустимых состояний* – является собственным подмножеством $\mathbb{S}$, элементы которого удовлетворяют:

- ограничениям типизации экземпляров классов;
- ограничениям, налагаемым отношением тип-подтип;
- ограничениям, налагаемым отношением класс-подкласс;
- ограничениям инвариантов.

Ограничения, выраженные в виде формул логики предикатов первого порядка, составляют множество $\mathbb{R} = \{\phi_1, \ldots, \phi_n\}$, т.е.

$$\mathbb{S}^a = \{\mathcal{S} \mapsto \mathcal{C} \in \mathbb{S} \mid \phi_1 \wedge \ldots \wedge \phi_n\}$$

Формулы предназначены для того, чтобы выразить ту информацию, содержащуюся в спецификации модуля, которая не выражена явным образом в структуре пространства состояний. Формирование формул из $\mathbb{R}$ осуществляется семантическими функциями. В разделе 2.4 описано, что означают приведенные виды ограничений, определен вид соответствующих формул и семантические функции.

В разделе 2.5 определены семантические функции формирования предикатов, отвечающих функциям канонической модели. Рассмотрим в качестве примера спецификацию метода типа `Student`, увеличивающего стипендию студента на величину `rise` в случае, если рейтинг студента выше, чем `limit`:

```
raiseScholarship: { in: function;
 params: { +limit/integer, +rise/integer };
 { predicative:
   { this.rating >= limit -> this.scholarship'= this.scholarship+rise }
 };
}
```

Действие метода задается предикативной спецификацией, описывающей смешанные предусловия и постусловия метода. Для выражения условий, связанных с изменением состояния ИС, ссылки на значения переменных при завершении метода изображаются идентификаторами переменных, выделенными следующими после них апострофами. Переменные без апострофа ссылаются на исходное состояние ИС. Таким образом, предикативные спецификации методов это формулы не просто над $\mathbb{S}$, но над $\mathbb{S}\times\mathbb{S}$, где первый компонент отвечает за исходное состояние ИС, второй – за конечное. Кроме того, у метода есть параметры, не связанные с состоянием ИС (`limit`, `rise`), а потому для того, чтобы полностью определить пространство, над которым действует предикативная спецификация метода, нужно добавить к $\mathbb{S} \times \mathbb{S}$ семантические домены типов параметров (множества $\mathbb{Z}$ целых чисел). Семантикой данного метода является *Student.raiseScholarship* – предикат, определенный на пространстве $\mathbb{S} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{S}$ следующим образом:

$$
\begin{array}{l}
Student.raiseScholarship(v, limit, rise) \rightleftharpoons \\
\quad v \in V_{Student} \wedge limit \in \mathbb{Z} \wedge rise \in \mathbb{Z} \wedge \\
\quad \exists\, w \bullet ((rating(v) \geq limit \Rightarrow w = scholarship(v) + rise) \wedge \\
\qquad \mathcal{S}' = \mathcal{S} \lhd\!\!\!- \{self(v) \mapsto (v \lhd\!\!\!- \{scholarship \mapsto w\})\})
\end{array}
$$

где операция перекрытия отношения r_1 отношением r_2, $r_1 \lhd\!\!\!- r_2$ определяется как $r_1 \lhd\!\!\!- r_2 = \mathrm{dom}\,(r_2 \lhd r_1) \cup r_2$, операция антиограничения отношения r множеством s – $s \lhd r = \{x, y \mid x \mapsto y \in r \wedge x \in \mathrm{dom}\, r - s\}$.

Вообще, предикативная спецификация метода может быть *формулой* либо *правилом*. Правило отличается от формулы тем, что (1) не может содержать конструкции, изменяющие состояние системы, и (2) может содержать такие операции композиции коллекций, типичные для языков запросов, как *соединение* и *объединение* коллекций. Семантика правил канонической модели рассмотрена в разделе 2.6, семантика формул – в разделе 2.7.

Разработанная формальная семантика позволяет проводить доказательные рассуждения о моделях информационных ресурсов, например, рассуждения о непротиворечивости, об уточнении или отображении моделей.

В третьей главе рассматривается метод моделирования канонических информационных моделей в языке AMN. Применение метода демонстрируется на примере определения отображения Θ абстрактного синтаксиса ядра канонической модели в спецификации AMN.

Отображение задается при помощи семантических функций, определенных на синтаксических доменах абстрактного синтаксиса канонической модели, а также при помощи алгоритмов. Модуль, как основная единица определения канонической модели, является единицей отображения канонической модели в AMN. Модуль представляется в AMN набором конструкций (абстрактных машин), состоящим из контекстной машины и машин, соответствующих АТД модуля. Контекстная машина содержит информацию, характеризующую модуль как целое, машины типов содержат информацию, характерную для отдельных типов. Таким образом, для модуля M канонической модели, $\Theta(M) = \{M_{ctxt}\} \cup M_{types}$, где M_{ctxt} – контекстная машина, M_{types} – множество машин, соответствующих АТД.

В разделе 3.1 изложены основные принципы моделирования канонических спецификаций в AMN. Моделирование АТД в AMN, также, как и формальная семантика АТД, базируется на *экстенсиональном принципе*: тип моделируется множеством своих экземпляров. Такое множество экземпляров называется *экстенсионалом* типа. В отличие от понятия экстенсионала в формальной семантике канонической модели, экстенсионал типа в AMN представляет собой

конечное множество *потенциальных* значений типа. Атрибутные функции (функции, моделирующие атрибуты) не определены изначально на потенциальных значениях. При необходимости получить значение типа с определенными значениями атрибутов, из экстенсионала типа выбирается произвольный незанятый элемент, и атрибутные функции доопределяются на нем требуемым образом.

Такой подход, в отличие от прямого моделирования в AMN семантики АТД, позволяет избежать бесконечности экстенсионалов и значительно упрощает моделирование функций в AMN операциями абстрактных машин. При этом упрощается процесс автоматического/интерактивного доказательства уточнения. Правомерность подобного подхода обоснована сохранением семантики канонической модели при отображении в AMN (доказательство этого факта приводится в главе 4 настоящей работы).

Для моделирования потенциальных экземпляров всех абстрактных типов вводится множество $AVAL$, и для каждого типа T с экстенсионалом E_T выполнено включение $E_T \subseteq AVAL$. Заметим, что поскольку $AVAL$ представляет собой множество потенциальных экземпляров типов, то оно значительно отличается от множества $\mathcal{AVAL}$, используемого для моделирования экземпляров типов при определении семантики канонической модели.

Экстенсионалы типов соотносятся в соответствии с отношением тип-подтип: для каждой пары типов T_1, T_2, где T_2 является подтипом T_1, выполнено отношение включения на экстенсионалах $E_{T_2} \subseteq E_{T_1}$. Таким образом моделируется иерархия типов.

Спецификации типов канонической модели могут ссылаться друг на друга в спецификациях атрибутов, методов, инвариантов, образуя *структуру спецификаций типов*. Основным принципом отображения структуры спецификаций типов канонической модели в AMN является стремление отобразить каждый АТД отдельной абстрактной машиной. Проблема состоит в том, что средства композиции абстрактных машин в AMN являются гораздо менее гибкими, чем средства организации структуры спецификаций типов канонической модели. Данное противоречие разрешается при помощи процедуры преобразования ориентированного графа структуры модуля M, представляющего композиционную структуру спецификаций типов модуля, — $DSG(M)$, в ориентированный граф, представляющий композиционную структуру набора абстрактных машин, — $DAMSG(M)$. Алгоритм преобразования изложен в разделе 3.1.3.

В разделе 3.2 рассмотрены правила формирования контекстной машины M_{ctxt} по спецификации модуля M канонической модели. Основное содержание

контекстной машины составляют экстенсионалы типов и отношения между экстенсионалами.

Правила формирования машин из множества M_{types} рассмотрены в разделе 3.3. Рассмотрены методы моделирования атрибутов, методов, инвариантов абстрактных типов конструкциями AMN. Так, например, атрибут a типа T моделируется функциональной переменной с именем a и областью определения E_T. Метод $meth$ типа T с входными параметрами p_k^i, выходными параметрами p_l^o, предикативной спецификацией f:

$meth$: {`in:function;`
  `params:`$\{+p_1/T_1, \ldots, +p_n/T_n, -p_{n+1}/T_{n+1}, \ldots, -p_m/T_m\}$;
  {`predicative:` $\{f\}$};
}

представляется в AMN *операцией* с именем $meth\_op$. Операцию формирует семантическая функция *mMethodSubst* (все семантические функции отображения канонической модели в AMN имеют префикс m). На вышеприведенной спецификации метода $meth$ функция действует следующим образом:

$p_{n+1}, \ldots, p_m \leftarrow meth\_op(o, p_1, \ldots, p_n) =$
 PRE $o \in ext\_T \wedge p_1 \in E(T_1) \wedge \ldots \wedge p_n \in E(T_n)$
 THEN
  $mSubstitution[\![f]\!]$
 END

Первым входным параметром операции $meth\_op$ является объект, для которого вызывается метод $meth$. Предусловием операции является корректная типизация входных параметров. Вспомогательная семантическая функция E сопоставляет типу моделирующее его в AMN множество. Семантическая функция *mSubstitution* формирует представление предикативных спецификаций канонической модели *обобщенными подстановками* (преобразователями предикатов, служащих для моделирования поведения) AMN. В разделе 3.5 рассмотрено действие функции *mSubstitution* на формулах канонической модели, в разделе 3.6 – на правилах.

Построенное отображение канонической модели в AMN позволяет использовать B-технологию для доказательства уточнения спецификаций канонической модели. Для того, чтобы сделать возможным использование B-технологии для доказательства непротиворечивости спецификаций канонической модели, в разделе 3.7 определены семантические функции преобразования конструкций AMN вида REFINEMENT в конструкции вида MACHINE.

В четвертой главе рассматривается метод доказательства корректности отображения информационных моделей с использованием их денотационно-

аксиоматической семантики. Метод демонстрируется на примере доказательства корректности алгоритмов отображения Θ ядра канонической модели в AMN.

В разделе 4.1 описаны общие принципы доказательства. Корректность Θ заключается в следующем. Спецификация модуля канонической модели, как и набор абстрактных машин AMN, задает некоторую систему, представляющую собой множество значений АТД, объединенных в коллекции. Система характеризуется состоянием и поведением. Состояние системы задается значениями атрибутов состояния объектов, входящими в данный момент в систему, а также составом коллекций. Поведение системы задается методами значений, входящих в систему. Методы могут изменять состояние системы. Спецификация модуля канонической модели, как и набор абстрактных машин AMN, задает множество допустимых состояний системы, а также способы изменения состояния системы (методы значений). Θ корректно, если для любой M – спецификации модуля канонической модели – существует инъективное отображение θ задаваемого ею множества состояний системы, в множество состояний системы, задаваемого $M_B = \Theta(M)$ – набором абстрактных машин AMN, и все методы сохраняют свое действие при отображении Θ: если метод m переводит систему, задаваемую M, из состояния s_1 в s_2, то метод $\Theta(m)$ переводит систему, задаваемую M_B, из состояния $\theta(s_1)$ в $\theta(s_2)$. Это означает, что системы, задаваемые M и M_B, будут вести себя одинаково.

Механизмом построения множества состояний системы по спецификации модуля канонической модели является формальная семантика ядра канонической модели, определенная в первой главе. Механизм выделения множества состояний системы, задаваемой множеством абстрактных машин, описан в разделе 4.2.

Доказательство корректности Θ состоит из следующих этапов:

- построение инъективного отображения θ из пространства $\mathbb{S}$ состояний системы, задаваемой спецификацией модуля канонической модели M, в пространство $\mathbb{S}_B$ состояний системы, задаваемой набором абстрактных машин $M_B = \Theta(M)$;
- проверка корректности отображения ограничений на пространство состояний;
- проверка корректности отображения спецификаций методов.

Заметим, что отображение θ можно естественным образом распространить на формулы. Обозначим $\theta_f(\cdot)$ отображение, определенное на формулах над пространством $\mathbb{S}$. Терм $\theta_f(\phi)$ получается из формулы ϕ применением

отображения θ ко всем термам, входящим в ϕ. Таким образом, формула фактически переносится из пространства состояний $\mathbb{S}$ в пространство $\mathbb{S}_B$.

Корректность отображения ограничений на пространство состояний заключается в следующем. Пусть $\mathbb{R} = \{\phi_1, \ldots, \phi_n\}$ теория, задающая выделение множества состояний спецификации модуля канонической модели M. Пусть $\mathbb{R}_B = \{\psi_1, \ldots, \psi_m\}$ теория, задающая выделение множества состояний набора абстрактных машин M_B. Отображение ограничения корректно, если конъюнкции $\psi_1 \wedge \ldots \wedge \psi_m$ и $\theta_f(\phi_1) \wedge \ldots \wedge \theta_f(\phi_n)$ эквивалентны.

Корректность отображения спецификаций методов поясняется следующей коммутативной диаграммой:

$$\begin{array}{ccccc} o \in M_1 & \xrightarrow{\Theta} & M_2 \ni \Theta(o) \\ \downarrow & & \downarrow \\ so & \xrightarrow{\theta_f} & \theta_f(so) & \Leftrightarrow & s\Theta(o) \end{array}$$

Предикативная спецификация метода m модуля M канонической модели отображается в обобщенную подстановку операции $\Theta(m)$ некоторой абстрактной машины из набора M_B отображением Θ, семантическая функция канонической модели отображает m в предикат $sm \in \mathbb{F}$ над пространством состояний $\mathbb{S}$. По предикату sm строится предикат $\theta_f(sm)$ над пространством состояний $\mathbb{S}_B$. Семантическая функция AMN отображает $\Theta(m)$ в $s\Theta(m) \in \mathbb{F}_B$. Корректность отображения m заключается в эквивалентности формул, задающих предикаты $\theta_f(sm)$ и $s\Theta(m)$.

Построение отображения из пространства состояний системы, задаваемой спецификацией модуля канонической модели, в пространство состояний системы, задаваемой набором абстрактных машин, а также доказательство его инъективности, проведены в разделе 4.3. Доказательство корректности отображения ограничений на пространство состояний рассмотрено в разделе 4.4. Доказательство корректности отображения спецификаций методов, в частности, отображения формул и правил канонической модели, рассмотрено в разделе 4.5.

В пятой главе рассмотрена методика и примеры использования представленных в диссертационной работе методов моделирования спецификаций для автоматизации доказательства корректности решения двух задач над множественными неоднородными источниками:

- задачи синтеза канонических моделей для посредников над неоднородными источниками информации и
- задачи композиции ИС из существующих программных и информационных компонентов в интероперабельных средах.

В разделе 5.1 представлено инструментальное средство автоматического отображения спецификаций канонической модели в AMN, реализующее алгоритмы, определенные в третьей главе диссертационной работы. Описан графический интерфейс инструментального средства и сценарий работы эксперта-конструктора ИС с инструментальным средством. Сценарий применим для доказательства уточнения при решении задач над неоднородными источниками информации. Инструментальное средство реализовано на языке Java 2 в среде Windows [9] и встроено в прототипы средств композиционного проектирования и интеграции информационных источников, разработанные в лаборатории композиционных методов проектирования ИС ИПИ РАН.

В разделе 5.2 рассмотрена методика использования инструментального средства для автоматизации доказательства корректности решения задачи синтеза канонических моделей для посредников над неоднородными источниками информации. Синтез канонической модели на основе набора моделей источников (исходных моделей) состоит в последовательном расширении канонической модели на основе каждой из исходных моделей. Расширение канонической модели на основе исходной модели M_i заключается в построении такого расширения E_i ядра канонической модели, что E_i *уточняется* моделью M_i. Таким образом, для доказательства корректности расширения канонической модели на основе исходной модели необходимо произвести следующие действия:

- построить отображение M_i в E_i;
- отобразить E_i в AMN;
- построить AMN-семантику для M_i;
- применить B-технологию для доказательства того, что M_i уточняет E_i.

В качестве примера рассмотрено расширение канонической модели на основе типа связи (relationship) языка ODL стандарта ODMG.

Тип связи при этом представляет собой единственный элемент исходной модели. Для типа связи построено его отображение в каноническую модель, AMN-семантика. Образ типа связи в канонической модели при помощи инструментального средства автоматически отображен в AMN. Полученные AMN-спецификации (их полный текст приведен в приложении) были введены в инструментальное средство автоматизации доказательства уточнения AMN (B-Toolkit 5.1.4).

Далее при помощи B-Toolkit автоматически были сформулированы теоремы, в совокупности утверждающие факт уточнения типом связи своего образа в канонической модели. Часть теорем была доказана с использованием

автоматических средств доказательства, остальные теоремы были доказаны интерактивно.

В разделе 5.2 рассмотрена методика использования инструментального средства для автоматизации доказательства корректности решения задачи композиции ИС из существующих программных и информационных компонентов в интероперабельных средах. Процесс композиционного проектирования ИС состоит из пяти основных этапов:

- поиск компонентов, онтологически релевантных спецификации требований;
- разрешение конфликтов между спецификациями требований и компонентов;
- выявление фрагментов спецификации компонентов, которые могли бы служить уточнением соответствующих фрагментов спецификации требований;
- построения композиции таких фрагментов в спецификацию, потенциально уточняющую спецификацию требований;
- доказательство факта уточнения спецификации требований композицией спецификаций компонентов.

В качестве примера приведены результаты первых четырех этапов композиционного проектирования части системы *Исследовательский фонд*, и подробно описан последний этап проектирования. Рассмотрены спецификации системы *Исследовательский фонд* в канонической модели (спецификации требований), а также спецификации компонентов, онтологически релевантные спецификациям требований. Данные спецификации при помощи инструментального средства были отображены в AMN (полный текст полученных AMN-спецификаций приведен в приложении) и введены в инструментальное средство автоматизации доказательства уточнения AMN. Далее автоматически были сформулированы и затем автоматически или интерактивно доказаны теоремы уточнения.

Заключение

Основные результаты работы сводятся к следующему:

- разработан метод формального определения канонических информационных моделей и на его основе – комбинированная денотационно-аксиоматическая семантика ядра канонической объектной информационной модели (языка СИНТЕЗ), позволяющая проводить доказательные рассуждения о моделях информационных ресурсов;

- разработан метод отображения канонических информационных моделей в теоретико-модельный язык спецификаций, основанный на логике предикатов первого порядка и теории множеств – Нотацию Абстрактных Машин (AMN) и на его основе – алгоритмы отображения ядра канонической информационной модели в AMN, позволяющие использовать инструментальные средства доказательства уточнения для автоматизированного доказательства уточнения спецификаций канонической модели;
- разработан метод доказательства корректности отображения информационных моделей с использованием их денотационно-аксиоматической семантики; с использованием метода построено доказательство корректности отображения ядра канонической модели в AMN;
- на основе вышеуказанных алгоритмов разработаны инструментальные средства автоматического отображения спецификаций ядра канонической модели в AMN, нашедшие применение в качестве составной части инструментария эксперта-конструктора, реализующего композиционное проектирование ИС, и прототипа средств поддержки предметных посредников;
- разработана методика использования представленных в диссертационной работе методов моделирования спецификаций для автоматизации доказательства корректности решения задач над неоднородными источниками информации; практическое действие методики показано на примерах задачи синтеза канонических моделей для посредников, а также задачи композиции ИС из существующих программных и информационных компонентов в интероперабельных средах.

Публикации автора по теме диссертации

1. Ступников С.А. Отображение канонической модели спецификаций в формальную нотацию для моделирования уточняющих спецификаций // Труды XXIV Конференции молодых ученых. – М.: МГУ им. М.В. Ломоносова, 2002. – С. 169-171.

2. Briukhov D.O., Kalinichenko L.A., Stupnikov S.A. Compositional approach for heterogeneous sources registration at a subject mediator // Emerging Database Research in Eastern Europe: Proceedings of the Pre-Conference Workshop of VLDB 2003. – Cottbus: Brandenburg University of Technology, 2003. – P. 5-12. *С.А. Ступникову в статье принадлежит формальное обоснование регистрации неоднородных источников в предметном посреднике.*

3. Kalinichenko L.A., Martynov D.O., Stupnikov S.A. Query rewriting using views in a typed mediator environment // Advances in Databases and Information Systems: Proc. of the East European Conference. – Springer-Verlag, 2004. – P. 37-53. *Ступникову С. А. в статье принадлежит формальная семантика правил языка СИНТЕЗ.*

4. Ступников С.А., Калиниченко Л.А. Формальная семантика канонической информационной модели в композиционной инфраструктуре распределенных информационных систем. Проблемы и методы информатики. II Научная сессия ИПИ РАН: Тезисы докладов / Под ред. И.А. Соколова. – М.: ИПИ РАН, 2005. – С. 151-153. *С.А. Ступниковым разработана формальная семантика канонической информационной модели.*

5. Ступников С.А. Формальная семантика ядра канонической объектной информационной модели // Системы и средства информатики: Спец. вып. Формальные методы и модели в композиционных инфраструктурах распределенных информационных систем / Под ред. И. А. Соколова. — М.: ИПИ РАН, 2005. – С. 40-68.

6. Ступников С.А. Отображение спецификаций, выраженных средствами ядра канонической модели, в язык AMN // Системы и средства информатики: Спец. вып. Формальные методы и модели в композиционных инфраструктурах распределенных информационных систем / Под ред. И. А. Соколова. — М.: ИПИ РАН, 2005. – С. 69-95.

7. Ступников С.А. Автоматизация верификации уточнения при композиционном проектировании информационных систем и посредников // Системы и средства информатики: Спец. вып. Формальные методы и модели в композиционных инфраструктурах распределенных информационных систем / Под ред. И. А. Соколова. — М.: ИПИ РАН, 2005. – С. 96-119.

8. Ступников С.А., Брюхов Д.О. Представление UML и OCL в канонической информационной модели // Системы и средства информатики: Спец. вып. Формальные методы и модели в композиционных инфраструктурах распределенных информационных систем / Под ред. И. А. Соколова. — М.: ИПИ РАН, 2005. – С. 120-129. *С.А. Ступниковым разработано представление языка OCL в канонической модели.*

9. Программа автоматического отображения спецификаций канонической информационной модели в Нотацию Абстрактных Машин: Свидетельство об официальной регистрации программы для ЭВМ N 2005611080 / С.А. Ступников. Зарегистрировано в Реестре программ для ЭВМ 05.05.2005.