ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО

ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

В Г. СМОЛЕНСКЕ

Кафедра: Вычислительной техники

Направление: 230100 – Информатика и вычислительная техника

Специальность: 230101 – Вычислительные машины, комплексы, системы и сети

«Способ и инструментальные программные средства многокомпонентного интерактивного моделирования комплексных задач обработки данных»

|  |  |
| --- | --- |
| Группа:  Студент:  Курс:  Номинация: | ВМ-12 (м)  Чандер Арсентий Петрович  6  Исследования в области технических наук |

Автор научной работы

/ / Чандер А.П.

СПИСОК СОКРАЩЕНИЙ

ИПС – инструментальные программные средства;

ПО–программное обеспечение, создаваемое на основе ИПС;

КЗОД – комплексная задача обработки данных

ВВЕДЕНИЕ

К комплексным задачам обработки данных относятся: задачи обработки изображений; иерархическая обработка данных, на каждом уровне которой осуществляется аккумулирование данных от нескольких источников и их преобразование; исследование квантовых переходов в гетерогенных структурах и многое другое. Данные задачи характеризуются тем, что их можно разделить на ряд отдельных операций (этапов), каждую из которых можно выполнить несколькими различными способами.

Подходом к решению таких комплексных задач, как правило, является обоснованный набор наиболее приемлемых алгоритмов выполнения каждого ее этапа. При этом необходимо учитывать не только эффективность реализации отдельных этапов обработки данных, но и учитывать эффективность использования совместно выполняемых пар, троек, в общем случае *n*-ок реализаций разных этапов задачи, выполненных совместно.

Типовым способом решения таких комплексных задач обработки данных является полный перебор всех возможных вариантов решения на каждом этапе, которые затем оцениваются, и определяются рациональные решения задачи. Недостаток этого способа очевиден – при большом количестве способов решения каждого этапа задачи общее число перебираемых вариантов стремительно возрастает, что сказывается на времени поиска рационального решения. Стоит отметить, что рациональное решение задачи ищется не на одном наборе входных параметров, а на большом количестве вариантов входных параметров.

В настоящий момент актуальным является создание способов программных средств многокомпонентного моделирования и автоматической генерации алгоритмов для выполнения отдельных этапов комплексной задачи обработки данных (КЗОД), а также моделей и методик оценки эффективности выполнения этапов обработки и всей задачи в целом.

*Объектом исследования* являются процессы моделирования комплексных задач обработки данных.

*Предметом исследования* являются способы и инструментальные программные средства интерактивного многокомпонентного моделирования комплексных задач обработки данных.

*Целью исследования* является повышение эффективности многокомпонентного моделирования комплексных задач обработки данных.

*Научная задачаисследования* состоит в разработке способа и инструментальных программных средств многокомпонентного интерактивного моделирования комплексных задач обработки данных.

Для достижения цели исследования решены следующие задачи:

* выполнен анализ комплексных задач обработки данных;
* осуществлен анализ методов и средств моделирования комплексных задач обработки данных;
* разработан способ многокомпонентного интерактивного моделирования комплексных задач обработки данных;
* созданы инструментальные программные средства многокомпонентного интерактивного моделирования комплексных задач обработки данных;
* выполнена оценка эффективности использования разработанных способа и программных инструментальных средств.

*Научная новизна* основных результатов работы заключается в следующем:

* разработан способ многокомпонентного интерактивного моделирования для решения комплексных задач обработки данных, предназначенный для программной реализации и использующий гибкий механизм как при оценке выполнения промежуточных этапов, так и всей задачи в целом с учетом различных стратегий при выборе наилучшего решения.

*Практическая значимость и реализация результатов работы*:

* разработаны структура и алгоритмы инструментальных программных средств многокомпонентного интерактивного моделирования комплексных задач обработки данных;
* предложена методика использования инструментальных программных средств многокомпонентного интерактивного моделирования комплексных задач обработки данных.

Теоретические и практические результаты данной работы использованы при разработке методики интеллектуального анализа данных и поддержки принятия решений в рамках НИР «Исследование и разработка методов, моделей и технологий интеллектуального анализа данных и поддержки принятия решений в топливно-энергетическом комплексе», выполненной при поддержке Минобрнауки России в рамках базовой части Госзадания «Проведение научно-исследовательских работ (фундаментальных научных исследований, прикладных научных исследований и экспериментальных разработок)», Минобрнауки России, договор № 1013140, № гос. Рег. 01201458416, 2014–2016 г.г.

*Апробация результатов работы*. Основные результаты работы докладывались и обсуждались на следующих конференциях: V Международная молодежная научная конференция «Тинчуринские чтения» (Казань, 2010); XVII Международной научно-техническая конференция студентов и аспирантов «Радиоэлектроника, электроника и энергетика» (Москва, 2011); IX и XI Международные научно-технические конференции студентов и аспирантов (Смоленск, 2012, 2014).

1. Разработка способа моделирования комплексных задач обработки данных

1.1. Постановка задачи

Рассмотрим комплексную задачу обработки данных (КЗОД)–сложную задачу, которая состоит из упорядоченного набора взаимосвязанных подзадач (этапов). Каждый этап – отдельная подзадача, решить которую можно несколькими различными способами. Решая подзадачу различными способами, в общем случае мы получаем различные (возможно похожие) решения. Комбинируя различные способы на каждом из этапов, мы получаем различные варианты решений всей задачи.

Под допустимым решением КЗОД будем понимать последовательность способов решения каждого этапа задачи.

Чтобы решить КЗОД нужно найти наиболее подходящее, с точки зрения некоторых критериев, допустимое решение. Для этого необходимо определить критерии, по которым будут оцениваться допустимые решения.

Решение КЗОД ищется не на одном наборе входных данных, а на совокупности большого числа различных входных данных.

Решение задачи на одном наборе входных данных может существенно отличаться от решения на другом наборе входных данных. В связи с этим необходимо выбирать решение, которое будет учитывать результаты на всех или на большей части предоставленных входных данных. В таком случае необходимо задать стратегию выбора оптимального решения.

*Отличительные признаки предлагаемого способа*

К отличительным признакам способа моделирования комплексных задач обработки данных относятся:

* многокомпонентность – разделение задачи на этапы и комбинирование различных способов выполнения каждого из этапов;
* интерактивность – возможность выбора пользователем (экспертом) подходящих вариантов и отсеивания неподходящих;
* оценка, в т.ч. интерактивная.

1.2. Описаниепредлагаемогоспособамоделирования комплексных задач обработки данных

Предлагаемый в работе способ предназначен для решения КЗОД. Его можно разделить формально на несколько этапов:

* разбиение задачи на этапы, формализация этапов задачи;
* поиск и оценка допустимых решений;
* выбор наилучшего решения;
* моделирование КЗОД.

Рассмотрим подробно каждый из этапов.

1.2.1. Разбиение задачи на этапы, формализация этапов задачи

В рамках этого этапа требуется выполнить следующие действия:

* описать в требуемом виде каждый способ решения каждого из этапов;
* описать в требуемом виде критерии, используемые для оценки различных этапов решения задачи;
* указать этапы и / или совокупности этапов, результат выполнения которых будет указывать пользователь (эксперт) интерактивно.

В разделе, посвященном программной реализации, указывается конкретный вид, к которому нужно привести каждый способ решения. Кроме этого, необходимо определенным образом – в виде текстового файла – описать все варианты решения каждого из этапов (ссылки на файлы, в которых описаны алгоритмы выполнения каждого из этапов).

Такая структура взята не случайно. На языке программирования, используемом для реализации инструментальных программных средств (ИПС), такой подход широко распространен.

Критерии, используемые для оценки различных способов выполнения этапа задачи, располагаются в том же месте, где и алгоритмы выполнения этих способов.

В предлагаемом способе есть возможность интерактивно (с помощью экспертной оценки) выбирать подходящие варианты решения задачи. Так, например, на указанном этапе пользователь может выбрать 2 варианта из 5, как подходящие, сократив количество перебираемых в дальнейшем вариантов на 60%. А если указать, что выбор осуществляется по окончанию пары подряд идущих этапов, состоящих, например, из 5 и 7 способов реализации первого и второго из этих этапов соответственно, то количество возможных пар составляет 35. Если, например, пользователь выберет 7 пар, как подходящие, то количество перебираемых вариантов сократиться на 80%.

1.2.2. Поиск и оценка допустимых решений

В настоящее время для поиска решения комплексной задачи обработки данных пользуются следующим алгоритмом:

1. перебрать все входные наборы данных и для каждого набора выполнить пункты 2–3;
2. перебрать все возможные варианты решения каждого из этапов КЗОД для заданных входных данных;
3. оценить каждое из полученных решений;
4. учитывая все полученные результаты, выбрать в соответствии с заданной стратегией лучший вариант решения задачи.

Этот алгоритм достаточно прост и легко поддается реализации, но у него есть существенный недостаток – использование полного перебора для поиска решений задачи с последующей оценкой каждого варианта.У него есть существенный недостаток.С ростом числа подзадач и способов реализации каждой из подзадач, нам приходится перебирать огромное количество вариантов, что значительно увеличивает время поиска решения КЗОД.

Поэтому предлагается следующий алгоритм, лишенный указанного выше недостатка:

1. перебрать все входные наборы данных и для каждого набора выполнить пункт 2;
2. получить допустимые решения и их оценки;
3. учитывая все полученные результаты, выбрать в соответствии с заданной стратегией лучший вариант решения задачи.

Отличие от предыдущего заключается в пункте 2 – в способе получения допустимых решений, который выполняется следующим образом:

1. выполнить всеми способами первый этап комплексной задачи и сохранить отдельно результаты для каждого способа;
2. для каждого из оставшихся этапов выполнить пункты 3–4;
3. получить все результаты предыдущего уровня и для каждого из них выполнить всеми указанными способами действие на текущем этапе;
4. сохранить отдельно все результаты выполнения текущего этапа;
5. после выполнения каждого этапа, начиная с первого, необходимо проверить все результаты на соответствие заранее заданным критериям (если они заданы) и исключить из списка результатов те, которые не удовлетворяют критериям;
6. после выполнения каждого этапа, начиная с первого, необходимо проверить, требуется ли интерактивных выбор результата этапа (последовательности этапов) для текущего этапа и, если требуется, то предоставить пользователю возможность выбора одного или нескольких подходящих вариантов (также оставить возможность указывать, что ни один из вариантов не подошел) и, соответственно, исключения неподходящих;
7. результаты выполнения последнего этапа, которые не были исключены в пунктах 5 и 6 считаются допустимыми, а последовательности способов на каждом из этапов – допустимыми решениями задачи.

Используя предложенный способ, можно получить существенный выигрыш во времени по сравнению с использованием полного перебора. Выигрыш достигается путем отсечения ветвей графа решений с помощью количественной оценки (пункт 5) и экспертной оценки (пункт 6).

Под графом решений в данном случае подразумевается граф, состоящий из слоев, каждый слой соответствует одному из этапов решения задачи. Вершины одного слоя – способы выполнения соответствующего этапа задачи – связаны ребрами только с вершинами предыдущего и последующего слоев, но не между собой. Путь в таком графе от вершины первого слоя до вершины последнего слоя и есть допустимое решение.

В простом способе полного перебора перебираются все листья этого дерева, а в предложенном способе некоторые ветви (группы ветвей) уже на ранних этапах отсекаются.

Этот способ можно еще больше улучшить, если переделать пункт 1 – перебор всех возможных входных данных. Это тоже полный перебор, который возможно будет избыточным для решения задачи. К одному из возможных вариантов улучшения можно отнести кластеризацию входных данных – разбиение всех входных данных на кластеры и использованию не всех, а лишь части данных по каждому кластеру.

Последний пункт – выбор в соответствии с заданной стратегией лучшего варианта решения задачи – рассмотрим отдельно в части, относящейся к выбору наилучшего решения.

В связи с большим количеством входных предусмотрен механизм, на базе которого происходит обработка результатов, например, с использованием статистики. В предлагаемом способе использование статистики результатов по каждому из этапов идет отдельно от основного алгоритма. Использованиестатистических данныхсводится к следующему:

* полному исключению способа выполнения одного из этапов, если этот способ никогда (или менее заданного процента случаев) не удовлетворяет критериям. Такое исключение проводится перед началом очередного прохода по входным данным;
* замене интерактивного выбора вариантов на автоматический выбор с использованием накопленной статистики результатов выбора эксперта.

1.2.3. Выбор наилучшего решения

Выбор наилучшего решения проводится с помощью определенной стратегии. Предлагаемый способ реализуется с использованием ИПС, являющихся платформой, на базе которой можно задавать различные стратегии выбора наилучшего решения.

1.2.4. Моделирование комплексных задач обработки данных

После того как выбрано наилучшее решение задачи формируется структура, необходимая для непосредственной обработки данных. Способы решения каждого из этапов задачи, которые вошли в лучший вариант собираются вместе. После того как структура сформирована, можно с ее помощью обрабатывать любые данных. В разделе, посвященном программным средствам, этот пункт раскрыт более подробно с точки зрения программной реализации. Эта структура называется конфигурацией, а за запуск и выполнение этой конфигурации отвечает отдельная подсистема ИПС.

2. Разработка инструментальных программных средств моделирования комплексных задач обработки данных

2.1. Обоснование технологии разработки инструментальных программных средств моделирования комплексных задач обработки данных

В ходе анализа КЗОД выявлено наличие нескольких различных подклассов этих задач, которые значительно отличаются друг от друга по своему смыслу, но для которых подходит предлагаемый в работе способ решения. Также установлено, что детали реализации способа для разных подклассов КЗОД схожи.

В связи с этим, принято решение о создании ИПС моделирования КЗОД, используя которые можно создавать программное обеспечение (ПО)для решения конкретных задач. В работе такоеПО создается для моделирования задач обработки изображений.

Для реализации поставленной задачи подходит большинство современных языков программирования высокого уровня.

Рассмотрены следующие варианты:

* Delphi;
* C#;
* Java.

Все три языка поддерживают концепцию ООП, которая используется в работе для создания ИПС.

К преимуществам Delphi относят сильную компонентную модель, позволяющую быстро создавать приложения с очень сложным пользовательским интерфейсом.

К недостаткам относят значительное отставание от конкурентов по возможностям языка и отсутствие серьезных фирм, активно продвигающих и развивающих язык.

Языки C# и Java во многом схожи, оба имеют большие возможности по созданию крупных программных продуктов, а также имеют схожий C++ подобный синтаксис.

В качестве языка программирования для разработки ИПС выбран язык Java, вследствие следующих причин:

* язык Java в большей степени кроссплатформенный;
* наличие развитых специализированных библиотек для этого языка;
* сокращенный набор требований для графического пользовательского интерфейса ИПС иПОна его основе.

После выбора языка программирования необходимо выбрать среду разработки. Существует, по меньшей мере, 17 различных сред для разработки приложений на языке Java.

К наиболее распространенным относят следующие:

* NetBeans;
* Eclipse;
* IntelliJ IDEA.

Наиболее развитой средой разработки является IntelliJIDEA, но только в Ultimate версии, которая является платной. В Community версии функционал этой среды разработки не сильно отличается от двух других вариантов. В остальном все три среды разработки предлагают схожий функционал – подсветка кода, упорядочивание структур кода, инструменты рефакторинга кода и т.п.

Выбор сделан в пользу среды разработки Eclipse из-за хорошо развитой системы расширений этой среды, позволяющей легко интегрировать проект с различными системами, упрощающими разработку, такими как системы контроля версий, системы автоматической сборки проекта и т.п.

2.2. Выбор вида входных данных для инструментальных программных средств

При выборе вида входных данных для ИПС рассмотрены следующие варианты:

* использование графического пользовательского интерфейса;
* передача параметров из ПО в ИПС в виде специальных объектов;
* использование файла с необходимыми параметрами.

*Использование графического пользовательского интерфейса.*

Такой подход для ИПС нецелесообразен, так как возникают проблемы:

* интеграции графического интерфейса ИПС в пользовательский интерфейс ПО;
* невозможность в ПО создавать полностью уникальный пользовательский интерфейс;
* сложность при интеграции ПО с ИПС на порядок выше при использовании пользовательского интерфейса в ИПС, так как необходимо в ПО эмулировать ввод данных в ИПС, а также управляющие воздействия, например, нажатия на кнопки.

*Передача параметров из ПО в ИПС в виде специальных объектов.*

Такой подход возможен. Его гораздо проще реализовать, чем согласовыватьдва пользовательских интерфейса.

Достоинством такого подхода является относительная простота реализации.

Недостатками такого подхода в случае создания ИПС являются:

* отсутствие гибкости в перечне параметров, передаваемых в ИПС;
* некоторое нарушение инкапсуляции ИПС, выраженное в необходимости перечислять весь перечень параметров и их типы в интерфейсных методах ИПС;
* накладные расходы, связанные с изменением числа и состава параметров, необходимых при корректировке ИПС и соответственно ПО, так как эти изменения приведут к необходимости корректировать и ПО.

Часть недостатков достаточно просто компенсируется при использовании техники «IntroduceParameterObject»– введения объекта параметров, который хранит в себе все параметры, передаваемые в систему.

Однако даже при такой реализации разработчику ПО придется самостоятельно создавать и заполнять такой объект перед тем, как его передать в ИПС.

*Использование файла с необходимыми параметрами.*

Этот способ распространен примерно также как и предыдущий.

Достоинствамиявляются:

* гибкость интерфейса ИПС, при изменении типов и количества параметров нет необходимости производить изменения вПО;
* простота передачи из ПО в ИПС, как и в случае с объектом параметров.

Недостатком является сложность, связанная с извлечением данных из файла.

*Использование файла с необходимыми параметрами.*

В качестве вида входных данных выбран именно файл, так как:

* использование файла минимизирует работу по созданию ПО;
* единственный существенный недостаток компенсируется подходом, рассмотренным ниже.

После выбора вида входных данных рассмотрим разновидности форматов файлов, которые могут быть использованы.

* XML;
* JSON;
* RDF;
* электронныетаблицы;
* файлыс разделителями (CSV);
* текстовыйдокумент;
* простойтекст;
* Проприетарные форматы.

Выбор сделан в пользу формата XML, средства работы с которыми наиболее развиты. В качестве инструмента, компенсирующего сложность использования файла, применяется технология JAXB (JavaArchitectureforXMLBinding). Технология JAXB позволяет преобразовывать xml файл в набор java-объектов, которые используются в ИПС, и наоборот.

Реализоваводин раз преобразование с помощью JAXB, можно компенсироватьпроблемы, связанные с использованием входных данных в виде файла, получая так называемый объект параметров, рассмотренный ранее.

Преимуществом такого подхода является то, что функции разбора содержимого файла и получения объекта параметров берет на себя ИПС.

2.3. Интерфейс инструментальных программных средств

С точки зрения интерфейса имеет смысл рассматривать отдельно интерфейсы ПО и ИПС.

Рассмотрим интерфейс ИПС. Интерфейс ИПС представлен следующим образом:

* на вход ИПС поступает xml-файл с конфигурацией для решаемой задачи;
* имеется основной метод (функция), вызов которого запускает поиск оптимального решения КЗОД;
* на выходе у ИПС – одно или несколько наилучших решений, представленных в виде конфигурации для подсистемы моделирования КЗОД;
* интерактивная составляющая ИПС – та часть ИПС, которая запрашивает данные у пользователя и использует их для своих целей, – взаимодействует с ПО, предоставляющее механизм взаимодействия с пользователем.

Интерфейс подсистемы моделирования КЗОД является частью общего интерфейса ИПС и представлен следующим образом:

* на вход поступает xml-файл с оптимальной конфигурацией для решаемой задачи;
* имеется метод (функция), который запускает моделирование КЗОД с использованием варианта решения, переданного в файле конфигурации;
* на выходе формируются результаты решения КЗОД на заданном входном наборе с помощью оптимального варианта решения.

2.4. Интерфейс программного обеспечения

Рассмотрим интерфейс ПО, разработанного на основе и с использованием предлагаемых ИПС.

* ПО предоставляет весь необходимый графический интерфейс пользователя;
* при необходимости,ПО может предоставлять пользователю возможность создавать и редактировать конфигурацию для ИПС;
* ПО предоставляет конкретную реализацию механизма взаимодействия с пользователем (например, в виде оконных сообщений), так как интерактивная составляющая ИПС взаимодействует с ПО.

При использовании такой связи между ИПС и ПО большая часть функций выполняют ИПС, которые разработаны и представлены в работе, и лишь небольшая часть функций выполняется с помощью ПО, которое разрабатывает пользователь ИПС.

Разработчик ПО должен обеспечить взаимодействие с пользователем, если этого требует решаемая задача. Графический интерфейс пользователя, а также любые другие элементы и функции добавляются в ПО только для того, чтобы обеспечить дополнительное удобство пользователям ПО.

В результате, пользователь ИПС – разработчик ПО – может приложить минимум усилий для решения своей задачи, так как весь нужный набор функций уже есть в ИПС.

2.5. Описание подсистем инструментальных программных средств

Инструментальные программные средства – сложный инструмент, который состоит из следующих подсистем:

* подсистема поиска оптимального решения (оптимальных решений);
* подсистема моделирования КЗОД;
* подсистема интерактивного взаимодействия с пользователем.

Подсистемапоискаоптимальногорешения:

* предоставляет возможность задать параметры поиска оптимального решения в файле конфигурации;
* ищетоптимальноерешение КЗОД;
* использует возможности подсистемы интерактивного взаимодействия для получения пользовательских оценок итоговых вариантов решения КЗОД, а также оценок промежуточных этапов решения КЗОД.

К параметрам поиска, принимаемым подсистемойпоискаоптимальногорешения, относятся:

* необходимость интерактивного взаимодействия с пользователем для оценки результатов этапа КЗОД;
* необходимость оценки результатов работы этапа КЗОД.

Подсистемамоделирования КЗОД:

* использует конфигурацию решения КЗОД для моделирования;
* предоставляет возможность использования дополнительных параметров при моделировании КЗОД.

В качестве дополнительных параметров могут выступать, например, следующие параметры:

* пути к различным папкам с необходимыми файлами или папкам, в которые будут помещаться результаты работы;
* параметры для самих алгоритмов выполнения этапов КЗОД, которые было решено вынести отдельно в конфигурацию с целью подбора их значений пользователем ИПС.

Сам подбор параметров для алгоритмов выполнения этапов КЗОД является нетривиальной задачей и не рассматривается в работе. Такой подбор может быть выполнен пользователем ИПС при необходимости самостоятельно.

Подсистема интерактивного взаимодействия с пользователем позволяет:

* запрашивать у пользователя оценку результатов выполнения этапа (группы этапов) КЗОД;
* запрашивать необходимые параметры у пользователя, значения которых не были явно указаны в конфигурациях для других подсистем.

2.6. Взаимодействиеинструментальных программных средств и программного обеспечения

Опишем более подробно связь между ИПС и ПО.

* ИПС являются своего рода библиотекой функций, которые будут использоваться в ПО;
* ИПС ничего не знают о деталях реализации ПО, которое разрабатывается на их основе, в то время как ПО знают, как взаимодействовать с ИПС для решения поставленной задачи;
* ПО использует функции и возможности ИПС через их интерфейс, при этом являясь отдельным приложением;
* ПО предоставляет конечным пользователям графический интерфейс, помогающий удобно и эффективно взаимодействовать с ИПС для решения поставленной задачи

ИПС ничего не знают о конкретных деталях реализации ПО, но в них заложены предположения о возможных видах пользовательского интерфейса ПО. Это сделано с целью делегирования ПО конкретных способов интерактивного взаимодействия с пользователем ПО. ИПС используют в алгоритмах абстрактный запрос данных у пользователя, а ПО реализует этот запрос, например, с помощью оконных форм с вводом/выводом требуемых данных.

Схема взаимосвязей подсистем ИПС, ПО и пользователя представлена на рисунке 1.



Рисунок 1. Схема взаимосвязей подсистем ИПС, ПО и пользователя

2.7. Методика использованияинструментальных программных средстви алгоритмы поиска решения КЗОД

Опишем методику использования разработанных ИПС:

* проанализировать КЗОД и выделить относительно самостоятельные и независимые этапы, которые могут быть выполнены не однозначно, а с использованием разных алгоритмов, способов;
* описать каждый вариант реализации этапа КЗОД;
* для каждого варианта при необходимости задать способ количественной и / или качественной оценки этого варианта;
* указать другие параметры, необходимые для поиска оптимального решения, например, пути к папкам, значения определенных параметров задачи и т.п.;
* вПО, например, в коде обработки запроса пользователя на поиск решения, запустить поиск оптимального варианта в ИПС;
* генерируются файлы конфигурации для решений, которые были выбраны, как оптимальные;
* выбрав одно из решений, необходимо вПО, например, по другому запросу пользователя, запустить моделирование КЗОД в ИПС;
* в некоторых случаях имеет смысл разделить ПО на две части – одну для поиска оптимального решения, другую – для использования оптимального решения для удовлетворения нужд пользователя ПО.

На рисунке 2 изображена диаграмма последовательности действий при взаимодействии пользователя с ПО и ИПС для решения задачи.

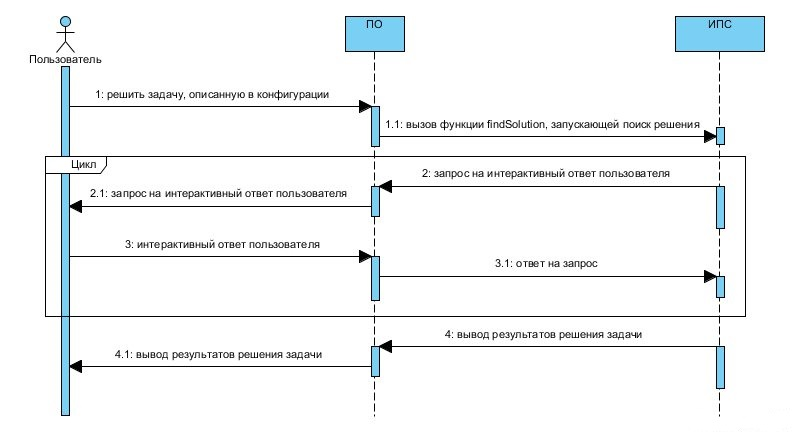


Рисунок 2. Диаграмма последовательности действий при взаимодействии пользователя с ПО и ИПС для решения задачи

*Разделение различных этапов задачи и их формальное описание*

Рассмотрим подробнее разделение КЗОД на этапы и описание вариантов выполнения каждого из этапов.

* КЗОД делится на последовательность этапов таким образом, чтобы результаты предыдущего этапа были входными данными для последующего этапа;
* все варианты реализации одного и того же этапа КЗОД должны быть одинаковыми по типу и смыслу, но могут существенно отличаться значениями выходных данных на одном и том же наборе входных данных;
* программно все варианты реализации каждого из этапов решения задачи должны быть описаны на языке Java в классах, реализующих интерфейс Executable– интерфейс, разработанный для ИПС, необходимый для корректного запуска последовательности этапов КЗОД. Сами реализации создаются в рамках разработки ПО, а ПО передает их в ИПС при запросе от ИПС автоматически. В конфигурацию, передаваемую в ИПС,необходимо поместитьтольконазванияклассов;
* все данные, не относящиеся напрямую к результатам предыдущего этапа КЗОД, передаются через специальный объект, названный как Context.

*Описание количественной и качественной оценки способа реализации этапа задачи*

Для решения КЗОД необходимо также обеспечивается возможность оценки результатов выполнения варианта реализации этапа КЗОД. Для этого используется расширение интерфейса Executable – ExecutableExt, которое содержит дополнительные методы, с помощью которых можно получить оценку результата и затем ее использовать для выявления наилучшего решения или для отсеивания неподходящих решений.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Рисунок 3. Алгоритмы поиска решения КЗОД | | |

К достоинствам такого подхода можно отнести то, что реализация алгоритма выполнения этапа КЗОД и расчет оценки результатов этого алгоритма находятся в одном файле с точки зрения физической организации и в одном классе с точки зрения логической организации.

К недостаткам можно отнести тот факт, что после нахождения решения та часть, которая отвечает за оценку результатов, использоваться для моделирования не будет. Также не будут использоваться все классы, в которых реализованы способы, не вошедшие в оптимальный вариант решения КЗОД.

2.8. Методика создания программного обеспечения с использованием предлагаемых инструментальных программных средств

Так как ИПС имеют относительно независимые подсистемы для поиска решения КЗОД и для моделирования КЗОД, то может быть создано как одно ПО, так и два раздельных ПО. Далее рассмотрена методика создания единого ПО, но она может быть распространена и на вариант с двумя раздельными ПО:

* создать пустой проект – шаблон ПО;
* добавить в список используемых библиотек файл, включающий в себя ИПС;
* добавить элементы пользовательского интерфейса, если они необходимы;
* описать алгоритмы, реализующие варианты этапов КЗОД, в отдельных классах с использованием интерфейса ExecutableExt или его адаптеров;
* добавить обработчики элементов пользовательского интерфейса (кнопок), каждый из которых состоит из 3–4 строк вызова функций ИПС – подсистемы поиска решений КЗОД и подсистемы моделирования КЗОД с указанием соответствующих конфигураций;
* в конфигурации поиска решений КЗОД для создаваемого ПО описать разделение задачи на этапы (теги tasks), разделение способов реализации этапа (теги variants), ссылки на созданные классы, реализующие требуемые алгоритмы;
* в конфигурации моделирования решения КЗОД указываются ссылки на созданные классы, реализующие требуемые алгоритмы (теги lineElement внутри тега ElementsPool).

При создании двух отдельных ПО придется частично дублировать реализацию алгоритмов в каждом изПО, реализация алгоритмов оценки не дублируется, так как для моделирования решения КЗОД она не требуется.

2.9. Оценка эффективности использованияинструментальных программных средств

Для оценки эффективности измеряется время, необходимое для решения комплексной задачи обработки данных. Как уже отмечалось выше, предложенный способ состоит из следующих этапов:

* разбиение задачи на этапы, формализация этапов задачи;
* поиск и оценка допустимых решений;
* выбор наилучшего решения;
* моделирование КЗОД.

Оценка эффективности проводилась для каждого из этапов отдельно, а затем результат аккумулировался.

В связи с тем, что задачу необходимо явно разбивать на отдельные этапы и формализовать их, затраты времени на выполнение этого этапа решения КЗОД возросли существенно – в среднем, в 2 раза. Такое разбиение достаточно трудоемко, выполняется разработчиком ПО, оно субъективно, но это все окупается за счет выигрыша на остальных этапах при использовании ИПС.

Поиск и оценка допустимых решений производится уже средствами ИПС. Этим объясняется достаточно существенный выигрыш во времени выполнения этого этапа – в среднем, в 3 раза.

При выборе наилучшего решения, а затем и при моделировании КЗОД, используются заданные на первом этапе критерии и стратегия выбора. В связи с этим также получается некоторый выигрыш во времени.В среднем, при решении всей задачи, время, необходимое для решения КЗОД, уменьшилось в 2.5 раза.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были разработаны способ моделирования комплексных задач обработки данных, а также инструментальные программные средства, реализующие этот способ. Разработано также тестовое ПО, решающее конкретную задачу на основе ИПС. Разработанные ИПС продемонстрировали свою эффективность, позволив существенно повысить скорость разработки ПОдля решения комплексных задач обработки данных.

СПИСОК ПУБЛИКАЦИЙ

1. Чандер А.П. Способ и программные средства многокомпонентного интерактивного моделирования комплексных задач обработки изображений //Сборник трудов XI международной научно-технической конференции студентов и аспирантов «Информационные технологии, энергетика и экономика», 2014 (в печати).
2. Чандер А.П.Исследование физики квантовых переходов в гетерогенных структурах//Сборник тезисов докладов пятой международной молодежной научной конференции «Тинчуринские чтения». 28-29 апреля 2010. Казанский государственный энергетический университет. Казань. 2010. Т.1. С. 260-261.
3. Чандер А.П. Исследование квантовых переходов в гетерогенных структурах//Сборник трудов семнадцатой международной научно-технической конференции студентов и аспирантов «Радиоэлектроник, электроника и энергетика». 24-25 февраля 2011. ГОУВПО «МЭИ (ТУ)». Москва. 2011. Т.1. С.178-179.
4. Чандер А.П. Разработка программного обеспечения для вывода решений на основе нейронной сети модели многослойного персептрона// Информационные технологии, энергетика, экономика. Сборник трудов девятой международной научно-технической конференции студентов и аспирантов. 19-20 апреля 2012. ГОУВПО «МЭИ (ТУ)». Смоленск. 2012. Т.1. С.151-155.