

На правах рукописи

Силич Мария Петровна

**ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ РАЗРАБОТКИ
ЦЕЛЕВЫХ ПРОГРАММ НА ОСНОВЕ
ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ МЕТОДОЛОГИИ
МОДЕЛИРОВАНИЯ**

Специальность

05.13.01 – Системный анализ, управление и обработка информации
(отрасль: информация и информационные системы)

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
доктора технических наук

Томск 2005 г.

Работа выполнена в Томском государственном университете систем управления и радиоэлектроники (ТУСУР)

Научный консультант: доктор технических наук, профессор
Ехлаков Юрий Поликарпович

Официальные оппоненты: доктор технических наук, профессор
Губарев Василий Васильевич

доктор технических наук, профессор
Марков Николай Григорьевич

доктор физ.-мат. наук, профессор
Кошкин Геннадий Михайлович

Ведущая организация – Алтайский государственный университет, г. Барнаул.

Защита состоится « 14 » декабря 2005 г. в 15 часов на заседании диссертационного совета Д 212.269.06 при Томском политехническом университете по адресу: 634034, г. Томск, ул. Советская, 84.

С диссертацией можно ознакомиться в библиотеке Томского политехнического университета

Автореферат разослан « ____ » _____ 2005 г.

Ученый секретарь
диссертационного совета
к.т.н., доцент

_____ М.А. Сонькин

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы. Целевые программы (ЦП) являются эффективным способом решения проблем, возникающих в различных сферах человеческой деятельности, т.к. позволяют направить усилия на реализацию заранее выбранного курса действий для достижения поставленных целей. Этот инструмент активно используется как территориальными органами управления, так и отдельными компаниями для решения социальных, экологических, экономических проблем, примерами которых являются: низкая энергетическая эффективность экономики региона, наличие угроз энергетической безопасности, недостаточная отдача от инновационной, инвестиционной деятельности и т.д.

Заметными тенденциями последнего времени являются: усложнение проблем, перерастание их в проблематику; повышение требований к качеству программ (их обоснованности, комплексности, эффективности); стремление к сокращению трудозатрат на разработку и реализацию программ. Все это обуславливает необходимость использования технологии, регламентирующей и стандартизирующей основные этапы разработки целевой программы. В соответствии с современными требованиями технология предполагает не только наличие регламентированной процедуры разработки и совокупности методов и моделей, рекомендуемых к использованию на тех или иных этапах, но и наличие автоматизированных средств, сопровождающих процесс синтеза программы. Только благодаря инструментальным поддерживающим средствам методология может приобрести прагматическую силу и широкое распространение. Таким образом, речь должна идти о создании информационной технологии разработки целевых программ, содержащей в качестве основных компонент регламент процесса разработки, методологию моделирования и инструментальные средства поддержки технологии.

Основным методологическим инструментом разработки целевых программ является аппарат системного анализа и теории принятия решений, в развитие которого большой вклад внесли такие ученые, как Месарович М., Мако Д., Такахара И., Холл А., Янг С., Оптнер С., Акофф Р., Саймон Г., Саати Т., Черняк Ю.И., Федоренко Н.П., Перегудов Ф.И., Тарасенко Ф.П., Михалевич В.С., Волкович В.Л., Клир Дж., Волкова В.Н., Ларичев О.И. и др. Методы и модели, разрабатываемые в рамках этого направления, могут быть использованы как на этапе анализа проблемосодержащей системы для выявления проблемных ситуаций, существующих тенденций и ограничений, так и на этапе синтеза проблеморазрешающей системы для постановки целей, разработки системы мер и программных мероприятий, реализующих поставленные цели, а также проектирования обеспечивающих комплексов (нормативно-правового, организационного, информационного).

Однако на пути применения системного подхода в информационной технологии разработки ЦП существуют определенные трудности. Одна из них связана со слабой формализованностью, «рыхлостью» методов системного анализа, которые, зачастую представляют собой набор эвристических приемов и рекомендаций, требующих творческого осмысления и применения, что затрудня-

ет их автоматизацию. Во многом эффективность использования методов зависит от опыта и квалификации разработчика. При этом ему приходится оперировать нечеткой и неполной информацией. Еще одна трудность состоит в необходимости сочетания множества различных методов, в том числе из других отраслей науки. Поэтому желательно использовать достаточно универсальный язык описания предметной области с тем, чтобы создаваемая декларативная модель могла служить интегрирующей компонентой для различных методов. Преодоление указанных трудностей видится на пути использования методов семантического моделирования, в частности объектно-ориентированных методов, развивающихся в рамках инженерии знаний и технологии экспертных систем, а также индустрии CASE-технологий и технологии реинжиниринга бизнес-процессов.

Вопросам инженерии знаний и технологии экспертных систем посвящено огромное количество трудов. Среди авторов можно выделить Минского М., Хейеса-Рота Ф., Уотермана Д., Лената Д., Уэно Х., Исидзука М., Осуга С., Форсайта Р., Заде Л., Сугено М., Мамдани Е., Поспелова Г.С., Поспелова Д.А., Попова Э.В., Леоненкова А.В., Гаврилову Т.А.. В рамках данного направления разработано множество разнообразных методов представления слабо формализуемых знаний, позволяющих отражать как фактуальные знания о структуре, свойствах объектов предметной области, так и операционные знания о логических, причинно-следственных, ассоциативных зависимостях, в том числе учитывающие неопределенности случайного или нечеткого характера. Использование экспертных знаний, таких как типовые описания различных классов сложных систем, описание зависимостей характеристик системы, стандартные основания декомпозиции, и др., могли бы значительно облегчить разработку целевых программ. В последнее время наблюдается тенденция к интеграции различных форм представления знаний, сочетанию их с традиционными процедурами, методами обработки баз данных. Большие возможности для комплексирования фактуального знания с процедурным предоставляет объектно-ориентированный подход (ООП), активно использующийся в информационных технологиях, в частности в CASE-технологиях. Важными преимуществами данного подхода являются также возможность накапливать теоретические и опытные знания в виде библиотек классов на основе механизма наследования; простота внесения изменений в проекты за счет использования свойств наследования и полиморфизма.

Современные CASE-технологии, предназначенные для создания программных продуктов с помощью автоматизированных инструментальных средств (CASE-средств), поддерживают не только функции проектирования компонент информационной системы, но и функции моделирования систем, являющихся объектом автоматизации, как составную часть процесса проектирования. Большинство из этих технологий используют методологию системного подхода как концептуальную основу построения модели предметной области. Это характерно и для более ранних технологий структурного анализа, так и для более поздних объектно-ориентированных технологий анализа/ проектирования (Object-Oriented Analysis/Design – OOA/D). Последние переживают пе-

риод бурного развития, что вызвано в немалой степени созданием объектного унифицированного языка моделирования UML (Unified Modeling Language) и объектно-ориентированных CASE-средств. Методы объектно-ориентированного анализа и проектирования рассматриваются в работах таких авторов, как Буч Г., Румбах Дж., Джекобсон А., Шлеер С., Меллор С., Бадд Т., Йордон Э., Вендров А.М., Новоженев Ю.В. и др.

Несмотря на успехи в развитии объектно-ориентированных методов моделирования и CASE-средств, их использование для разработки целевых программ не всегда приемлемо. Во-первых, они в основном предназначены для проектирования информационных систем, в связи с чем они оперируют собственным понятийным аппаратом, ориентированным на данную область. Для обозначения многих понятий теории систем вводятся новые термины, термины же системного анализа используются зачастую некорректно, в новых ракурсах и отношениях. Разработанные языки моделирования вполне приемлемы при построении моделей информационных систем, но не всегда удобны для описания других предметных областей. Во-вторых, ООА/D-методы и средства используются лишь для декларативного описания структуры и свойств автоматизируемой системы. Подобное описание отражает либо существующее состояние системы (модель “As is”), либо желаемое (модель “To be”). Нерешенной проблемой при этом остается переход от моделей “As is” к моделям “To be”. Другими словами, существующие CASE-средства, как правило, не содержат процедур генерации и выбора вариантов реализации системы и ее отдельных компонент, а также процедур координации решений, принимаемых на разных уровнях представления системы. Большое количество разнообразных прикладных методов системного анализа и теории принятия решений, ориентированных на поиск средств достижения целей системы, оказались невостребованными в рамках существующих CASE-технологий.

Объектно-ориентированные методологии моделирования и инструментальные средства поддержки, разработанные в рамках CASE-технологий, активно используются в технологии реинжиниринга бизнес-процессов, предназначенной для перепроектирования существующих бизнес-систем на основе новых концепций менеджмента и использования новых информационных технологий. Этому сравнительно новому направлению на стыке теории менеджмента и информатики посвящены труды Хаммера М., Чампи Дж., Робсона М., Уллаха Ф., Шеера А-В., Попова Э.В., Ойхмана Е.Г., Калянова Г.Н., Зиндера Е.З. и др. Объектно-ориентированное моделирование, признанное базовой методологией реинжиниринга, используется для наглядного отражения в виде графических диаграмм и схем существующих и новых, перепроектированных бизнес-процессов. Слабым местом, как и в технологии проектирования информационных систем, является поиск и выбор оптимальных стратегий проектирования в виду отсутствия формализованных методик выполнения этих этапов.

Таким образом, несмотря на имеющиеся предпосылки, проблема создания информационной технологии разработки целевых программ, призванной облегчить процесс разработки и повысить качество его результата, еще не решена, что определяет актуальность темы диссертационной работы.

Цель работы: создание информационной технологии разработки целевых программ с применением объектно-ориентированных методов моделирования и инструментальных средств поддержки, обеспечивающей повышение эффективности процесса разработки.

Для достижения поставленной цели были поставлены и решены следующие **задачи**:

- выявление требований к информационной технологии разработки целевых программ;
- разработка принципов, лежащих в основе технологии;
- разработка объектно-ориентированной методологии моделирования предметной области;
- разработка алгоритмов формирования объектно-ориентированной модели и поиска решений на модели для нахождения оптимальных скоординированных решений, обеспечивающих достижение целей программы;
- разработка типового регламента процесса разработки целевой программы на основе объектно-ориентированной методологии моделирования;
- создание комплекса инструментальных средств поддержки информационной технологии;
- использование информационной технологии для разработки региональных целевых программ в сфере энергосбережения, программ создания и совершенствования организационных комплексов в инновационной сфере, а также для анализа и планирования экономической деятельности хозяйствующих субъектов.

Методы исследования. Для решения поставленных задач в работе используются методы системного анализа, теории иерархических многоуровневых систем, теории принятия решений, инженерии знаний и технологии экспертных систем, объектно-ориентированного моделирования и проектирования.

Научная новизна. В диссертационной работе впервые предложен оригинальный научно обоснованный подход к созданию информационной технологии разработки целевых программ на основе объектно-ориентированной методологии моделирования. Решена крупная научная проблема, имеющая важное значение для развития теории анализа и синтеза проблеморазрешающих систем.

Получены следующие основные результаты, обладающие научной новизной:

1. Впервые сформулированы принципы создания информационной технологии разработки целевых программ, удовлетворяющие требованиям, предъявляемым к технологии. Предложена схема автоматизированной разработки целевых программ, основанная на выдвинутых принципах.

2. Разработана новая методология объектно-ориентированного моделирования систем OMSD (Object Model for System Design), главной особенностью которой является представление модели проблемосодержащей / проблеморазрешающей системы в виде совокупности взаимосвязанных моделей пяти видов:

- модели классов, отражающей структуру множества классов, используемых для описания компонент системы;

- модели объектов, отражающей описание множества вариантов реализации компоненты системы;
- модели зависимостей атрибутов, отражающей отношений функциональной зависимости между атрибутами компоненты системы;
- модели компонент системы, отражающей структуру системы в виде дерева подсистем и элементов;
- модели координации, отражающей отношения зависимости между атрибутами дочерних подсистем и материнской системы.

3. Разработаны новые алгоритмы синтеза модели объектов и поиска решений на ней: генерации объектов и выбора оптимальных решений в условиях неопределенной информации о возможных состояниях среды, позволяющие осуществлять формирование и выбор вариантов реализации компоненты системы. Показана их эффективность по сравнению с существующими алгоритмами.

4. Разработаны новые алгоритмы формирования модели зависимостей атрибутов и ее использования для поиска решений: формирования оценочных зависимостей, прямого и обратного вывода на сети функциональных зависимостей, поиска оптимальных решений, обладающие большей вычислительной эффективностью. Показана зависимость трудоемкости алгоритмов оптимизации от параметров модели.

5. Разработан новый алгоритм координации для согласования вариантов подсистем друг с другом и с выбранным вариантом материнской подсистемы, основанный на методе распространения ограничений и методе перераспределения допустимого уровня затрат. Алгоритм позволяет сократить перебор координирующих воздействий.

6. Предложена новая технология процесса разработки целевой программы на основе объектно-ориентированной методологии моделирования, предполагающая последовательное формирование и использование для принятия решений моделей процесса разработки, проблемосодержащей и проблеморазрешающей систем.

Теоретическая значимость работы заключается в развитии теории анализа и синтеза проблеморазрешающих систем и методов семантического моделирования.

Предложенная объектно-ориентированная методология моделирования объединяет подходы, разработанные в рамках таких научных направлений, как системный анализ, инженерия знаний, CASE-технологии проектирования информационных систем, реинжиниринг бизнес-процессов и вносит существенный вклад в развитие теории семантического моделирования систем. Она предоставляет возможность комплексировать различные методы из широкого спектра методов, используемых для анализа и синтеза систем, используя в качестве интегрирующего начала декларативную модель системы в виде совокупности взаимосвязанных моделей, описанных на едином объектно-ориентированном языке представления знаний.

Разработанный типовой регламент процесса разработки целевой программы, основанный на предложенной схеме автоматизированной разработки ЦП и выдвинутых принципах декларативности, иерархичности, итеративности, типизации и комплексированности, обобщает и развивает существующие схемы системного анализа в виде поэтапных процедур принятия решений.

Предложенные автором алгоритм генерации объектов и алгоритм выбора оптимальных решений в неопределенной ситуации развивают переборные методы генерации и оценки вариантов решений, разрабатываемые в рамках теории выбора и принятия решений. Предложенные методы прямого и обратного вывода, а также методы поиска оптимальных решений на модели функциональных зависимостей, представляют интерес с точки зрения развития методов управления выводом на системах правил-продукций, разрабатываемых в теории инженерии знаний. Предложенный алгоритм формирования оценочных зависимостей интересен тем, что позволяет объединить идеи метода анализа иерархий с методами вывода на системах продукций в рамках модели функциональных зависимостей. В разработанном автором алгоритме координации вариантов подсистем нашли воплощение идеи некоторых методов теории координации, в частности метода распространения ограничений и метода распределения ресурсов.

Практическая значимость работы. Разработанная информационная технология разработки ЦП с применением объектно-ориентированных методов моделирования и созданных инструментальных средств поддержки может быть использована для создания широкого круга социально-экономических проблеморазрешающих систем, в частности для разработки целевых программ развития регионов по различным аспектам, разработки программ развития организационно-технологических комплексов, создания инвестиционных программ. Использование технологии позволяет существенно сократить трудоемкость процесса разработки и повысить качество конечного результата за счет использования типовых фрагментов знаний, структурирования информации о системе на базе единой модели, возможности в автоматизированном режиме с использованием различных методов генерировать варианты реализации компонент системы, оценивать, согласовывать и выбирать оптимальные варианты.

Кроме того, данная технология может быть использована студентами вузов в учебном процессе для выполнения курсовых и дипломных проектов, а также в ходе изучения таких дисциплин, как теория систем и системный анализ, моделирование систем, реинжиниринг бизнес-процессов и т.д.

Связь работы с научными программами, планами, темами. Основные результаты, полученные в диссертации, нашли отражение в госбюджетных научно-исследовательских работах, проводимых Томским государственным университетом систем управления и радиоэлектроники: «Разработка научных основ проектирования интеллектуальных компьютерных технологий управления техническими объектами, технологическими процессами и процессами обучения» (номер НИР 2.1.97, номер гос. регистрации 01980002350); «Разработка принципов построения и основ теории интеллектуальных систем обучения и управления техническими объектами и технологическими процессами» (номер НИР 1.2.02); «Развитие теории и методов объектно-ориентированного моделирования и пространственно-временного анализа бизнес-процессов в территориально-распределенных системах» (номер НИР 1.2.04, номер гос. регистрации 0120.0 411196).

Реализация результатов и их внедрение. Разработанные технология, методология моделирования, алгоритмы и инструментальные средства использованы для создания: программы углубления энергоэффективности на территории Томской области на 2004-2008 г.г.; региональной программы реформирования системы ресурсобеспечения населения; технологии формирования программы энергетической безопасности региона; программы совершенствования деятельности Томского технопарка; проекта регионального учебно-научно-инновационного комплекса; информационной системы оценки финансово-экономического состояния предприятия; информационной системы формирования программы инвестирования в разработку нефтегазовых месторождений.

Основные положения, выносимые на защиту:

1. Предложенные принципы создания информационной технологии разработки целевых программ и схема автоматизированной разработки ЦП, обеспечивают выполнение требований, предъявляемых к основным компонентам технологии.

2. Разработанная объектно-ориентированная методология моделирования позволяет отражать различные аспекты проблемосодержащей / проблеморазрешающей системы: структурированное множество типовых классов описания компонент системы в виде модели классов; иерархию подсистем и элементов системы в виде модели компонент; множество вариантов реализации отдельных компонент в виде модели объектов; функциональные зависимости между атрибутами компоненты системы в виде модели зависимостей атрибутов; зависимости между атрибутами различных компонент в виде модели координации.

3. Предложенные алгоритмы формирования объектно-ориентированной модели системы и поиска решений на модели, позволяющие генерировать различные варианты реализации компонент системы, пополнять описания вариантов, оценивать их и осуществлять выбор оптимальных вариантов, а также согласовывать варианты различных компонент системы, обеспечивают не только адаптацию методов теории принятия решений и дедуктивного логического вывода к объектному языку описания модели, но и большую эффективность по сравнению с известными методами.

4. Предложенный регламент разработки целевых программ позволяет разработчику гибко настраивать последовательность этапов и выбирать наиболее подходящие методы из спектра методов, рекомендуемых для различных этапов, исходя из особенностей конкретной ситуации.

5. Разработанные инструментальные средства поддержки облегчают процесс разработки целевых программ за счет автоматизации процедур создания модели и поиска решений на модели, за счет использования экспертных знаний и типовых компонент модели, за счет систематизации информации о проекте.

6. Результаты применения созданной информационной технологии для разработки целевых программ различного назначения и информационных систем показали ее эффективность, в частности сокращение трудозатрат, обеспечение комплексности и обоснованности принимаемых решений.

Публикации. По теме диссертационной работы опубликовано 62 научные работы, в том числе 1 монография, 28 статей (из них 15 статей в изданиях, рекомендованных ВАК для опубликования результатов докторских диссертаций).

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на следующих конференциях, симпозиумах и семинарах: Всероссийской научно-технической конференции «Синтез и проектирование многоуровневых систем управления» (Барнаул 1982); Всесоюзном научно-практическом семинаре «Прикладные аспекты управления сложными системами» (Кемерово, 1983); республиканском семинаре «Опыт и проблемы разработки территориальных АСУ» (Томск, 1983); Всесоюзной конференции «Проблемы и методы принятия решений в организационных системах управления» (Москва, 1988); Всесоюзной конференции «Территориальные неоднородные информационно-вычислительные системы» (Новосибирск, 1988); Всесоюзной научно-технической конференции «Человеко-машинные системы и комплексы принятия решений» (Таганрог, 1989); Всесоюзном совещании по экспертным системам (Суздаль, 1990); международных научно-практических конференциях СИБРЕСУРС (Новосибирск - 1996, Барнаул - 1998); международной конференции «Теория и техника передачи, приема и обработки информации» (Харьков, 1998); международном симпозиуме SIBCONVERS (Томск, 1999); Всероссийских научно-технических конференциях «Энергетика: экология, надежность, безопасность» (Томск, 1999, 2002); Всероссийском совещании «Энергосбережение и энергетическая безопасность регионов России» (Томск, 2000); международной научно-практической конференции «Актуальные проблемы управления-2001» (Москва, 2001); международном симпозиуме «Информационные и системные технологии в индустрии, образовании и науке» (Караганда, 2003); пятой научно-практической конференции «Современные средства и системы автоматизации» (Томск, 2004); международных русско-корейских симпозиумах KORUS (Томск -1998, Новосибирск-1999, Томск-2001, Ульсан-2003, Томск-2004, Новосибирск-2005).

Структура и объем работы. Диссертационная работа включает: введение, пять глав, заключение, список литературы из 229 наименований, 8 приложений. Общий объем диссертации составляет 386 страниц машинописного текста. Работа содержит 78 рисунков, 11 таблиц.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность направления исследований, приводятся цель, задачи работы и методы исследования, формулируется научная новизна, теоретическая и практическая значимость полученных результатов, приводятся результаты внедрения и работы, основные положения, выносимые на защиту, а также сведения о публикациях, апробации работы, объеме и структуре диссертации.

Первая глава посвящена формированию концепции информационной технологии разработки целевых программ. На основе анализа общесистемной технологии, а также существующих информационных технологий разработки систем различной природы (информационных, технических, бизнес-систем) выявлены основные тенденции в развитии данных технологий:

- использование типовой последовательности этапов, соответствующей общесистемной последовательности принятия решений, с конкретизацией этапов, адаптирующей ее для применения в некоторой предметной области;
- переход к итеративным схемам организации работ, позволяющим корректировать цели разработки и ранее принятые решения;
- интеграция различных методов моделирования, последовательное формирование совокупности связанных моделей, начиная от концептуальных представлений и заканчивая моделями реализации;
- использование иерархического и объектно-ориентированного подходов в моделировании систем;
- применение интегрированных инструментальных средств, обеспечивающих объединение различных методов, используемых на разных этапах и разными участниками проекта, использование типовых моделей из библиотеки стандартных решений.

Основными особенностями разработки целевых программ, определяющими необходимость создания информационной технологии, снижающей затраты на разработку и улучшающей качество программ, являются: комплексность (сложность) исходной проблематики и соответствующей проблеморазрешающей системы; слабая формализуемость обрабатываемой информации и используемых методов; уникальность (нетипичность) программ. Исходя из этих особенностей сформулированы требования, предъявляемые к информационной технологии:

- универсальность, высокая степень обобщенности, позволяющая применять ее для разработки самых разнообразных программ;
- гибкость, простота адаптации, настройки на конкретную предметную область;
- наглядность и обозримость формируемой модели проблемосодержащей / проблеморазрешающей системы;
- использование при построении моделей опыта экспертов;
- открытость и интегрируемость инструментальных средств, возможность их расширения, а также сопряжения с другими приложениями, реализующими различные методики.

Для удовлетворения перечисленных требований предлагаются следующие принципы создания информационной технологии:

- регламент должен предписывать вид декларативной модели, формируемой на каждой стадии, и совокупность методов, используемых для ее построения и принятия решений на модели (принцип декларативности);
- схема применения этапов разработки ЦП должна быть итеративной – по типу спиральной или макетной схемы (принцип итеративности);

- разработка целевой программы должна осуществляться на разных уровнях абстрагирования с использованием иерархического подхода (принцип иерархичности);
- при разработке целевой программы следует использовать типовые знания (принцип типизации);
- должна быть предусмотрена возможность сочетания, интегрирования разнообразных методов и процедур принятия решений и соответствующих программных компонент (принцип комплексности).

Обобщенная схема автоматизированной разработки целевых программ, основанная на предложенных принципах, приведена на рис. 1.

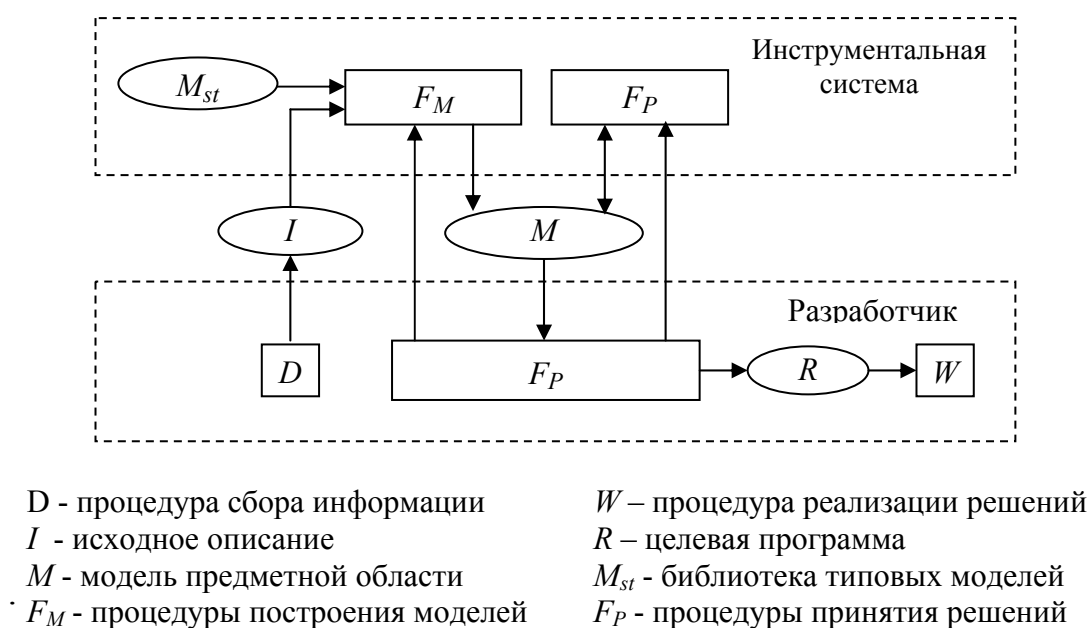


Рис. 1. Схема автоматизированной разработки целевых программ

Инструментальная система, построенная по принципу систем поддержки, вместо единой процедуры, содержит библиотеку процедур, которые наряду с неавтоматизированными процедурами принятия решений могут использоваться на различных этапах разработки целевой программы. Декларативная модель M служит интегрирующим звеном для множества процедур. Подсистема моделирования F_M инструментального средства позволяет разработчику формировать модель, используя при этом библиотеку типовых моделей M_{st} и библиотеку процедур F_P .

Во второй главе предлагается объектно-ориентированная методология моделирования систем OMSD (Object Model for System Design). Модель предметной области, формируемая в процессе разработки целевой программы, представляет собой совокупность моделей процесса разработки, проблемосодержащей и проблеморазрешающей систем. Модель любой из систем, в свою очередь, содержит множество взаимосвязанных моделей пяти видов (рис. 2): классов (ClassModel), объектов (ObjectModel), зависимостей атрибутов (AttributeModel), компонент (ComponentModel), координации (CoordinateModel).

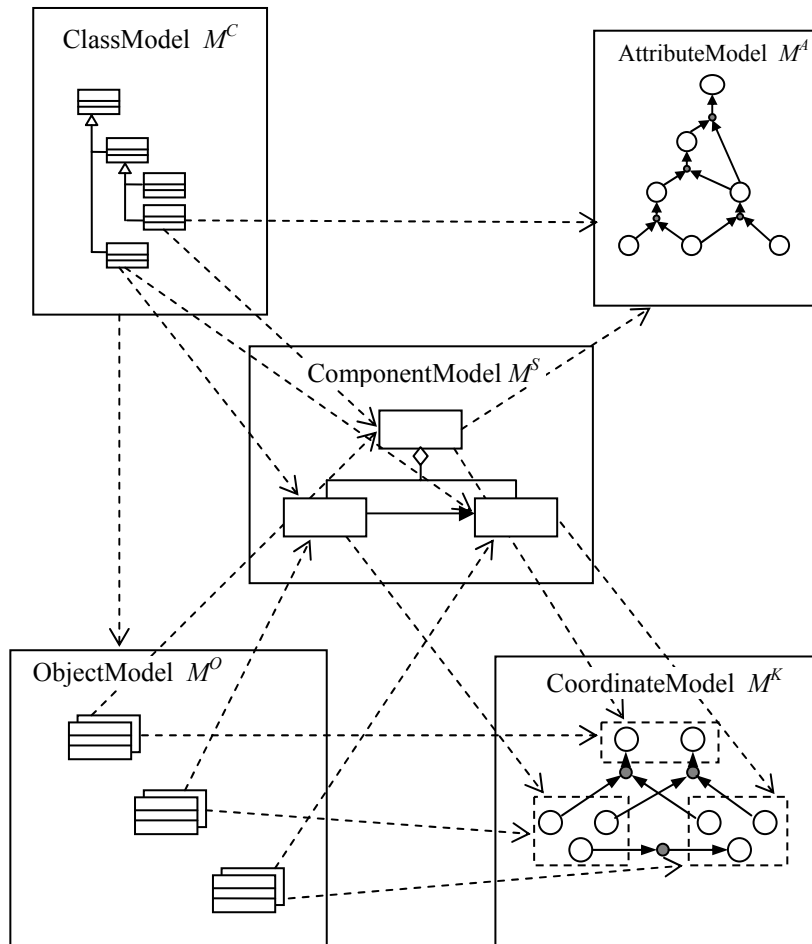


Рис. 2. Составляющие объектно-ориентированной модели системы

Модель компонент отражает структуру системы в виде дерева подсистем и элементов. Каждой из компонент сопоставляется модель объектов, позволяющая рассматривать компоненту как некий целостный объект, описываемый набором характеристик (атрибутов) и выполняемых им действий (методов). Объектное описание вариативно, т.е. может отражать различные состояния или варианты компоненты, однако структура описания инвариантна. Множество структур (классов), используемых для описания компонент системы, отражается в модели классов в виде иерархии наследования. Для описания причинно-следственных, логических отношений между атрибутами компоненты системы используется модель зависимостей атрибутов. Модель координации позволяет отражать отношения зависимости между атрибутами дочерних подсистем и материнской системы.

Каждая из моделей, как и любой объект в рамках объектно-ориентированной парадигмы, рассматривается как совокупность трех частей: имени, описательной (декларативной) части и процедурной. Описательная часть содержит множество некоторых концептов модели, связанных различного рода отношениями. Процедурная часть представлена в виде совокупности методов, позволяющих формировать модель и осуществлять поиск решений.

Модель классов содержит дерево наследования классов и множество методов для работы с классами:

$$M^C = \langle T^C, P^C \rangle.$$

Дерево $T^C = \langle C, R^{in} \rangle$ отражает множество классов C , связанных отношением наследования (inheritance) $R^{in} \in C \times C$. Множество методов P^C включает методы наследования, позволяющие создавать новые классы, наследуя их от имеющихся классов, и методы обобщения, позволяющие автоматически или в диалоговом режиме формировать дерево наследования.

Класс $c_i \in C$ представляет собой кортеж:

$$c_i = \langle n(c_i), \{n(c_j)\}, A(c_i), P(c_i) \rangle,$$

где $n(c_i)$ – имя класса; $n(c_j)$ – имя класса-предка ($c_j R^{in} c_i$), $A(c_i) = \{a_m\}$ – множество атрибутов класса, включающее подмножества новых, наследуемых и перекрывающих атрибутов; $P(c_i) = \{p_n\}$ – множество методов (присоединенных процедур) класса, включающее подмножества новых, наследуемых и перекрывающих методов.

Каждый из атрибутов задается тройкой:

$$a_m = \langle n(a_m), t(a_m), D(a_m) \rangle,$$

где $n(a_m)$ – имя атрибута (может быть составным, включающим имя класса); $t(a_m)$ – тип атрибута ($\langle \text{String} \rangle$, $\langle \text{Real} \rangle$, $\langle \text{Integer} \rangle$, $\langle \text{Date} \rangle$, $\langle \text{Time} \rangle$, $\langle \text{Object} \rangle$, ...); $D(a_m) = \{d_k\}$ – множество значений (доменов) атрибута.

Домен задается в зависимости от типа атрибута. Например, для атрибута типа $\langle \text{Real} \rangle$ или $\langle \text{Integer} \rangle$ – в виде интервала, для атрибута типа $\langle \text{String} \rangle$ – в виде списка строк, для атрибута типа $\langle \text{Object} \rangle$ – в виде имени класса (списка имен классов). Если описание домена опущено, атрибут может принимать любое значение соответствующего типа.

Посредством атрибутов типа $\langle \text{Object} \rangle$ в описание объекта может включаться описание другого объекта. Например, атрибуты подсистемы могут содержать ссылки на объекты, описывающие элементы данной подсистемы или некоторые комплексные свойства, которые сами описываются множеством собственных атрибутов.

Множество методов класса $P(c_i)$ содержит обязательное подмножество методов доступа к значениям атрибутов $P^a(c_i) \subseteq P(c_i)$, которые могут быть двух типов: `get_value` – метод получения значения; `set_value` – метод задания значения.

Классы, в отличие от традиционного подхода, могут сопоставляться не только неделимым компонентам (элементам), но и подсистемам, системе в целом, а также отношениям. Включение в определение атрибута домена позволяет рассматривать класс не просто как шаблон описания, а как обобщенное описание множества объектов одного класса (аналогично фрейму общего понятия). Введение понятия перекрывающих атрибутов позволяет на основе обобщенных описаний создавать более узкие.

Модель объектов создается на базе некоторого класса и представляет собой множество мультиобъектов и множество методов для работы с мультиобъектами:

$$M^O(c_i) = \langle \{O^P(c_i)\}, P^O \rangle.$$

Мультиобъектом называется множество объектов (набор реализаций) одного класса, выделенных в соответствии с некоторым признаком p :

$$O^P(c_i) = \{o_k \mid o_k = o(p), c(o_k) = c_i\} \subset O(c_i).$$

Каждый объект содержит множество значений атрибутов, дополненных факторами уверенности.

$$o_k = \langle n(o_k), n(c), \{d_k(a_m) / cf(d_k(a_m))\} \rangle,$$

где $n(o_k)$ – имя объекта; $n(c) \mid c = c(o_k)$ – имя класса, на базе которого реализован объект; $d_k(a_m) \in D(a_m) \mid a_m \in A(c(o_k))$ – значение атрибута; $cf(d_k(a_m)) \in [0; 1]$ – фактор уверенности в значении атрибута.

Признак p представляет собой специальный атрибут (множество атрибутов), каждому значению которого (комбинации значений) соответствует некоторая конкретная реализация o_k , являющаяся, таким образом, функцией от p . Выделяют три основных типа базовых признаков: время, пространство и группа (аналогично понятию «базы», введенному Клиром в его системологии).

Множество методов модели объектов P^O включает методы формирования, генерации объектов и методы поиска решений (поиска объектов по заданному шаблону, сравнения и оценки объектов, вывода закономерностей и др.).

В отличие от традиционного подхода, мультиобъекты используются не столько для описания однотипных элементов, сколько для отражения множества вариантов реализации компоненты системы, множества ее состояний в различные моменты времени, в различных точках пространства и т.д. На рис. 3 представлен пример мультиобъекта в графической и в табличной форме. Каждый объект мультиобъекта, созданного на базе класса «Потребление энергоресурсов» в соответствии с комбинированным признаком «год - регион», содержит конкретные значения атрибутов, достигнутые в заданном году в заданном регионе.

Энергопотребление (2000, N-ская обл.):	Энергопотребление	2000, N-ская обл.	2001, N-ская обл.	2002, N-ская обл.
Энергопотребление (2001, N-ская обл.):	Энергоемкость ВРП, т.у.т./т.руб	182	126	100
Энергопотребление (2002, N-ская обл.):	Годовой прирост энергоемкости, %	-4,3	-2,2	-0,7
Потребление энергоресурсов	Электроемкость ВРП, кВтч/т.руб	150	105	81,8
Год: 2002	Годовой прирост электроемкости, %	-4,8	-2,99	-1,12
Регион: N-ская обл.				
Энергоемкость ВРП, т.у.т./т.руб: 100				
Годовой прирост энергоемкости, %: -0,7				
Электроемкость ВРП, кВтч/т.руб: 81,8				
Годовой прирост электроемкости, %: -1,12				

Рис. 3. Пример мультиобъекта в графической нотации и в табличной форме

Модель зависимостей атрибутов используется для отражения каузальных отношений между атрибутами, описывающими компоненту системы. Модель создается на базе некоторого класса c и включает множество атрибутов данного класса $A(c)$, множество ненаправленных отношений зависимости (dependence) R^d и множество отношений функциональной зависимости (functional dependence) R^{fd} , а также множество методов P^A формирования зависимостей и логического вывода:

$$M^A(c) = \langle A(c), R^d, R^{fd}, P^A \rangle.$$

Направленное отношение функциональной зависимости R^{fd} $((a_1, \dots, a_n) R^{fd} a_0)$, связывающее атрибут-функцию a_0 с атрибутами-аргументами $a_1, \dots, a_n$, может быть получено на базе ненаправленного отношения зависимости R^d путем «разрешения» сопоставляемого ему выражения (например, уравнения), относительно выбранного атрибута.

Множество отношений функциональной зависимости представляют собой сеть зависимостей (рис. 4). В истоках сети находятся базовые атрибуты $\{a_i\}^B$, значения которых не зависят от значений других атрибутов. В стоках сети располагаются целевые атрибуты $\{a_i\}^G$, значения которых определяют целевое состояние компоненты системы. Отношение функциональной зависимости можно представить, как объект стандартного класса 'FuncDependence', содержащий имена атрибутов-аргументов, имя атрибута-функции, закономерность (вид функциональной зависимости), а также методы для прямого и обратного вывода значений атрибутов.

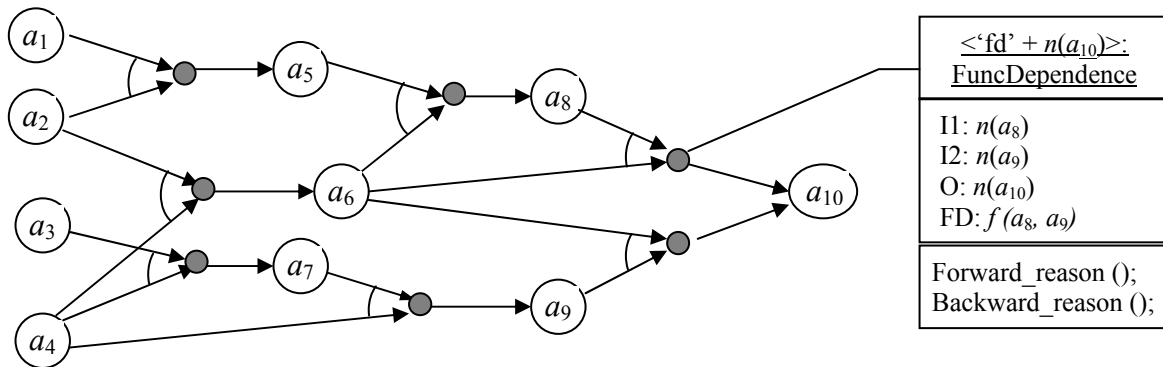


Рис. 4. Сеть функциональных зависимостей атрибутов

Закономерность, показывающая, как именно значение атрибута-функции определяется значениями атрибутов-аргументов, задается в виде совокупности m конъюнктивных правил Rl_k вида:

$$Rl_k : (d(a_1) \in D_k(a_1)) \& \dots \& (d(a_n) \in D_k(a_n)) \rightarrow (d(a_0) = f_k(a_1, \dots, a_n)) / cf(Rl_k),$$

где $d(a_i)$ - текущее значение атрибута a_i ; $D_k(a_i) \subseteq D(a_i)$ - допустимая область атрибута a_i ; $f_k(a_1, \dots, a_n)$ - функция, представленная либо в виде конкретного значения, либо в виде формулы, либо в виде некоторой процедуры-функции; $cf(Rl_k) \in [0; 1]$ - фактор уверенности правила.

Область допустимых значений $D_k(a_i)$ в условной части правила задается либо в виде списка (для лингвистических значений), либо в виде интервалов (для числовых значений). Она может состоять из одного значения и, наоборот, совпадать со всей областью определения атрибута (атрибут может принимать любое значение), а также задаваться в виде дизъюнкции нескольких областей.

Формула в консеквенте правила может включать текущие значения атрибутов-аргументов, константы, знаки арифметических операций и математических функций и т.д. Процедура-функция может иметь в составе входных параметров текущие значения атрибутов-аргументов и должна возвращать значение определяемого атрибута. Метод определения этого значения может быть любым. Примеры стандартных процедур: SQL-запрос на поиск значения атрибута в базе данных, запрос к пользователю, фаззификация, дефаззификация, генерация случайного значения.

Активизация правила осуществляется в случае, если текущие значения атрибутов-аргументов входят в допустимые области значений, указанные в условиях правила. Определение фактора уверенности при агрегации условий осуществляется по правилу минимизации, при активизации заключения – по правилу минимизации или алгебраического произведения.

От типового класса 'FuncDependence' могут быть наследованы классы, описывающие специальные виды зависимостей. Один из специальных видов зависимостей – отношение оценочной зависимости (estimated dependence) R^{ed} . Оно используется в случае, когда разработчику не удастся полностью определить вид зависимости: можно говорить лишь об относительных степенях влияния атрибутов-аргументов (в баллах от 0 до 1), определяющих степень изменения значения атрибута-функции. Примерами таких оценочных отношений являются: зависимость уровня достижения цели от той или иной стратегии; зависимость степени реализации стратегии от выполнения тех или иных мероприятий и т.д. Закономерность отношения оценочной зависимости $(a_1, \dots, a_n) R^{ed} a_0$ представлена правилами, содержащими в консеквенте взвешенную сумму значений атрибутов-аргументов:

$$d(a_1) \in D_k(a_1) \& \dots \& d(a_n) \in D_k(a_n) \rightarrow d(a_0) = v_1 \cdot d(a_1) + \dots + v_n \cdot d(a_n) / cf,$$

где $d(a_i) \in [0; 1]$ – текущее значение атрибута a_i ($t(a_i) = \langle \text{Real} \rangle$); $D_k(a_i)$ – допустимая область атрибута a_i ; $v_i \in [0; 1]$ – степень влияния i -го атрибута-аргумента на атрибут-функцию ($\sum_{i=1}^n v_i = 1$); $cf \in [0; 1]$ – фактор уверенности правила.

Перечислим основные достоинства модели зависимостей атрибутов. Модель позволяет объединять различные формы операционного знания. Декларативное описание зависимостей дает возможность легко изменять вид зависимостей, не меняя методы, их обрабатывающие. Структурирование исполняемых утверждений посредством функциональной сети повышает наглядность, «прозрачность» модели и существенно снижает трудоемкость вывода.

Модель компонент отражает структуру системы и включает множество подсистем S и множество элементов E , связанных различного рода отношениями (рис. 5):

$$M^S = \langle S, E, R^{ag}, R^{cn}, R^{as}, P^S \rangle.$$

Среди отношений выделяется, прежде всего, отношение агрегации (aggregation) $R^{ag} \in S \times S$, связывающее материнскую систему с ее дочерними подсистемами. Множество отношений данного типа составляет дерево подсистем. Дерево, как правило, формируется путем последовательной декомпозиции. При этом могут использоваться стандартные основания декомпозиции, задаваемые в виде объектов класса «DecomposeBase».

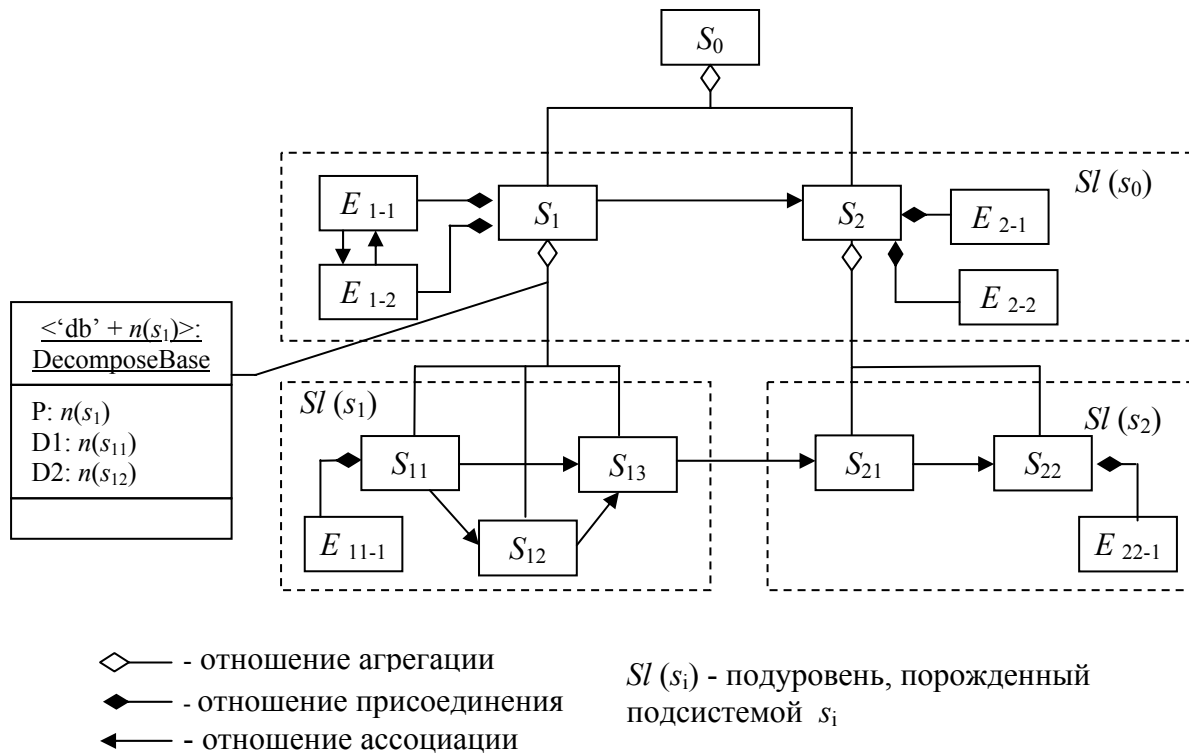


Рис. 5. Отношения между компонентами системы

Еще один вид отношений - отношение присоединения (connection) $R^{cn} \in S \times E$, которое устанавливается между подсистемой и присоединяемым к ней элементом. При этом в класс, сопоставляемый подсистеме, вводится атрибут типа <Object>, указывающий на элемент. Значением этого атрибута в реализации класса является имя мультиобъекта, сопоставленного элементу.

Помимо связей, отражающих структурные отношения (агрегации и присоединения), между подсистемами или между элементами могут устанавливаться отношения ассоциации (association) $R^{as} \in S \times S, R^{as} \in E \times E$. Они отражают коммуникационные, пространственные, временные, разнообразные эмпирические отношения. Для упрощения рассматриваются только отношения ассоциации внутри одного подуровня (подуровень составляют дочерние подсистемы одной материнской системы и присоединенные элементы).

Модель компонент выступает каркасом для всех остальных моделей, т.к. каждой из компонент системы сопоставляется класс описания, и на основе классов могут быть сформированы модель объектов, модель зависимостей атрибутов, координационная модель:

$$s_i = \langle n(s_i), c(s_i), M^O(c(s_i)), M^A(c(s_i)), \{M^K(c(s_i))\} \rangle ,$$

$$e_i = \langle n(e_i), c(e_i), M^O(c(e_i)), M^A(c(e_i)) \rangle .$$

Множество методов P^S для работы с моделью компонент включает методы создания и редактирования иерархии компонент, методы создания диаграмм взаимосвязей, а также методы вывода на диаграммах взаимосвязей.

Модель координации, сопоставляемая некоторой подсистеме s_0 , предназначена для отражения отношений зависимости, установленных на множестве атрибутов $A(s_0)$, включающем множество атрибутов данной подсистемы и множества атрибутов доминируемых компонент, т.е. компонент, связанных с данной подсистемой отношениями агрегации или присоединения.

$$M^K(s_0) = \langle A(s_0), R^d, R^{fd}, P^K \rangle ,$$

где $A(s_0) = A(c(s_0)) \cup (\cup A(c(s_i))) \cup (\cup A(c(e_j))) \mid (s_0 R^{ag} s_i), (s_0 R^{cn} e_j)$ – множество атрибутов координационной модели, R^d, R^{fd} – межуровневые и межподсистемные зависимости, P^K – методы модели координации.

Среди отношений выделяются межуровневые зависимости, представленные, как правило, зависимостями атрибутов материнской системы от атрибутов дочерних подсистем и межподсистемные, отражающие взаимное влияние атрибутов дочерних подсистем (рис. 6).

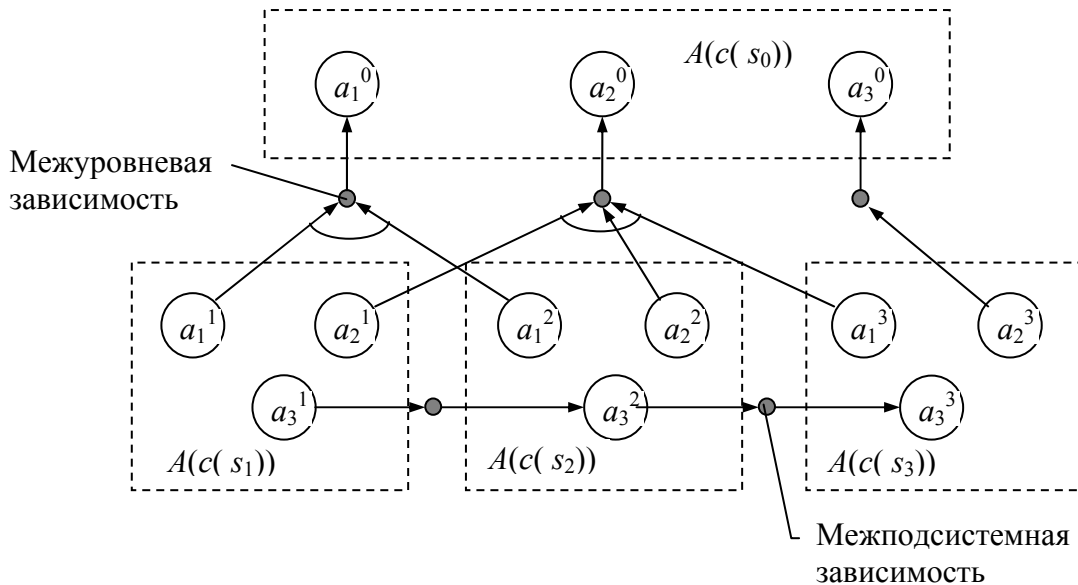


Рис. 6. Координационная модель подуровня

Множество методов P^K включает методы формирования модели, методы интеграции и методы координации.

Модель используется для согласования вариантов подсистем подуровня (или элементов) друг с другом и с вариантом материнской подсистемы.

В третьей главе описывается технология разработки целевых программ на основе объектно-ориентированной методологии моделирования, в том числе алгоритмы, используемые для формирования моделей и поиска решений на них, а также регламент, содержащий рекомендации для каждого этапа разработки относительно вида формируемых моделей и спектра методов, которые могут быть использованы разработчиком.

Для модели объектов разработаны два новых алгоритма. Алгоритм генерации объектов предназначен для формирования вариантов компоненты системы на основе заданного класса. Исходными данными является описание класса c . Необходимо сформировать мультиобъект $O^P(c) = \{o_k\}$, содержащий множество вариантов реализации класса. Предлагается модифицированный блочно-лабиринтный метод морфологического анализа.

Для того, чтобы обеспечить более логичную последовательность формирования вариантов предлагается осуществлять предварительное ранжирование атрибутов. Все множество комбинируемых атрибутов $A^K = \{a_i\}, i = \overline{1, n_K}$ разбивается на кластеры, соответствующие различным аспектам описания компоненты. В каждом кластере выделяется один основной атрибут, определяющий основную характеристику по данному аспекту. Остальные атрибуты кластера являются дополнительными характеристиками. Сначала ранжируются основные атрибуты кластеров по убыванию важности, затем – атрибуты каждого кластера (по убыванию важности кластеров). Таким образом, при генерации вариантов сначала комбинируются атрибуты, задающие «каркас» вариантов, затем «каркас» дополняется деталями (от наиболее важных до наименее значимых).

Кроме того, в предлагаемом алгоритме более четко по сравнению с традиционным методом определена процедура исключения вариантов. На каждом шаге сгенерированные варианты после проверки на совместимость оцениваются по критериям качества и затрат, для которых задаются уровни притязаний. Вводится оценка максимального количества вариантов, оставляемых после отбрасывания на каждом шаге. Оценка вычисляется по формуле: $N_i^{os} = (N^{ob} / n_K) / |D(a_{i+1})|$, где N^{ob} – общее обозримое количество вариантов, $|D(a_{i+1})|$ – количество значений атрибута, комбинируемого с оставшимися вариантами на следующем шаге. Введенные ограничения на количество комбинаций допускают рассмотрение не всех возможных комбинаций. Однако расположение значений атрибутов в порядке предпочтения обеспечивает рассмотрение комбинаций с наилучшими значениями атрибутов.

Второй алгоритм предназначен для выбора решений (сценариев действий) в неопределенной ситуации. Исходными данными является класс c_1 для описания состояния среды и класс c_2 для описания сценария действий. Необходимо выбрать наиболее предпочтительный сценарий действий $o_j \mid c(o_j) = c_2$ с учетом возможных ситуаций $o_i \mid c(o_i) = c_1$.

Создается объектное описание возможных ситуаций (состояний среды) $O^{p1}(c_1) = \{o_i\}$ и вариантов сценариев действий $O^{p2}(c_2) = \{o_j\}$ на основе соответствующих классов. Затем формируется комбинированный мультиобъект $O^p(c) = \{o_k\}$ путем комбинирования объектов-ситуаций и объектов-сценариев. Каждому объекту $o_k \in O^p(c)$, полученному в результате комбинирования объектов o_i и o_j , сопоставляется значение критерия $K_{ij}(o_k)$. Фактор уверенности критерия определяется исходя из факторов уверенности, рассчитываемых для ситуаций и сценариев. Интегральная оценка j -го сценария действий по критерию определяется как взвешенное среднее оценок для каждой из возможных ситуаций:

$$K_j = \frac{\sum_{i=1}^n K_{ij}(o_k) * cf(K_{ij}(o_k))}{\sum_{i=1}^n cf(K_{ij}(o_k))}.$$

Это дискретный вариант центроидного метода преобразования множества нечетких значений к четкому значению. Оптимальным решением является сценарий с максимальной оценкой.

Алгоритм может рассматриваться как аналог метода «математического ожидания» для решения задач типа «игры против природы». Однако вместо вероятностей используются факторы уверенности, что позволяет использовать алгоритм в ситуациях, когда о реализациях недетерминированных исходов нельзя собрать статистику для построения вероятностных распределений.

Для модели функциональных зависимостей атрибутов разработан ряд алгоритмов: алгоритмы прямого и обратного вывода для решения задач интерпретации и поиска допустимого решения, алгоритмы оптимизации, а также алгоритм формирования оценочных зависимостей.

Алгоритм прямого вывода предназначен для решения задачи интерпретации на модели функциональных зависимостей атрибутов. Исходными данными являются значения базовых атрибутов: $D^B = \{d(a_i) / cf(d(a_i))\}, a_i \in \{a_i\}^B$. Требуется найти все возможные значения целевого атрибута $\{d(a_c) / cf(d(a_c))\}$, соответствующие исходным данным.

Предварительно находится, так называемый, «путь вывода» – цепочка атрибутов, являющихся целями вывода на каждом шаге процесса нахождения решения. В «путь вывода» последовательно включаются: целевой атрибут; атрибуты, отстоящие от целевого на одно ребро; атрибуты, отстоящие от целевого на два ребра и т.д. Для удобства полученная последовательность переписывается в обратном порядке (от базовых атрибутов до заданного атрибута): $Q = \langle a_1, a_2, \dots, a_{c-1}, a_c \rangle$.

Далее последовательно перебираются атрибуты из «пути вывода» и для каждого из них выполняется метод прямого вывода соответствующего отношения зависимости, содержащего данный атрибут как атрибут-функцию. Метод

вычисляет значение атрибута-функции и фактор уверенности на основании закономерности и значений атрибутов-аргументов.

Для каждого небазового атрибута a_i в ходе логического вывода может возникать «ветвление» по числу правил с истинной левой частью. Для организации вывода с «ветвлениями» предлагается использовать стратегию «сначала вглубь». При первом «проходе» по «пути вывода» для любого из атрибутов a_i «срабатывает» первое правило с истинным условием. Его текущее значение и фактор уверенности запоминается в модели объектов – в первом экземпляре соответствующего класса: $d(a_i), cf(d(a_i)) \in o_1$. Кроме того, запоминается номер «сработавшего» правила n_i^* . Дойдя до конца «пути вывода» Q , мы получаем первый вариант решения. На следующем этапе осуществляется возврат к последней точке ветвления. Сначала это последний в «пути вывода» атрибут a_c . Для данного атрибута ищется истинное правило с номером $n_c > n_c^*$. Если такое правило находится, то генерируется новый экземпляр класса, содержащего данный атрибут. В него копируются значения и факторы уверенности атрибутов, стоящих в «пути вывода» перед a_c , и дописываются $d(a_c)$, $cf(d(a_c))$, выведенные с помощью последнего правила. Номер этого правила запоминается, как номер последнего «сработавшего» правила n_i^* . Перебрав таким же образом все «ветви» для рассматриваемой точки ветвления, переходим к атрибуту, стоящему в «пути вывода» перед атрибутом, который был точкой ветвления на предыдущем этапе. Для данного атрибута перебираются все возможные «ветви». По каждой из них процесс вывода осуществляется от зафиксированной точки «ветвления» до целевых атрибутов. Затем поднимаемся к предыдущей точке «ветвления» и т.д. пока не достигнем базовых атрибутов.

Процесс вывода может быть изображен в виде дерева И/ИЛИ (рис. 7).

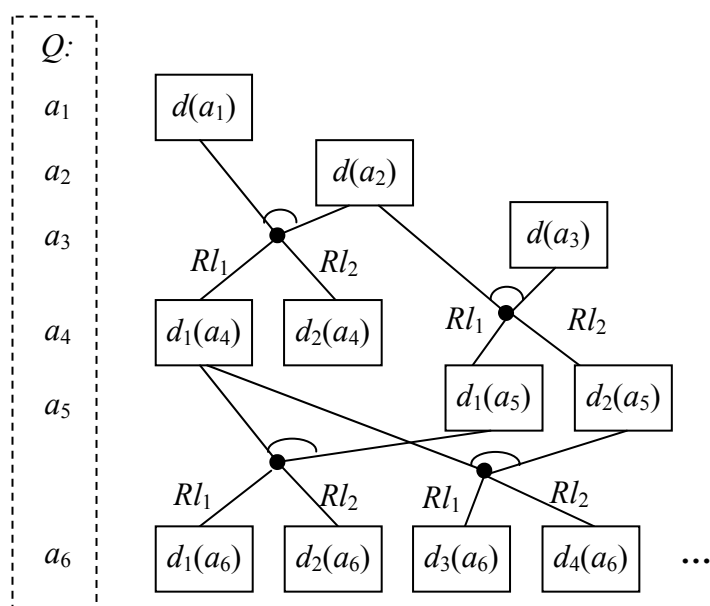


Рис. 7. Пример дерева И/ИЛИ для алгоритма прямого вывода

Для задачи поиска допустимого решения на модели функциональных зависимостей атрибутов предлагается алгоритм обратного вывода. Исходными данными является заданное значение $d(a_c)$ целевого атрибута. Требуется найти все возможные комбинации значений базовых атрибутов: $\{d(a_i)\}, a_i \in \{a_i\}^B$, обеспечивающие достижение заданного значения целевого атрибута,

Алгоритм осуществляет обратный вывод аналогично алгоритму прямого вывода. Для каждого небазового атрибута из «пути вывода» выполняется метод обратного вывода соответствующего отношения зависимости, содержащего данный атрибут как атрибут-функцию. Метод осуществляет поиск правила, содержащего в заключении значение, совпадающее с выводимым значением или входящее в допустимую область для данного атрибута, и записывает области значений атрибутов-аргументов, содержащиеся в условной части, в модель объектов. При этом если для некоторого атрибута a_i ранее уже была определена допустимая область значений $D_1(a_i)$, находится пересечение данной области с областью $D_2(a_i)$, записанной в правиле: $D_3(a_i) = D_1(a_i) \cap D_2(a_i)$. Если $D_3(a_i) = \emptyset$, то правило отбрасывается, в противном случае текущей областью становится область пересечения (рис. 8). В случае, если в заключении правила записано не конкретное значение атрибута-функции, а формула или процедура-функция, то генерируются различные комбинации значений атрибутов-аргументов, осуществляются вычисления и затем уже проверяется, входит ли вычисленное значение в допустимую область.

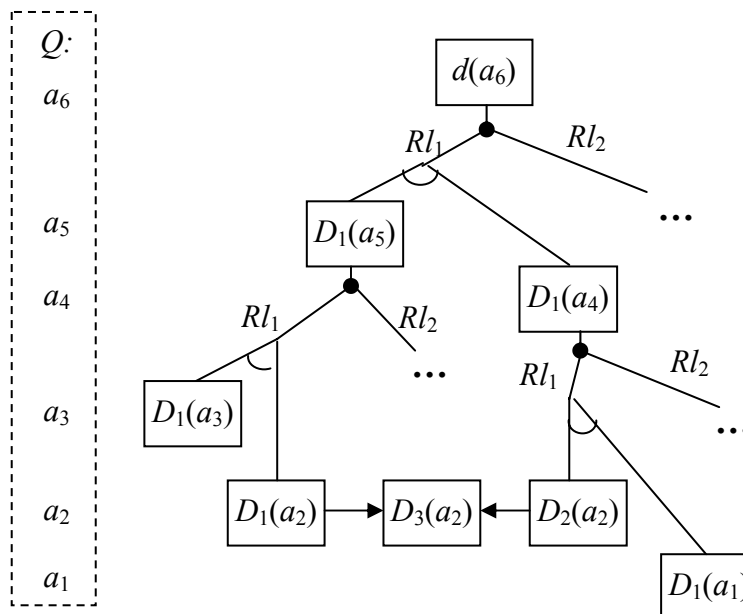


Рис. 8. Пример дерева И/ИЛИ для алгоритма обратного вывода

В процессе поиска решения могут возникать «ветвления». Во-первых, в допустимую область значений атрибута может входить несколько различных значений, стоящих в правых частях разных правил. Во-вторых, к одному и тому же значению некоторого атрибута могут приводить несколько разных правил с различными левыми частями. Для организации вывода с «ветвлениями» пред-

лагается так же, как и в алгоритме прямого вывода, использовать стратегию «сначала вглубь». При первом «проходе» для любого из атрибутов «срабатывает» первое правило с истинным заключением. Номера «сработавших» правил запоминаются. После «прохода» до конца «пути вывода» осуществляется возврат к последней точке «ветвления», потом - к предпоследней и т.д.

Основными достоинствами предложенных алгоритмов вывода являются: возможность обрабатывать правила, содержащие разнообразные формулы и процедуры; более высокая скорость вычислений по сравнению с традиционными алгоритмами прямого и обратного вывода на системе продукций за счет структурирования базы правил посредством сети зависимостей атрибутов. Кроме того, алгоритм прямого вывода позволяет уменьшить объем вычислений по сравнению с традиционным алгоритмом за счет целенаправленности вывода, а алгоритм обратного вывода – за счет возможности рассмотрения в качестве цели вывода на каждом шаге не отдельного значения, а области значений.

Для поиска оптимальных решений на модели функциональных зависимостей атрибутов предложены алгоритмы оптимизации методом прямого и обратного вывода. Целевое состояние задается ограничениями $\{D^{Og}(a_i)\}$, $a_i \in \{a_i\}^{Og}$, а также критерием эффективности a_i^{kr} , на множестве значений которого задается отношение доминирования: $d_1 \succ \dots \succ d_k \succ \dots \mid d_k \in D(a_i^{kr})$. Необходимо найти вариант, т.е. объект o_k^* , содержащий определенную комбинацию значений атрибутов, обеспечивающую выполнение ограничений и достижение наилучшего значения критерия $d_k^*(a_i^{kr})$. В общем случае вариантов с одним и тем же наилучшим значением критерия может быть несколько.

При поиске решения методом прямого вывода генерируются различные комбинации значений базовых атрибутов $\{d(a_i) \mid a_i \in \{a_i\}^B\}$, для каждой из которых с помощью алгоритма прямого вывода для решения задачи интерпретации определяется соответствующие значения атрибутов, определяющих целевое состояние. В процессе выполнения алгоритма проверяется выполнение ограничений для атрибутов $a_i \in \{a_i\}^{Og}$, а также сравнивается значение критерия для текущего варианта $d_k(a_i^{kr})$ с текущим наилучшим значением критерия $d_k^*(a_i^{kr})$. Если очередной вариант улучшает значение критерия, то он запоминается и текущим наилучшим значением критерия становится значение критерия для данного варианта. Работа алгоритма продолжается, пока не будут перебраны все комбинации значений базовых атрибутов.

При поиске методом обратного вывода сначала с помощью алгоритма обратного вывода для решения задачи поиска допустимого решения ищется комбинация значений базовых атрибутов, обеспечивающая выполнение ограничений и достижение наилучшего значения критерия, затем (если решение не находится) – достижение ближайшего к наилучшему значению критерия и т.д.

Выбор алгоритма решения задачи оптимизации связан с вычислением трудоемкости алгоритмов. Под трудоемкостью алгоритма будем понимать ко-

личество обращений к правилам. Максимальная трудоемкость алгоритма прямого вывода определяется по формуле:

$$T_{pr} = \prod_{i=1}^m z_i \times (k_{m+1} + (v_{m+1} \times k_{m+2}) + \dots + (v_{m+1} \times \dots \times v_{n-1} \times k_n)),$$

где z_i – количество значений i -го базового атрибута, $i = \overline{1, m}$ (m – количество базовых атрибутов); k_j – количество правил для j -го атрибута из «пути вывода», v_j – количество «ветвлений» для j -го атрибута, $j = \overline{m+1, n}$ (n – общее количество атрибутов).

Максимальная трудоемкость алгоритма обратного вывода, определяется по формуле:

$$T_{ob} = z_n \times (k_n + (w_n \times k_{n-1}) + \dots + (w_n \times \dots \times w_{m+2} \times k_{m+1})),$$

где z_n – количество значений n -го атрибута (критерия); w_j – количество «ветвлений» для j -го атрибута в «пути вывода», $j = \overline{n-m, n}$.

При этом предполагается, что правила не содержат формулы и процедуры. Если же некоторое правило включает формулу или процедуру, то его обработка эквивалентна обработке совокупности «обычных» правил, общее количество которых равно количеству комбинаций значений атрибутов-аргументов. На рис. 9. приведены графики зависимости максимальной трудоемкости алгоритмов прямого и обратного вывода от количества базовых атрибутов для модели, имеющей следующие характеристики: $n = 10$, $\forall i \ z_i = 5$; $\forall j \ k_j = 25$; $v_j = 2$; $w_j = 5$.

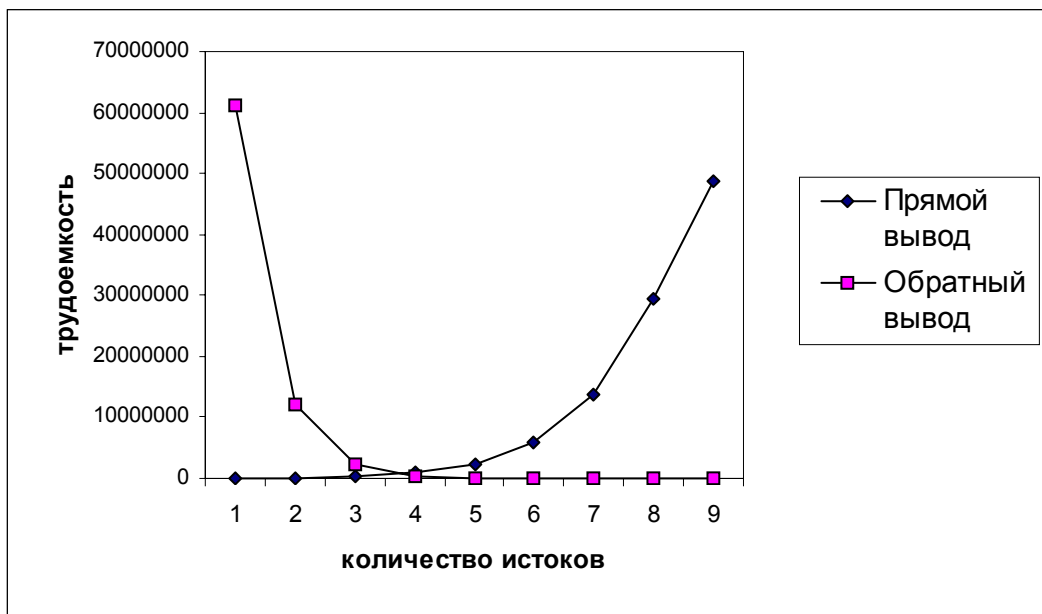


Рис. 9. Зависимость трудоемкости алгоритмов прямого и обратного вывода от количества истоков в сети

Как видно из графика, что решение задачи оптимизации методом прямого вывода эффективно при небольшом количестве базовых атрибутов и методом обратного вывода – когда количество базовых атрибутов значительно превышает число остальных.

Алгоритм формирования оценочных зависимостей в составе модели функциональных зависимостей решает две основные проблемы, связанных с построением оценочных отношений: преобразование значений атрибутов-аргументов, измеряемых на некоторой предметной шкале (шкале абсолютных значений), в значения на универсальной шкале $[0;1]$ и определение степеней влияния атрибутов-аргументов на атрибут-функцию.

Первая проблема решается путем сопоставления исходным атрибутам $a_j (j = \overline{1, n})$ атрибутов $a_i (i = \overline{1, n})$, измеряемых на универсальной шкале (рис. 10). Если исходный атрибут a_j имеет тип $\langle \text{Integer} \rangle$ или $\langle \text{Real} \rangle$, то преобразование значений осуществляется на основе функции отображения $\pi: D(a_j) \rightarrow [0; 1]$. Если исходный атрибут a_j имеет тип $\langle \text{String} \rangle$, то для каждого из лингвистических значений $d_k \in D(a_j)$ строится функция принадлежности μ_{d_k} на универсальной шкале, и для преобразования используется процедура дефаззификации.

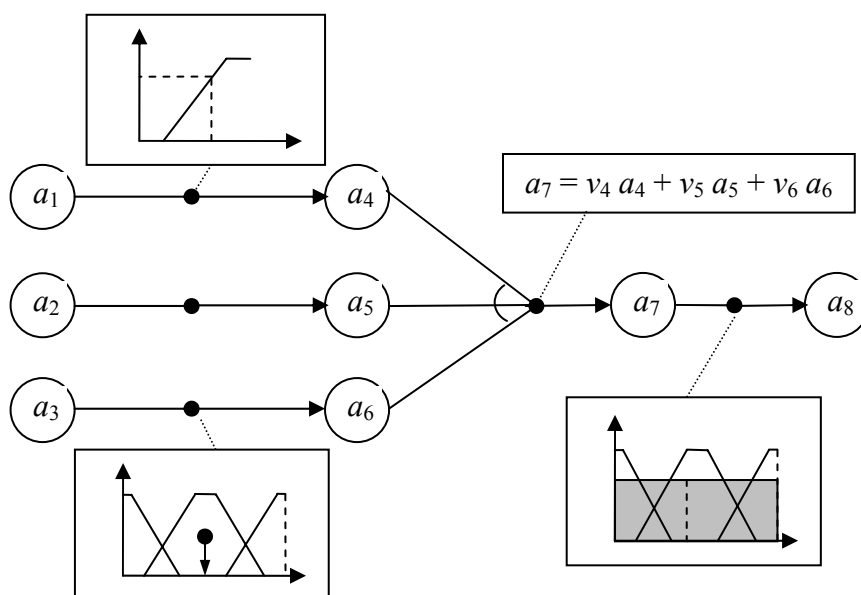


Рис. 10. Пример сети функциональных зависимостей, содержащей отношение оценочной зависимости

Для выявления степеней влияния v_i атрибутов-аргументов $a_i (i = \overline{1, n})$ на атрибут-функцию a_0 предлагается использовать один из методов выявления предпочтений, например, метод парных сравнений или ранжирования. Оценка степени согласованности выявленных приоритетов интерпретируется, как фактор уверенности правила.

Для преобразования балльного значения атрибута-функции в лингвистическое в модель водится еще один атрибут, принимающий значения типа $\langle \text{String} \rangle$. Определение его значения может осуществляться с помощью процедуры фаззификации. При этом, если лингвистическое значение находится по нечеткому балльному значению оценки, то предварительно функции принад-

лежности должны быть «усечены» на уровне вычисленного фактора уверенности.

Модель функциональных отношений, содержащая отношения оценочной зависимости, может использоваться для решения двух задач: для оценки состояния компоненты системы и для оценки влияния базового атрибута на целевой атрибут. И в том, и в другом случае используется алгоритм прямого вывода на модели зависимости. Для первой задачи, таким образом, вывод осуществляется аналогично алгоритму нечеткого вывода Сугено. Однако исходные данные могут быть представлены не только в виде нормированных балльных значений, но и в виде значений, измеряемых на некоторой «предметной» шкале. Решение второй задачи можно сравнить с вычислением глобального приоритета методом анализа иерархий (МАИ) Саати. Однако, в отличие от МАИ предложенный алгоритм позволяет определять не только оценку влияния базового атрибута на целевой атрибут, но и фактор уверенности в данной оценке, а также лингвистическое значение оценки. В процессе вывода могут использоваться различные правила, учитывающие различные условия, различные мнения экспертов (групп экспертов) и различные степени согласованности мнений экспертов

Для модели координации разработан алгоритм нахождения оптимального сочетания локальных вариантов дочерних подсистем. Исходными данными являются: выбранный обобщенный вариант реализации материнской системы $o(s_0)$, межподсистемные и межуровневые зависимости. Межуровневые зависимости представлены функцией затрат и функцией качества. Первая связывает атрибут материнской подсистемы z , отражающий глобальные затраты на реализацию варианта подуровня, с атрибутами z_i , отражающими локальные затраты дочерних подсистем: $z = \sum_i^n z_i$ (при этом для z задана допустимая область

$D_k(z) = [0, z^*]$). Вторая связывает критерий эффективности подуровня K , в качестве которого выступает критерий материнской подсистемы, с локальными критериями K_i дочерних подсистем: $K = \sum_i^n v_i \cdot K_i$ (при этом $\sum_i^n v_i = 1$). Межпод-

системные зависимости, устанавливаемые для каждой пары дочерних подсистем s_1 и s_2 , связанных отношением ассоциации, как правило, представлены в виде функций атрибутов одной подсистемы от атрибутов другой: $(a_0(s_1)) R^{fd} (a_1(s_2), \dots, a_n(s_2))$. Необходимо найти непротиворечивое сочетание локальных вариантов дочерних подсистем $\langle o(s_1), o(s_2), \dots, o(s_n) \rangle$, обеспечивающее наилучшее значение критерия эффективности подуровня и удовлетворяющее ограничению, наложенному на суммарные затраты.

В силу монотонности функции качества следует стремиться к нахождению оптимальных в смысле локальных критериев вариантов дочерних подсистем в условиях ограничений. Таким образом, оптимальные варианты для дочерних подсистем определяются на частных моделях, но при этом материнская подсистема воздействует на локальные задачи оптимизации через изменение ограничений на локальные затраты. Причем логично большую «свободу» в вы-

боре оптимального варианта предоставить подсистемам, которые дают больший вклад в критерий подуровня. Таким образом, допустимый уровень затрат распределяется между дочерними подсистемами в соответствии с их «весом». Для каждой дочерней подсистемы формируется ограничение $z_i \leq z_i^*$, где $z_i^* = v_i \cdot z^*$. Ограничение вносится в частную модель подсистемы

Для учета при решении локальных задач оптимизации межподсистемных зависимостей предлагается использовать метод «распространения ограничений», в котором выбор оптимальных вариантов для каждой из дочерних подсистем осуществляется последовательно, причем для первой подсистемы решение находится без учета межподсистемных зависимостей, а для остальных – с учетом уже выбранных вариантов для предыдущих подсистем. Последовательность перебора подсистем $Q = \langle s_1, s_2, \dots, s_{n-1}, s_n \rangle$ формируется по «слоям» аналогично методу нахождения «пути вывода» в алгоритмах прямого и обратного вывода на модели зависимостей атрибутов. Первый «слой» составляет подсистема, имеющая наибольший «вес». За ней следуют подсистемы второго «слоя», связанные с подсистемой предыдущего «слоя», в порядке убывания веса и т.д.

Затем из последовательности Q выбирается очередная подсистема s_i и определяется локальный оптимальный вариант $o(s_i)$ с учетом ограничений на локальные затраты и межподсистемных ограничений. При этом значения атрибутов уже рассмотренных подсистем, соответствующие выбранным вариантам этих подсистем, выступают ограничениями для текущей подсистемы, атрибуты же еще не рассмотренных подсистем, рассматриваются как свободные переменные.

При переходе к следующей подсистеме осуществляется перераспределение уровня допустимых затрат между подсистемами. Находится разница между уровнем допустимых затрат для текущей подсистемы и реальными затратами на выбранный оптимальный вариант: $\Delta z = z_i^* - z_i$, которая перераспределяется между оставшимися подсистемами. Для каждой из подсистем $s_j, j = \overline{i+1, n}$, стоящих в последовательности Q следом за текущей подсистемой s_i , к ранее определенному уровню допустимых затрат z_j^* добавляется «остаток»

$$\Delta z_j = \Delta z \cdot \frac{v_j}{v_{i+1} + \dots + v_n}. \text{ Соответствующие ограничения на локальные затраты}$$

вносятся в частные модели подсистем. После того, как будут перебраны все подсистемы, проверяется, имеется ли «остаток». Если $\Delta z > 0$, то принимается решение, либо закончить поиск, считая найденную комбинацию вариантов подсистем квазиоптимальной, либо перераспределить остаток между подсистемами и пройти еще одну итерацию.

Рассмотренный алгоритм не гарантирует нахождения оптимального варианта подуровня, однако он позволяет находить вариант, близкий к оптимальному, за приемлемое время в случае, когда решение задачи оптимизации методом полного перебора нецелесообразно.

Для того, чтобы облегчить разработчику выбор методов и моделей, используемых на каждой стадии процесса разработки целевой программы, предложен типовой регламент, основанный на объектно-ориентированной методологии моделирования. В соответствии с принципом декларативности регламент представлен в виде совокупности рекомендаций для каждого из основных этапов типовой последовательности относительно содержания этапа, вида формируемых моделей и спектра методов, которые могут быть использованы разработчиком.

На подготовительном этапе формируется объектная модель самого процесса разработки с целью планирования работ. На рис. 11 представлен фрагмент модели компонент процесса, содержащий в качестве подсистем основные этапы и подэтапы.

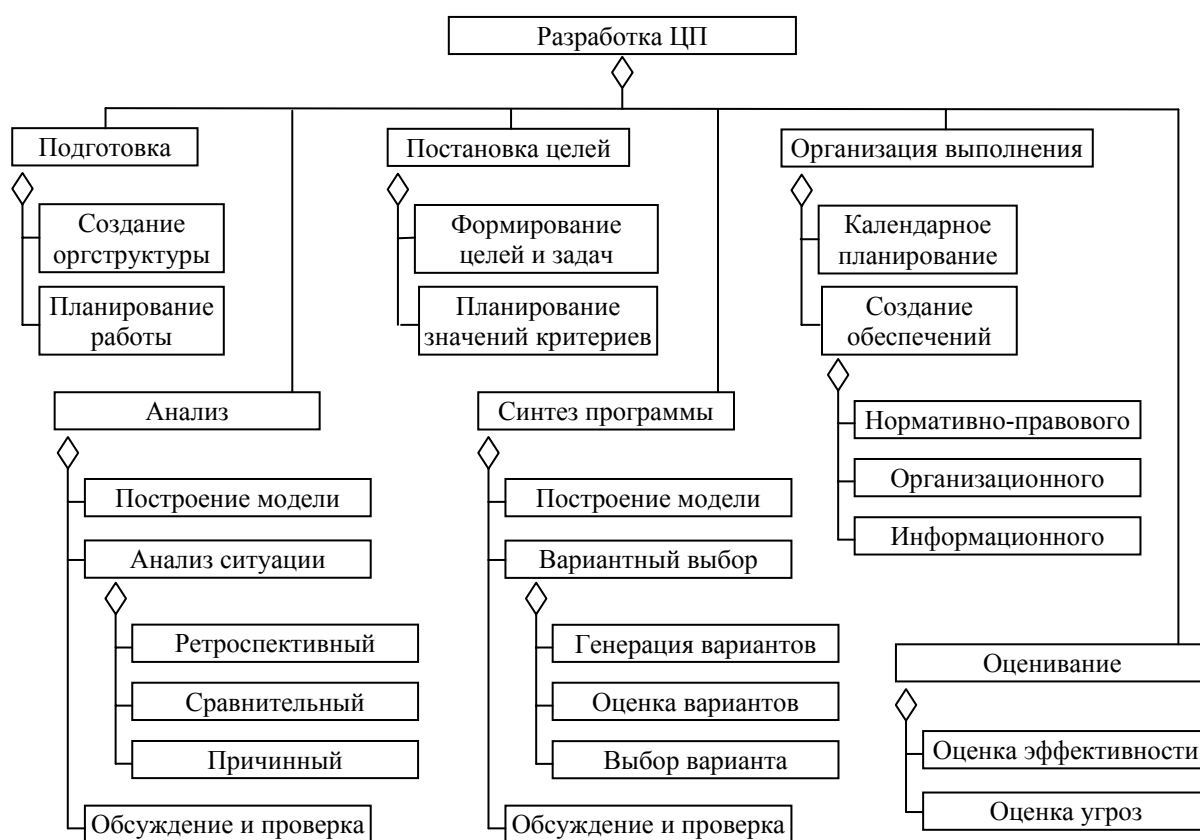


Рис. 11. Модель компонент процесса разработки программы

Для формирования модели использовались стандартные основания декомпозиции. Каждому подуровню модели компонент может быть сопоставлена диаграмма отношений ассоциации, отражающая взаимосвязи между этапами. Для создания структурированного описания этапов подсистемам дерева компонент сопоставляются классы, на основе которых формируются варианты реализации этапов и выбирается наиболее перспективная комбинация вариантов, согласованных по срокам, общей стоимости проекта и т.д.

На этапе анализа исследуется проблемосодержащая система. Сначала формируется модель компонент, каждой компоненте которой сопоставляются

классы. Для построения иерархии подсистем могут быть использованы стандартные основания декомпозиции, а для составления классов – типовые описания, хранящиеся в библиотеке типовых моделей. Классы, как правило, в качестве базовых атрибутов содержат первичные количественные показатели, характеризующие деятельность компоненты системы, а в качестве целевых – интегрированные показатели. Для задания функциональных зависимостей между атрибутами одной компоненты и различных компонент используются соответственно модели зависимостей атрибутов и координационные модели.

На базе классов создаются экземпляры описаний в виде мультиобъектов и выполняются определенные методы в зависимости от вида анализа. Так, при ретроспективном анализе формируются мультиобъекты по временному признаку, отражающие состояния компонент в различные моменты времени, и на основе сравнения объектов выявляются тенденции, закономерности. При сравнительном анализе формируются мультиобъекты по признаку «группа», содержащие описания аналогов, а также «идеальных» вариантов компонент, осуществляется сравнение и оценка введенных объектов.

На этапе постановки целей формулируются цели и задачи создаваемой программы, исходя из проблем, выявленных на этапе анализа. При формировании модели компонент выделяются подсистемы, соответствующие различным аспектам целеполагания (политическому, экономическому, социальному, экологическому) или соответствующие группам целей заинтересованных сторон, имеющих отношение к рассматриваемой проблематике. Класс подсистемы кроме атрибутов, содержащих словесные формулировки целей и задач, включает в качестве присоединенной компоненты описания критериев. Для отражения плановых значений критериев формируется мультиобъект по временному признаку.

Основным содержанием этапа синтеза программы является разработка комплекса мероприятий исходя из целей и задач, выявленных на предыдущем этапе. Разработка осуществляется иерархически. На верхних уровнях осуществляется выработка стратегических решений для программы в целом и отдельных подпрограмм, а на нижних уровнях осуществляется разработка конкретных мероприятий. Иерархия компонент (подпрограмм) может соответствовать иерархии компонент проблемосодержащей системы, построенной на этапе анализа. Каждой компоненте сопоставляется класс, в пространстве атрибутов которого осуществляется формирование и выбор альтернативных вариантов. В состав атрибутов включаются атрибуты, описывающие сценарии действий (стратегии, мероприятия) на содержательном уровне, а также количественные атрибуты, отражающие показатели эффективности, затраты, временные характеристики и т.д.

Выбор оптимального варианта отдельной компоненты осуществляется на модели объектов компоненты, возможно дополненной моделью зависимостей атрибутов. Процесс поиска решения предполагает генерирование различных вариантов, их оценку и выбор оптимального варианта. При этом разработчик может использовать любой из доступных ему методов, руководствуясь рекомендациями относительно выбора методов. Задача иерархического многоуровневого выбора может быть сведена к совокупности задач выбора на двухуров-

невых моделях, содержащих на нижнем уровне частные модели подсистем одного подуровня, на верхнем уровне – модель материнской подсистемы. Для согласования вариантов дочерних подсистем друг с другом и с вариантом материнской подсистемы используется координационная модель, содержащая межподсистемные и межуровневые зависимости. В качестве стратегии «прохождения» иерархии подсистем рекомендуется использовать смешанную стратегию, сочетающую нисходящий и восходящий методы.

Заканчивается каждый этап, как правило, обсуждением и проверкой полученных результатов, после которой может быть принято решение о возврате на любой из предыдущих этапов с целью корректировки его результатов.

Четвертая глава посвящена описанию инструментальных средств, созданных для поддержки информационной технологии разработки целевых программ.

Для построения объектно-ориентированных моделей систем и их использования для анализа и выработки оптимальных решений разработан программный комплекс «OMPS». Комплекс позволяет создавать визуальные модели систем, содержащие: описания компонент системы в виде мультиобъектов, созданных на базе классов; отношения между мультиобъектами; функциональные зависимости между атрибутами; а также осуществлять расчеты значений атрибутов на основе правил, сопоставленных функциональным зависимостям.

Комплекс имеет многоуровневую архитектуру, основанную на клиент-серверном подходе (рис. 12), что обеспечивает гибкость и модульность структуры, возможность параллельной работы, возможность интеграции со специализированными приложениями.

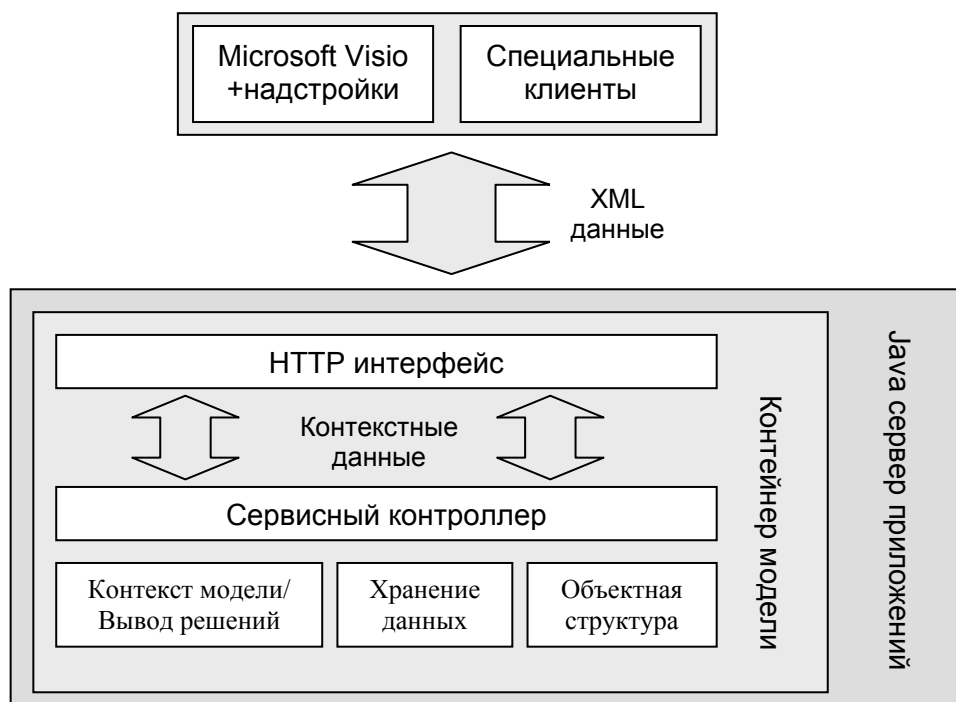


Рис. 12. Архитектура комплекса «OMPS».

Серверную часть инструментального комплекса составляет платформенно-независимое приложение (контейнер модели), разработанное на основе Java-технологий стандарта J2EE и обладающее сетевыми интерфейсами для взаимодействия с клиентским программным обеспечением пользователя. Унификация формата передачи данных между клиентской и серверной частями в формате XML, позволяет, при необходимости, создать специализированное клиентское приложение для работы с конкретной моделью или проектом.

В качестве базового инструмента пользователя для редактирования и настройки модели выступает пакет Visio из состава Microsoft Office с дополнительными программными надстройками. Данная среда обладает встроенными возможностями не только для построения разнообразных диаграмм, но и их гибкой настройки для конкретных задач, в том числе с помощью интегрированного языка разработки VBA. Специальное клиентское программное обеспечение, реализующее функции редактирования классов, зависимостей и подсистем, а также генерации мультиобъектов, обеспечивает более удобный интерфейс, нежели стандартный пакет Visio.

Созданная модель сохраняется в виде XML-файла и отправляется серверному приложению для обработки. Серверная часть осуществляет обработку полученных от клиента через HTTP-интерфейс XML-структур и вызывает соответствующие сервисные функции. Сервисный контроллер отвечает за реализацию всей логики системы, делегирует операции с данными подсистеме хранения данных, вызывает механизмы вывода решений для заполненной объектной модели и отправляет обработанные контекстные данные верхним уровням для представления пользователю в виде XML-файла результатов.

Для поддержки процессов моделирования функциональных зависимостей между атрибутами, описывающими сложные системы и их компоненты, разработан инструментальный комплекс «WinEsisp». Комплекс относится к классу оболочек экспертных систем и предоставляет непрограммирующему пользователю удобные средства для быстрого создания прототипов экспертных систем в статических предметных областях. Создание базы знаний, содержащей модель функциональных отношений, осуществляется путем ввода информации в диалоговом режиме. Редактор базы знаний содержит ряд специализированных редакторов – для ввода информации об атрибутах, для ввода правил-продукций, формул, построения функций принадлежности и т.д. Встроенный механизм логического вывода может быть использован для решения задач интерпретации, поиска допустимого решения и поиска оптимального решения. Удобный пользовательский интерфейс и сервисные возможности (развитая справочная система, настройка опций, подсистема объяснений и формирования отчетов) облегчает пользователю создание базы знаний, ее тестирование и проведение экспериментов.

Разработан также ряд специализированных инструментариев, используемых для решения «узких» задач: инструментальное средство моделирования организационных структур, система формирования задач управления организацией, средство выявления предпочтений экспертов.

В **пятой главе** описываются результаты применения созданной информационной технологии, методологии моделирования и инструментальных средств для разработки целевых программ и информационных систем.

Технология используется в Томском центре управления энергосбережением для синтеза региональных систем в сфере энергосбережения. В частности, на ее основе была разработана программа углубления энергетической эффективности на территории Томской области на 2004-2008 г.г. Проведенный в ходе разработки ретроспективный и сравнительный анализ состояния сферы энергосбережения с использованием объектной модели позволил оценить уровень энергосбережения региона; выявить основные тенденции и проблемы. Была сформирована система политических, экономических, социальных и экологических целей и спланированы значения критериев для контроля хода выполнения программы. Были выявлены и оценены с использованием модели зависимостей атрибутов основные направления программы, создающие условия для реализации потенциала энергосбережения, а также разработан комплекс мероприятий, стимулирующих эффективное использование энергии

С использованием технологии была создана программа реформирования системы ресурсообеспечения населения Томской области. Была сформирована объектная модель системы ресурсообеспечения региона, отражающая ее существующее и ретроспективные состояния. Проведенный на модели анализ позволил выявить проблемы, основными из которых были признаны: постоянно возрастающая стоимость услуг; низкое качество услуг; высокая степень износа основных фондов; неудовлетворительное финансовое состояние системы ресурсообеспечения. В ходе причинного анализа были сформированы иерархии причин в виде модели зависимостей атрибутов и выявлены коренные причины, приведшие к неудовлетворительным результатам деятельности системы ресурсообеспечения. В соответствии с основными группами причин, а также с учетом целей всех заинтересованных сторон (потребителей услуг, ресурсоснабжающих организаций, областных и муниципальных органов управления и др.) были выделены четыре направления реформирования, для каждого из которых был разработан комплекс программных мероприятий.

На основе объектно-ориентированной информационной технологии разработки ЦП была создана технология формирования типовой программы энергетической безопасности региона. Для анализа энергобезопасности предлагается объектная модель, содержащая структурированное множество показателей, характеризующих различные аспекты безопасности (показатели обеспеченности энергоресурсами, финансовые, экологические показатели и др.). Среди показателей выделяются: первичные показатели (данные мониторинга), индикаторы безопасности, индексы изменения индикаторов и оценки ситуации. Разработана модель зависимостей атрибутов, позволяющая определять оценку уровня энергетической безопасности региона на основе данных мониторинга. Разработку мероприятий поддержки безопасности предлагается осуществлять с использованием типовой модели оценки угроз энергетической безопасности. Для организации выполнения программы создан типовой перечень нормативно-

правовых актов и типовая организационная структура системы управления реализацией программы.

Еще одной сферой применения информационной технологии разработки целевых программ явилось проектирование (перепроектирование) систем в инновационной сфере. Так, на ее основе была разработана программа совершенствования деятельности Томского технопарка, а также проект регионального учебно-научно-инновационного комплекса. Использование информационной технологии позволило оценить существующее состояние инновационной сферы, выявить систему целей и стратегий их достижения, разработать комплекс мер, а также организационные структуры, обеспечивающие их реализацию.

Объектно-ориентированная методология моделирования OMSD и инструментальные средства использовались также при создании информационных систем, предназначенных для анализа и планирования экономической деятельности. На базе комплекса «WinEsis» была разработана ИС оценки финансово-экономического состояния предприятия, которая позволяет осуществлять статический и динамический анализ и прогнозирование финансовой устойчивости предприятия, а также выбор параметров функционирования для поддержания или увеличения устойчивости. Анализ результатов применения системы для ряда предприятий показал достаточно высокую согласованность оценок, полученных с помощью системы, с заключениями экспертов. С помощью инструментального комплекса «OMPS» была создана ИС формирования программы инвестирования в разработку нефтяных месторождений, позволяющая осуществлять выбор варианта разработки месторождений в условиях неопределенности информации о предполагаемой экономической ситуации, геологических и технологических рисках. Применение системы облегчает формирование программы за счет использования экспертных знаний. При этом разработчик может предполагать различные варианты развития экономической ситуации и изменения технологических, геологических, производственных факторов.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В диссертационной работе по результатам выполненных теоретических и экспериментальных исследований создана информационная технология разработки целевых программ на основе объектно-ориентированной методологии моделирования. Предложенная технология объединяет подходы, выработанные в рамках таких научных направлений, как системный анализ, инженерия знаний, CASE-технологии проектирования информационных систем, реинжиниринг бизнес-процессов и вносит значительный вклад в развитие теории анализа и синтеза проблеморазрешающих систем. Внедрение разработанных автором моделей, алгоритмов и программно-инструментальных средств имеет существенное значение для решения проблемы повышения эффективности процесса формирования целевых программ.

Перечислим основные результаты, в которых отражена научная новизна и практическая ценность данной работы:

1. Предложены принципы создания информационной технологии разработки целевых программ, учитывающие особенности процесса разработки и удовлетворяющие требованиям, предъявляемым к технологии. На основе выдвинутых принципов разработана схема автоматизированной разработки, предоставляющая возможность комплексировать различные методы анализа и синтеза систем, используя в качестве интегрирующего начала декларативную модель в виде совокупности взаимосвязанных моделей, описанных на едином объектно-ориентированном языке представления знаний.

2. Разработана методология объектно-ориентированного моделирования систем OMSD. Структура объектно-ориентированной модели системы (проблемосодержащей, проблеморазрешающей или системы разработки) представлена в виде совокупности взаимосвязанных моделей пяти видов:

- модели классов, отражающей структуру множества классов, используемых для описания компонент системы;
- модели объектов, отражающей описание множества вариантов реализации компоненты системы;
- модели зависимостей атрибутов, отражающей отношений функциональной зависимости между атрибутами компоненты системы;
- модели компонент системы, отражающей структуру системы в виде дерева подсистем и элементов;
- модели координации, отражающей отношения зависимости между атрибутами дочерних подсистем и материнской системы.

Таким образом, модель позволяет отражать и описывать в декларативном виде различные аспекты системы: структурный и характеристический (фактуальное знание), логический и процедурный (операционное знание), типовой и стандартный (интенционал), конкретный и ситуационный (экстенционал).

3. Разработаны алгоритмы формирования объектно-ориентированной модели системы и поиска решений на модели:

- алгоритм генерации объектов, основанный на модифицированном методе морфологического анализа. Алгоритм по сравнению с традиционным методом обеспечивает рассмотрение обозримого количества вариантов на каждом шаге, более логичную последовательность формирования вариантов, более четкую процедуру исключения вариантов;
- алгоритм выбора оптимальных решений в условиях неопределенной информации о возможных состояниях среды, аналогичный методу «математического ожидания» для решения задач типа «игры против природы». Использование факторов уверенности позволяет применять алгоритм в ситуациях, когда о реализациях недетерминированных исходов нельзя собрать статистику для построения вероятностных распределений;
- алгоритмы прямого и обратного вывода на модели функциональных зависимостей атрибутов, позволяющие находить все возможные варианты решения задачи интерпретации и задачи поиска допустимого решения. Предложенные алгоритмы позволяют обрабатывать правила, содержащие разнообразные формулы и процедуры, и обеспечивают более высокую скорость вычисле-

ний по сравнению с традиционными алгоритмами прямого и обратного вывода на системе продукций;

- алгоритмы поиска оптимальных решений на модели функциональных зависимостей атрибутов, использующие прямой и обратный вывод. Проведенные исследования трудоемкости алгоритмов позволили выявить условия их эффективности;

- алгоритм формирования оценочных зависимостей атрибутов в составе модели функциональных отношений. Модель, построенная с помощью данного алгоритма, может использоваться для оценки состояния компоненты системы и для оценки влияния базового атрибута на целевой атрибут. В первом случае вывод осуществляется аналогично алгоритму нечеткого вывода Сугено, во втором - аналогично процессу вычисления глобального приоритета методом анализа иерархий Саати;

- алгоритм поиска оптимального сочетания локальных вариантов дочерних подсистем, использующий метод распространения ограничений для выполнения межподсистемных ограничений и метод перераспределения затрат для выполнения ограничений со стороны материнской подсистемы. Алгоритм позволяет находить вариант, близкий к оптимальному, за приемлемое время в случае, когда решение задачи оптимизации методом полного перебора нецелесообразно.

4. Предложен типовой регламент процесса разработки целевой программы, содержащий рекомендации относительно выполнения основных типовых этапов жизненного цикла разработки. Рекомендации содержат указания по составу этапов, их последовательности, а также используемым на каждом из этапов методам формирования объектно-ориентированной модели и поиска решений на ней.

5. Разработаны инструментальные средства поддержки, призванные облегчить процесс формирования целевой программы за счет автоматизации процедур создания модели системы и поиска решений на модели:

- комплекс «WinEsisp», поддерживающий процессы моделирования функциональных зависимостей атрибутов. Комплекс относится к классу оболочек экспертных систем и предоставляет непрограммирующему пользователю удобные средства для быстрого создания прототипов экспертных систем: специализированный редактор базы знаний, встроенный механизм логического вывода и сервисные возможности;

- комплекс «OMPS» для построения объектно-ориентированных моделей сложных систем и их использования в процессе анализа и выработки оптимальных решений. Комплекс имеет многоуровневую архитектуру, основанную на клиент-серверном подходе, что обеспечивает гибкость и модульность структуры, возможность параллельной работы, возможность интеграции со специализированными приложениями;

- инструментальное средство моделирования организационных структур, позволяющее строить графические модели (взаимодействия с окружающей средой, деятельности организации, структуры организации, дерева целей), использовать в режиме экспертной системы рекомендации относительно выбора

типа оргструктуры, осуществлять экспертную оценку вариантов структуры по множеству критериев;

- инструментальная система формирования задач управления организацией, позволяющая автоматически генерировать задачи путем перебора элементов, полученных в результате декомпозиции управляемого процесса и процесса управления организацией. Для преобразования сгенерированных формальных записей задач управления в предложения на естественном языке система содержит лингвистический транслятор, работающий по принципу использования шаблонов;

- инструментальное средство выявления предпочтений экспертов, которое может быть использовано при формировании оценочных зависимостей атрибутов, при генерации объектов, при выборе оптимального варианта реализации компоненты системы. Средство позволяет находить количественные значения относительного превосходства объектов (атрибутов) в виде нормализованных балльных оценок с помощью одного из методов экспертного оценивания – ранжирования или парных сравнений.

6. Созданная информационная технология разработки целевых программ была использована для синтеза:

- программы углубления энергоэффективности на территории Томской области;
- региональной программы ресурсобеспечения населения;
- программы энергетической безопасности региона;
- программы совершенствования деятельности Томского технопарка;
- проекта создания регионального учебно-научно-инновационного комплекса;
- информационной системы оценки финансово-экономического состояния предприятия;
- информационной системы инвестирования в разработку нефтегазовых месторождений.

Результаты применения технологии для формирования различных целевых программ и информационных систем показали ее эффективность, в частности обеспечение комплексности рассмотрения проблем, учет интересов всех заинтересованных сторон, обоснованность и взвешенность предложений и рекомендаций, а также сокращение трудоемкости разработки.

ПУБЛИКАЦИИ

По теме диссертации опубликовано 62 научные работы. Основные результаты опубликованы в следующих работах:

1. Силич М. П. Системная технология: объектно-ориентированный подход / М. П. Силич. – Томск : Том. гос. ун-т систем управления и радиоэлектроники, 2002. – 224 с.
2. Силич М. П. Объектная модель как основа информационной технологии проектирования сложных социально-экономических систем / М. П. Силич // Кибернетика и системный анализ. – 2005. - № 5. – С. 136-146.

3. Силич М. П. Формирование и выбор вариантов на объектной модели компоненты сложной системы / М. П. Силич // Автоматизация и современные технологии. – 2005. – №4. – С. 21-26.
4. Силич М. П. Использование объектно-ориентированного подхода для формирования предложений на естественном языке // М. П. Силич // Научно-техническая информация. Сер. 2. Информационные процессы и системы. – 2005. – № 2. – С. 8-12.
5. Силич М. П. Инструментальный комплекс для создания экспертных систем, использующих модели функциональных отношений / М. П. Силич, Н. Ю. Хабибулина // Известия Томского политехнического университета. – 2005. – Т. 308, № 2. – С. 149-152.
6. Силич М. П. Поиск решений на модели функциональных отношений / М. П. Силич, Н. Ю. Хабибулина // Информационные технологии. – 2004. – № 9. – С. 27-33.
7. Силич М. П. Объектно-ориентированная модель сложной системы / М. П. Силич // Ползуновский вестник. – 2004. – №3. – С. 93-98.
8. Силич М. П. Поиск решения на двухуровневой объектной модели сложной системы / М. П. Силич // Ползуновский вестник. – 2004. – №3. – С. 104-108.
9. Силич М. П. Использование слабоформализуемых зависимостей в модели функциональных отношений / М. П. Силич // Известия Томского политехнического университета. – 2004. – Т. 307. – №6. – С. 21-25.
10. Силич М. П. Разработка программы повышения энергетической эффективности на территории Томской области / М. П. Силич // Известия вузов. Проблемы энергетики. – 2004. – № 5-6. – С. 71-78.
11. Силич М. П. Объектная модель выбора инвестиционных проектов разработки нефтегазовых месторождений / М. П. Силич, Г. В. Стародубцев // Автоматизация, телемеханизация и связь в нефтяной промышленности. – 2004. – № 11. – С. 16-21.
12. Силич М. П. Инструментальное средство построения объектно-ориентированных моделей для оптимального проектирования сложных систем / М. П. Силич // Вестник Томского государственного университета. – 2004. – Т. 284. – С. 217-221.
13. Силич М. П. Информационная система оценки финансово-экономического состояния предприятия / М. П. Силич, Н. Ю. Хабибулина, З. В. Горбатов // Вестник Томского государственного университета. – 2004. – Т. 284. – С. 90-95.
14. Силич М. П. Метод объектного моделирования для проектирования сложных систем / М. П. Силич, В. А. Силич // Автоматизация и современные технологии. – 2003. – №4. – С. 14-21.
15. Силич М. П. Проектирование сложной системы на основе объектно-ориентированного подхода / М. П. Силич, В. А. Силич // Известия Томского политехнического университета. – 2003. – Т. 306. – №2. – С. 99-103.
16. Силич М. П. Разработка концепции реформирования сферы ресурсобеспечения населения на основе системной технологии / М. П. Силич, С. А.

- Косяков, В. В. Литвак и др. // Известия Томского политехнического университета. – 2003. – Т. 306. - № 1. – С. 148-152.
17. Силич М.П. Методология иерархического семантического моделирования / М.П. Силич // Информационные технологии в территориальном управлении, промышленности, образовании: Сб. статей / Под ред. Ю.П. Ехлакова. – Томск: изд-во Том. гос. ун-та систем управления и радиоэлектроники, 2005. – С. 14-24.
 18. Силич М. П. Иерархический выбор варианта реализации сложной системы на основе объектно-ориентированного подхода / М. П. Силич, В. А. Силич // Труды междунар. симпозиума «Информационные и системные технологии в индустрии, образовании и науке». – Караганда, 2003. – С. 232-234.
 19. Силич М. П. Методология объектного моделирования как основа системной технологии / М. П. Силич // Информационные технологии в территориальном управлении, промышленности, образовании : Сб. статей. – Томск : Том. гос. ун-т систем управления и радиоэлектроники, 2002. – С. 26-38.
 20. Силич М. П. Концепция энергетической безопасности субъектов Федерации / М. П. Силич, В. В. Литвак, В. А. Силич, М. И. Яворский // Ресурсы регионов России. – 2001. – № 2. – С. 32-44.
 21. Силич М. П. Совершенствование деятельности технопарка на основе системной технологии / М. П. Силич, В. А. Силич, С. З. Ямпольский // Актуальные проблемы управления-2001 : Мат-лы междунар. науч.-практ. конф. – М., 2001. – С. 70-73.
 22. Силич М. П. Принципы создания системной технологии / М. П. Силич // Интеллектуальные автоматизированные системы проектирования, управления и обучения : Сб. статей / Под ред. В. П. Тарасенко. – Томск : изд-во НТЛ, 2000. – С. 214-223.
 23. Силич М. П. Система управления разработкой и реализацией программы социально-экономического развития Томской области / М. П. Силич, А. В. Винарский, Ю. И. Краснов и др. // Интеллектуальные автоматизированные системы проектирования, управления и обучения : Сб. статей / Под ред. В. П. Тарасенко. – Томск : изд-во НТЛ, 2000. – С. 193-199.
 24. Силич М. П. Технология обеспечения энергетической безопасности / М. П. Силич, С. А. Косяков, В. В. Литвак и др. // Энергосбережение и энергетическая безопасность регионов России : Мат-лы Всерос. совещания. – Томск, 2000. – С. 6-18.
 25. Силич М. П. Метод формирования гибридных моделей для построения экспертных систем / М. П. Силич // Автоматическое и автоматизированное управление сложными системами : Сб. статей / Под ред. В. П. Тарасенко. – Томск : изд-во Том. ун-та, 1998. – С. 126-131.
 26. Силич М. П. Экспертная система для проектирования сложных объектов на ранних этапах / М. П. Силич, В. А. Силич // Автоматическое и автоматизированное управление сложными системами : Сб. статей / Под ред. В. П. Тарасенко. – Томск : изд-во Том. ун-та, 1998. – С.120-125.

27. Silich M. P. Analysis of solution searching algorithms on the model of functional relations / M. P. Silich, V. A. Silich, N. U. Khabibulina // Proceedings of the 8th Korea- Russia International Symposium on Science and technology (KORUS 2004). – Tomsk, 2004. – vol 1. – P. 157-161.
Силич М.П. Анализ алгоритмов поиска решений на модели функциональных отношений / М. П. Силич, В. А. Силич, Н. Ю. Хабибулина // Доклады 8-го международного российско-корейского симпозиума по науке и технологии (КОРУС 2004). – Томск, 2004. – т. 1. – С. 157-161.
28. Silich M. P. Hierarchical object model of variant choice / M. P. Silich, V. A. Silich // Proceedings of the 7th Korea- Russia International Symposium on Science and technology (KORUS 2003). – Ulsan, 2003. – P. 378-381.
Силич М.П. Иерархическая объектная модель вариантного выбора / М.П. Силич, В.А. Силич // Доклады 7-го международного российско-корейского симпозиума по науке и технологии (КОРУС 2003). – Ульсан, 2003. – С. 378-381.
29. Silich M. P. Object-oriented system technology / M. P. Silich, V. A. Silich // Proceedings of the 5th Korea- Russia International Symposium on Science and technology (KORUS 2001). – Tomsk, 2001. – P. 98-101.
Силич М.П. Объектно-ориентированная системная технология / М.П. Силич, В.А. Силич // Доклады 5-го международного российско-корейского симпозиума по науке и технологии (КОРУС 2001). – Томск, 2001. – С. 98-101.
30. Silich M. P. Use of object-oriented approach for automatization of the system analysis / M. P. Silich // Application of the Conversion Research Results for International Cooperation: Mat. of the third International Symposium SIB-CONVERS'99. – Tomsk, 1999. – P. 242-243.
Силич М.П. Использование объектно-ориентированного подхода для автоматизации системного анализа / М.П. Силич // Применение результатов конверсии научных исследований в международном сотрудничестве: Мат-лы 3-го международного симпозиума СИБКОНВЕРС'99. – Томск, 1999. – С. 242-243.
31. Silich M. P. Technology of territory governing in crisis conditions / M. P. Silich, V. V. Litvak, V. A. Silich, M. I. Javorsky // Abstracts of the third russian-korean international symposium KORUS'99. – Novosibirsk, 1999. – P. 131.
Силич М.П. Технология территориального управления в кризисных ситуациях / М.П. Силич, В.В. Литвак, В.А. Силич, М.И. Яворский // Тезисы 3-го международного российско-корейского симпозиума по науке и технологии (КОРУС'99). – Новосибирск, 1999. – С. 131.